



TPC Benchmark™ C
Full Disclosure Report

NEC Express5800/180Ra-7 (4 SMP)

**Using Microsoft SQL Server 2000, Enterprise Edition
and Microsoft Windows2000 Advanced Server**

First Edition
Submitted for Review
June , 2000

NEC, the Sponsors of this benchmark test, believe that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. The Sponsor assumes no responsibility for any errors that may appear in this document. The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, The Sponsor provides no warranty of the pricing information in this document.

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

All performance data contained in this report were obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly. NEC do not warrant or represent that a user can or will achieve similar performance expressed in transactions per minute (tpmC) or normalized price/performance (\$/tpmC). No warranty of system performance or price/performance is expressed or implied in this report.

Copyright 2000 NEC Corporation.

All rights reserved.

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

Printed in USA, 2000

NEC and Express5800 are registered trademarks of NEC Corporation.

TPC Benchmark, TPC-C and tpmC are trademarks of the Transaction Processing Performance Council.
Microsoft, Windows 2000 and SQL Server are registered trademarks of Microsoft Corporation.
Intel, and Pentium are registered trademarks of Intel Corporation.

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



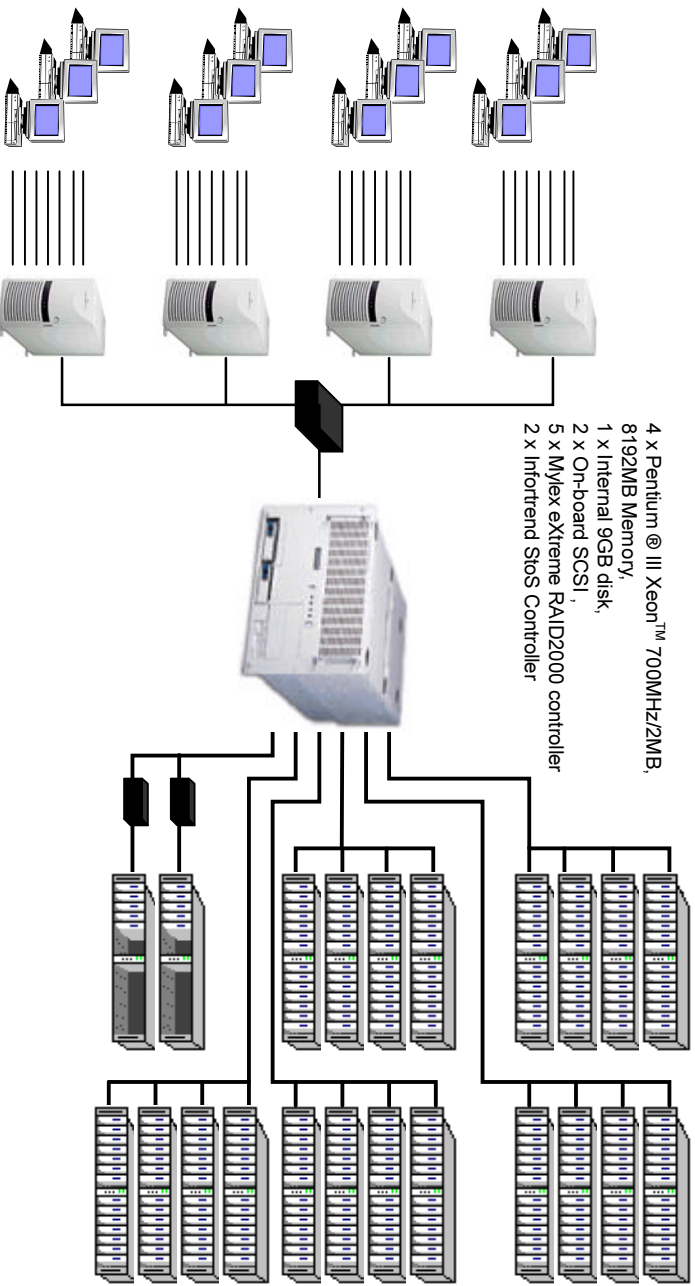
Express5800/180Ra-7 C/S

TPC-C Rev.3.5
Reported Date
June 23 ,2000

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
\$ 592,075	31,764.37 tpmC	\$18.63 per tpmC	September 29,2000
Processors	Database Manager	Operating System	Other Software
4 x Pentium®III Xeon™ 700MHz 2MB L2 cache	Microsoft SQL Server 2000 Enterprise Edition	Microsoft Windows 2000 Advanced Server	Windows2000 Server w/ COM+, IIS5.0 Microsoft VC++
			Number of Users
			25,200

Total of 25,200 PCs

Server : Express5800/180Ra-7



System Component	Server	Each Dient	
Processors	4	2	
Cache	Pentium® III Xeon™ 700MHz 2MB L2 Cache	Pentium® III 866MHz 256KB L2 Cache	
Memory	1	1	
	8192MB	512MB	
Disk Controllers	6	1	
	Mylex eXtreme RAID 2000 On-board SCSI Infortrend S2S controller	On-board SCSI	
Disk Drives	281	1	
	9GB (8.47GB usable) 18GB (16.95 GB usable)	9GB	
Total Storage			
	2473 GB	9GB	
Others	1	2	
	1	1	
	CD-ROM Drive On-board Ether controller	Adaptec 4port Ether NIC CD-ROM Drive On-board Ether controller	



NEC Express5800/180Ra-7
C/S

TPC-C REV 3.5

Report Date:

June 23, 2000

Description	Part Number	Brand	Third Party		Unit Price	Qty	Extended Price	5-yr Mnt. Price
			Pricing	Price				
Server Hardware								
Express5800/180Ra-7 system								
Base system with 1 x Pentium III Xeon 700MHz/2MB	NS070000555	NEC	1		9,160	1	9,160	2,198
Pentium III Xeon 700MHz/2MB BTO Option	UBP-2641-1C-IN	NEC	1		3,899	3	11,697	2,807
2GB (4 x 512MB) memory,	AMS-7020-IN-00	NEC	1		8,599	4	34,396	8,255
9GB 10K rpm HDD,	BDH-1968-IN-00	NEC	1		419	1	419	101
DDS-3 DAT Drive	ADT-2410-IN-00	NEC	1		749	1	749	180
CD-ROM, On-board LAN, KBMS,	Included	NEC	1		0	1	0	0
Windows2000 Advanced Server Hot Loaded	NUT-A200-IN-US	Microsoft	1		3,459	1	3,459	0
NEC MultiSync A500+	011145	NEC	1		179	1	179	43
	Subtotal						60,059	13,584
Disk Subsystem								
extreme RAID2000 4channel controller (+2spares)	E2000-4-64NB	Mylex	3		2,050	8	16,400	0
3U Disk Expansion Unit(14 slots)	ADR-0802-IN-00	NEC	1		2,000	22	44,000	10,560
9GB 10K rpm HDD	BDH-1960-IN-00	NEC	1		419	280	117,320	28,157
18GB 10K rpm HDD	BDH-1860-IN-00	NEC	1		599	10	5,990	1,438
42U Rackframe	AZA-4200-00-00	NEC	1		1,879	2	3,758	902
Infotrend S2S RAID Controller (+2spares)	IFT-3102U2G	Infotrend	4		2,092	4	8,368	
64MB EDO SIMM								
Battery back-up module w/ battery (+2spares)	IFT-9070B	Infotrend	4		170	4	680	0
APC Smart UPS (+2 spares)	SU2200NET	APC	5		878	4	3,512	0
	Subtotal						200,028	41,056
Server Software								
SQL Server/2000 Ent. Edition, per process licensing		Microsoft	2		19,999	4	79,996	10,475
Windows2000 Advanced Server, 25 Client Licenses		Microsoft	2		(included in svr.)	1	0	0
	Subtotal						79,996	10,475
Client Hardware								
NEC Express5800/120Ld								
Base System with 1 x Pentium III 866MHz/256KB	NS020001855	NEC	1		2,909	4	11,636	2,793
1 x Pentium III 866MHz/256 BTO Option	UBP-2718-IN-00	NEC	1		1,599	4	6,396	1,535
1 x 128MB memory,	AMS-5128-IN-00	NEC	1		399	16	6,384	1,532
1 x 9GB 10K rpm HDD,	CDH-1942-00-00	NEC	1		419	4	1,676	402
CD-ROM, On-board LAN, KBMS	Included	NEC	1		0	4	0	0
Adaptec ANA-62044 NIC (+2spares)	ANA-62044-SGL	Adaptec	5		528	10	5,280	0
NEC MultiSync A500+	011145	NEC	1		179	4	716	172
	Subtotal						32,088	6,434
Client Software								
Microsoft Windows 2000 Server, w/5 Client Licenses		Microsoft	2		999	4	3,996	0
Visual C++ Professional 6.0		Microsoft	2		549	1	549	0
	Subtotal						4,545	0
User Connectivity								
Cisco Catalyst2912-XL-A (+2spares)	WS-C2912-XL-A	Cisco	5		1,110	3	3,330	0
Complex 16-pt 10Mbps HUB (+10% Spare)	TP1016C-110V	Complex	6		80	1756	140,480	0
	Subtotal						143,810	0
	TOTAL						520,526	71,549

Notes:
 All Microsoft maintenance is covered by the maintenance costs of Microsoft SQL Server
 Pricing: 1-NEC 2-Microsoft 3-Mylex 4-ADS 5-CDW 6-SDD
 3,4.5,6 : with 5-year warranty

Audited by Francois Raab, InfoSizing, Inc

Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflects 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 terms, please inform the TPC at pricing@tpc.org. Thank you.

5-Yr. Cost of Ownership: **\$592,075**
 tpmC Rating: **31764.37**

\$ / tpmC: 18.63

Numerical Quantities Summary

MQTh, Computed Maximum Qualified Throughput

31764.37 tpmC

%throughput difference ,reported & reproducibility runs

0.13 %

<u>Response Times(in seconds)</u>	<u>90%</u>	<u>Average</u>	<u>Maximum</u>
New-Order	0.30	0.23	5.17
Payment	0.19	0.15	4.73
Stock-Level	1.49	0.95	5.43
Delivery(interactive portion)	0.11	0.10	0.72
Delivery(deferred portion)	0.52	0.31	3.56
Order-status	0.23	0.18	3.77
Menu	0.11	0.10	1.17

Response time delay added for emulated components

0.1

Transaction Mix , in percent of total transaction

New-Order	44.86%
Payment	43.05%
Order-status	4.02%
Delivery	4.05%
Stock-level	4.02%

Keying/Think Times (in seconds)

	<u>Min.</u>	<u>Average</u>	<u>Max</u>
New-Order	18.00 0.00	18.01 12.04	18.04 120.51
Payment	3.00 0.00	3.02 12.05	3.05 120.50
Stock-Level	2.00 0.00	2.02 5.01	2.05 50.50
Delivery	2.00 0.00	2.02 5.06	2.04 50.50
Order-status	2.00 0.00	2.02 10.11	2.05 100.50

Test Duration

Ramp-up time	58 minutes
Measurement interval	30 minutes
Number of checkpoints	1
Checkpoint interval	30 minutes
Number of transactions (all types) completed in measurement interval	2,210,511

ABSTRACT	1
TPC BENCHMARK TM C METRICS	1
STANDARD AND EXECUTIVE SUMMARY STATEMENTS	1
AUDITOR.....	1
PREFACE	2
TPC BENCHMARK TM C OVERVIEW	2
DOCUMENT STRUCTURE	2
GENERAL ITEMS	3
ORDER AND TITLES	3
SUMMARY STATEMENT	3
NUMERICAL QUANTITIES SUMMARY	3
APPLICATION PROGRAM	3
SPONSOR	4
PARAMETERS AND OPTIONS	4
CONFIGURATION DIAGRAMS	4
MEASURED CONFIGURATION	5
PRICED SYSTEM CONFIGURATION	6
CLAUSE 1 : LOGICAL DATABASE DESIGN AND RELATED ITEMS	7
TABLE DEFINITIONS	7
TABLE ORGANIZATION	7
INSERT AND DELETE OPERATIONS	7
DISCLOSURE OF PARTITIONING	7
REPLICATION OF TABLES	7
ADDITIONAL AND/OR DUPLICATED ATTRIBUTES IN ANY TABLE	7
CLAUSE 2 : TRANSACTION AND TERMINAL PROFILES RELATED ITEMS	8
RANDOM NUMBER GENERATION	8
TERMINAL INPUT/OUTPUT SCREEN LAYOUT	8
TERMINAL FEATURE VERIFICATION.....	8
PRESENTATION MANAGER OR INTELLIGENT TERMINAL	8
TRANSACTION PROFILES	8
TRANSACTION MIX.....	9
QUEUEING MECHANISM	9
CLAUSE 3 : TRANSACTION AND SYSTEM PROPERTIES RELATED ITEMS	10
TRANSACTION SYSTEM PROPERTIES (ACID)	10
ATOMICITY TESTS	10
Completed Transactions	10
Aborted Transactions	10
CONSISTENCY TESTS	10
ISOLATION	10
DURABILITY	11
<i>Loss of Memory and Loss of Log</i>	<i>11</i>
<i>Loss of Data</i>	<i>11</i>
CLAUSE 4 : SCALING AND DATABASE POPULATION RELATED ITEMS	12
INITIAL CARDINALITY OF TABLES	12
DISTRIBUTION OF TABLES AND LOGS	13
TYPE OF DATABASE	14
DATABASE MAPPING.....	14
180-DAYS SPACE	14
CLAUSE 5 : PERFORMANCE METRICS AND RESPONSE TIME RELATED ITEMS	15
THROUGHPUT.....	15

RESPONSE TIMES.....	15
KEYING AND THINK TIMES.....	15
RESPONSE TIME FREQUENCY DISTRIBUTION CURVES AND OTHER GRAPHS.....	15
RESPONSE TIME VERSUS THROUGHPUT PERFORMANCE CURVE.....	18
NEW-ORDER THINK TIME.....	19
NEW-ORDER THROUGHPUT vs. ELAPSED TIME.....	19
STEADY STATE.....	20
WORK PERFORMED DURING STEADY STATE.....	20
REPRODUCIBILITY.....	20
MEASUREMENT PERIOD DURATION.....	20
REGULATION OF TRANSACTION MIX.....	20
TRANSACTION STATISTICS.....	20
CHECKPOINT COUNT AND LOCATION.....	20

CLAUSE 6 : SUT, DRIVER, AND COMMUNICATION DEFINITION RELATED ITEMS 21

DESCRIPTIONS OF RTE.....	21
EMULATED COMPONENTS.....	21
FUNCTIONAL DIAGRAMS AND DETAIL OF DRIVER SYSTEM.....	21
NETWORK CONFIGURATIONS AND DRIVER SYSTEM.....	21
NETWORK BANDWIDTH.....	21
OPERATOR INTERVENTION.....	21

CLAUSE 7 : PRICING RELATED ITEMS 22

HARDWARE AND SOFTWARE COMPONENTS.....	22
AVAILABILITY.....	22
THROUGHPUT, AND PRICE PERFORMANCE.....	22
COUNTRY SPECIFIC PRICING.....	22
USAGE PRICING.....	22

CLAUSE 9 : AUDIT RELATED ITEMS..... 23

AUDITOR'S REPORT.....	23
AVAILABILITY OF THE FULL DISCLOSURE REPORT.....	23
AUDITOR'S LETTER.....	24

APPENDIX A : APPLICATION SOURCE CODE..... 26

APPENDIX B : DATABASE DESIGN..... 91

APPENDIX C : TUNABLE PARAMETERS..... 119

APPENDIX D : SPACE CALCULATION..... 154

APPENDIX E : PRICE QUOTATION..... 155

Abstract

This report documents the compliance of NEC Corporation's TPC Benchmark™ C tests on the NEC Express 5800/180Ra-7 client/server system with version 3.5 of the TPC Benchmark C Standard Specification. 4 Clients (NEC Express 5800/120Ld) were used as the front-end clients.

The operating system and the DBMS used on the server were Microsoft Windows 2000 Advanced Server and Microsoft SQL Server 2000. The operating system on the clients was Microsoft Windows 2000 Server. Those clients ran Microsoft's IIS server 5.0 and COM+.

Two standard metrics, transaction-per-minute-C(tpmC) and price per tpmC(\$/tpmC) are reported, in accordance with the TPC Benchmark™ C Standard. The independent auditor's report by Francois Raab appears at the end of this report.

TPC Benchmark™ C Metrics

The standard TPC Benchmark™ C metrics, tpmC (transactions per minute), price per tpmC (five year capital cost per measured tpmC) are reported.

System	SW	Total System Cost	tpmC	\$ per tpmC	Availability Date
NEC Express5800/180Ra-7	Microsoft Windows 2000 Advanced Server Microsoft SQL Server 2000, Enterprise Edition	\$592,075	31764.37	\$18.63	September 29, 2000

Standard and Executive Summary Statements

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

Auditor

The benchmark configuration, environment and methodology were audited by Francois Raab of Information Paradigm, Inc. to verify compliance with the relevant TPC specification s.

Preface

The TPC Benchmark™ C was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry. This full disclosure report is based on the TPC Benchmark™ C Standard Specifications Version 3.5.

TPC Benchmark™ C Overview

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

TPC Benchmark™ C is an On Line Transaction Processing (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 of 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.

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 benchmark when critical capacity planning and/or product evaluation decisions are contemplated.

Document Structure

This TPC Benchmark™ C Full Disclosure Report is organized as follows:

- The main body of the document lists each item in Clause 8 of the TPC-C Standard and explains how each requirement is satisfied.
- Appendix A contains the source code of the TPC-C application code used to implement the TPC-C transactions.
- Appendix B contains the database definition and population code used in the tests.
- Appendix C contains the tunable parameters used in the TPC-C tests.
- Appendix D contains space calculation table.
- Appendix E contains third-party price quotations.

TPC Benchmark™ C Full Disclosure

The TPC Benchmark™ C Standard Specification requires test sponsors to publish, and make available to the public, a full disclosure report for the results to be considered compliant with the Standard. The required contents of the full disclosure report are specified in Clause 8. This report is intended to satisfy the Standard's requirement for full disclosure. It documents the compliance of the benchmark tests with each item listed in Clause 8 of the TPC Benchmark™ C Standard Specification.

In the Standard Specification, the main headings in Clause 8 are keyed to the other clauses. The headings in this report use the same sequence, so that they correspond to the titles or subjects referred to in Clause 8.

Each section in this report begins with the text of the corresponding item from Clause 8 of the Standard Specification, printed in italic type. The plain text that follows explains how the tests comply with the TPC Benchmark™ C requirement. In sections where Clause 8 requires extensive listings, the section refers to the appropriate appendix at the end of this report.

General Items

Order and titles

The order and titles of sections in the Test Sponsors Full Disclosure Report must correspond with the order and titles of for TPC-C standard specification. The intent is to make it as easy as possible for readers to compare and contrast material in different Full Disclosure reports.

The order and titles of sections in this report correspond with that of the TPC -C standard specification.

Summary Statement

The TPC Executive Summary Statement must be included near the beginning of the Full Disclosure.

The TPC Executive Summary Statement is included at the beginning of this report.

Numerical Quantities Summary

The numerical quantities listed below must be summarized near the beginning of the Full Disclosure Report.

- *measurement interval in minutes,*
- *number of checkpoints in the measurement interval,*
- *computed maximum Qualified Throughput in tpmC,*
- *percentage difference between reported throughput and throughput obtained in reproducibility run,*
- *ninetieth percentile, average and maximum response times for the New-Order, Payment, Order-Status, Stock-Level, Delivery(deferred and interactive) and Menu transactions,*
- *time in seconds added to response time to compensate for delays associated with emulated components, and percentage of transaction mix for each transaction type.*

These numerical quantities are summarized at the beginning of this report.

Application Program

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

Appendix A contains the application source codes used in the TPC -C benchmark.

Sponsor

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

This benchmark test was sponsored by NEC Corporation . NEC has authorized NEC Corp. to publish TPC -C performance and price/performance results for the NEC Epress5800/180Ra-7. Price quotations contained in Appendix E correspond to the NEC Epress5800/180Ra-7 server.

Parameters and Options

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

- Database tuning options
- Recovery/locking options
- Operating system and application configuration parameters

Appendix C contains the tunable parameters used in the TPC -C tests.

Configuration Diagrams

Provide diagrams of both the measured and priced configurations, accompanied by a description of the differences. This includes, but not limited to:

- Number and type of processors
- Size of allocated memory, and any specific mapping/partitioning or memory unique to the test
- Number and type of disk drive units (and controllers, if applicable)
- Number of channels or bus connections to disk units, including their protocol type
- Number of LAN(e.g. Ethernet) connections, including routers, workstations, terminals, etc., that were physically used in the test or are incorporated into the pricing structure.
- Type and the run-time execution location of software components(e.g., DBMS, client processes, transaction monitors, software drivers, etc.)

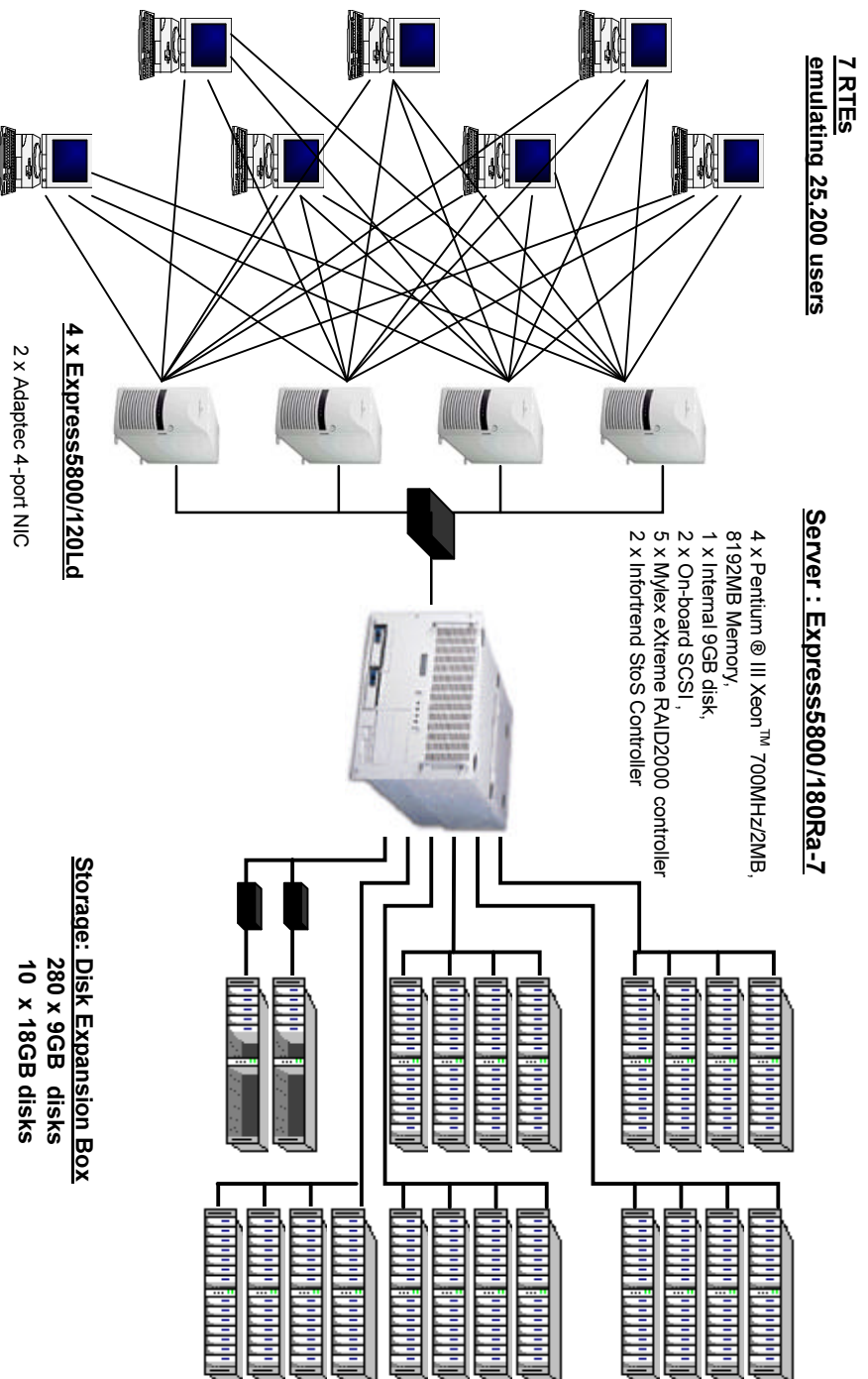
Figure 1.1 shows the measured configuration diagram.

Figure 1.2 shows the priced configuration diagram.

Measured Configuration

The following figure represents the measured configuration. The benchmark system used a remote terminal emulator(RTE) to initiate transactions and measure response times of transactions, as well as record various data for each transaction.

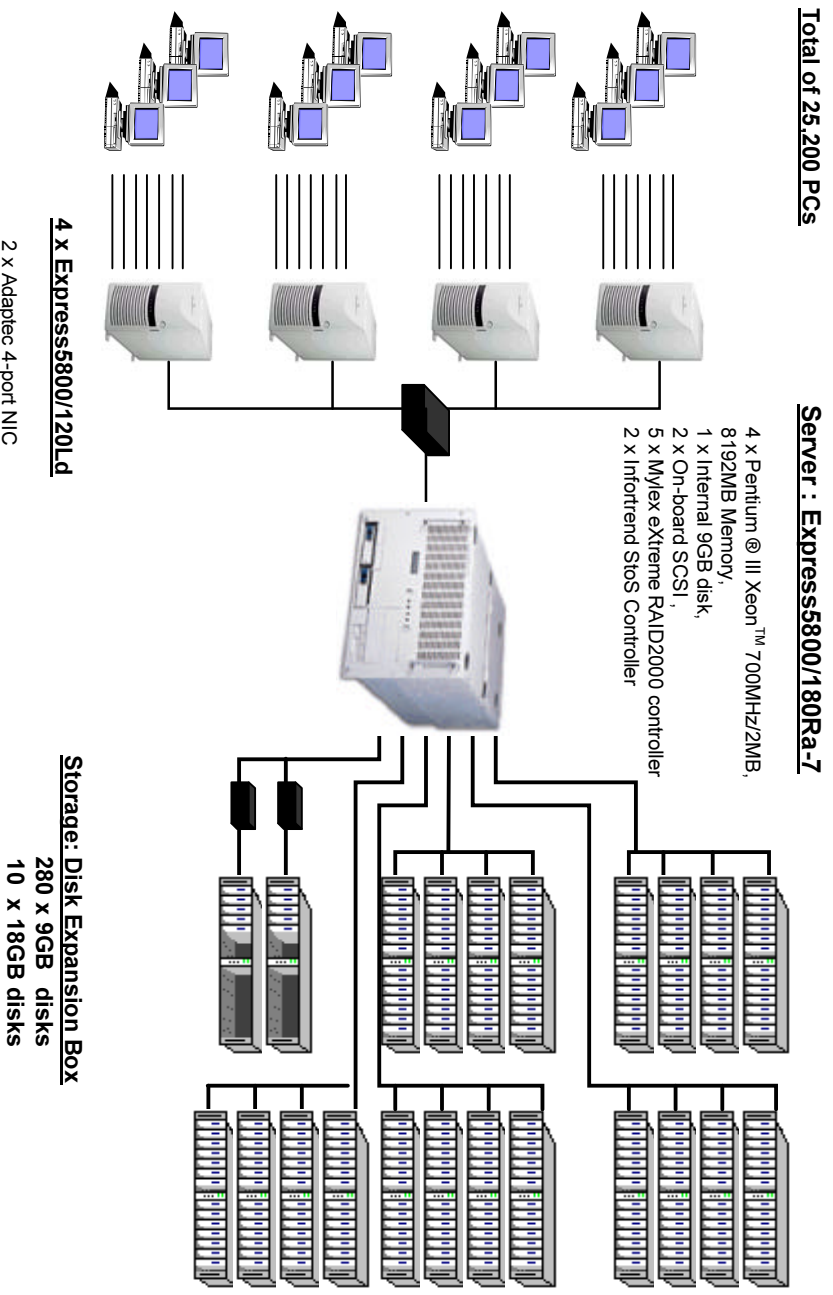
Figure 1.1 Express5800/180Ra-7, Measured Configuration Diagram



Priced System Configuration

The following figure depicts the priced system, whose cost determines the normalized price per tpmC reported for the test.

Figure1.2: Express5800/180Ra-7, Priced Configuration Diagram



Clause 1 : Logical Database Design and Related Items

Table Definitions

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

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

Table Organization

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

Appendix B contains the code used to define the physical organization of tables and indices

Insert and Delete Operations

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

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

Disclosure of Partitioning

While there are a 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 on any table in this benchmark.

Replication of Tables

Replication of tables, if used, must be disclosed.

No tables were replicated in this benchmark test.

Additional and/or Duplicated Attributes in any Table

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

No duplications or 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 described.

Random numbers were generated internally by the Microsoft BenchCraft RTE program which was already audited independently.

Terminal Input/Output Screen Layout

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

All screen layouts followed the specifications exactly.

Terminal feature Verification

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

Each of five transaction types was tested by the auditor. The auditor verified that all the features specified in Clause 2.2.2.4 were provided.

Presentation Manager or Intelligent Terminal

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

Comment1: 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.

Comment2: This disclosure also requires that all data manipulation functions also be described. Within this disclosure, the purpose of such additional function(s) must be explained.

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

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 were 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 transaction s must be disclosed.

The percentage of Payment and Order-Status transaction s 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 1 shows the numerical quantities required by Clause 8.1.3.5 through 8.1.3.10.

Transaction Mix

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

Table 1 shows the mix of transaction types seen by the SUT during the reported measurement interval. Following table summarizes the data required for disclosure in section 3.5 through 3.11.

Table 1 Transaction Statistics

Statistic	Value	
New Order	Home warehouse order lines	99.00%
	Remote warehouse order lines	1.00%
	Rolled back transactions	0.98%
	Average items per order	10.01
Payment	Home warehouse payments	85.01%
	Remote warehouse payments	14.99%
	Accessed by last name	60.01%
Order Status	Accessed by last name	60.34%
Delivery	Skipped deliveries	0
Transaction Mix	New Order	44.86%
	Payment	43.05%
	Stock Level	4.02%
	Delivery	4.05%
	Order Status	4.02%

Queuing Mechanism

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

The client application processes submitted delivery transactions to named pipe delivery server software running on the client machines. There was a single delivery server with multiple execution threads running on each client machine. These delivery servers were responsible for processing deliveries queued to the named pipe and submitting them to the database server. The source code is listed in Appendix A.

Clause 3 : Transaction and System Properties Related Items

Transaction System Properties (ACID)

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

The TPC Benchmark™ C Standard Specification defines a set of transaction processing system properties that a system under test (SUT) must support during the execution of the benchmark. Those properties are Atomicity, Consistency, Isolation and Durability (ACID). This section quotes the specification definition of each of those properties and describes the tests done as specified and monitored by the auditor, to demonstrate compliance.

Atomicity Tests

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

Completed Transactions

Perform the Payment for randomly selected warehouse, district and customer (by customer number as specified in Clause 2.5.1.2) and verify that the records in the CUSTOMER, DISTRICT and WAREHOUSE tables have been changed appropriately.

The value of w_ytd, d_ytd, c_balance, c_ytd_payment and c_payment_cnt of a randomly selected warehouse, district, and customer were retrieved. The Payment transaction was executed on the same warehouse, district, and customer. The transaction was committed. The values w_ytd, d_ytd, c_balance, c_ytd_payment, and c_payment_cnt were retrieved again. It was verified that all values had been changed appropriately.

Aborted Transactions

Perform the Payment transaction for randomly selected warehouse, district and customer (by customer number as specified in Clause 2.5.1.2) and substitute a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that records in CUSTOMER, DISTRICT and WAREHOUSE tables have Not been changed.

The value of w_ytd, d_ytd, c_balance, c_ytd_payment and c_payment_cnt of randomly selected warehouse, district, and customer were retrieved. The Payment transaction was executed on the same warehouse, district, and customer. The transaction was rolled back. The values of w_ytd, d_ytd, c_balance, c_ytd_payment, c_payment_cnt were retrieved again. It was verified that none of the values had changed.

Consistency Tests

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 over 10 minutes and included a checkpoint under 3500 users (350 active warehouse) condition. The shell script was executed before and after the run. 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 level is obtained.

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

Case A was followed for Isolation Test 7.

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.

- *Permanent irrecoverable failure of any single durable medium containing database, ABTH files/tables, or recovery log data.*
- *Instantaneous interruption(system crash/system hang) in processing which requires system reboot to recover.*
- *Failure of all or part of memory(loss of contents)*

Loss of Memory and Loss of Log

Because the loss of power erases the contents of memory, both of instantaneous interruption and loss of memory were combined into a single test. Also loss of log was combined into the test.

The following steps were performed on a database of 2520 warehouses under the full load of users.

1. A sum of D_NEXT_O_ID of all rows in the district table was taken.
2. Full load of users were logged in to the database and running transactions for 5 minutes.
3. A checkpoint was initiated.
4. One disk drive of mirror paired drives holding the transaction log was removed.
An NT alert message was displayed and logged in the NT event log.
The benchmark run continued without interruption.
5. The running continued 2 minutes.
6. the system was powered off.
7. The RTE was shutted down.
8. The system was powered back up. SQL Server was restarted and automatically recovered.
9. A new count of D_NEXT_O_ID was taken.
10. This number was compared with the number of new orders reported by the RTE.

Loss of Data

Loss of data was demonstrated on a 10 Warehouse database for convenience. The standard driving mechanism was used to generate the transaction load of 100 users for the test. To demonstrate recovery from a permanent failure of durable media containing TPC-C tables, the following steps were performed. A fully scaled database would also pass this test.

1. A 10 Warehouse database was built having similar characteristics to the large database.
2. The database was backed up using SQL Server backup facilities.
3. A sum of D_NEXT_O_ID was taken.
4. 100 users were logged in to the database and running transactions.
5. One disk drive in the array was removed causing SQL Server error. SQL Server was shutted down.
6. SQL Server was restarted and a dump of the transaction log was taken.
7. The 10 Warehouse database was restored from backup.
8. The transaction log was restored and transactions rolled forward.
9. A new count of D_NEXT_O_ID was taken.
10. This number was compared with the number of new orders reported by the RTE.

Clause 4 : Scaling and Database Population Related Items

Initial Cardinality of Tables

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

The TPC-C database was originally built with 2520 warehouses.

Table 2 Number of Rows for Server

Table	Cardinality as benchmarked
Warehouse	2,520
Distinct	25,200
Customer	75,600,000
History	75,600,000
Orders	75,600,000
New Order	22,680,000
Order Line	756,002,500
Stock	252,000,000
Item	100,000
Deleted Warehouse Rows	0

Constant Value for the NURand function

The following values were used as constant value inputs to the NURand function for this benchmark.

C_LAST (Build) 123
C_LAST (RUN) 233

Distribution of Tables and Logs

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

Table 3 depicts the distribution of the database over the disks of the tested system.

Figure 1.1, 1.2 shows the disk configuration for measured and priced system.

Table 3 : Data Distribution

Disk Administrator Configuration						
Disk	Partition1	Partition2	Unused Space	HA#	LD#	Usage
0 8.46GB	C: NTFS 8.46GB			1	1	System SQL Server
1 84.56GB	E: (RAW) 39.06GB		(None)	2	1	Log (Mirrored)
2 458.93GB	F: (RAW) 21.00GB	K: (RAW) 37.11GB	45.50GB	3	1	Data MSSQL70_misc1 MSSQL70_cs1
3 458.93GB	G: (RAW) 21.00GB	L: (RAW) 37.11GB	400.83GB	4	1	Data MSSQL70_misc2 MSSQL70_cs2
4 458.93GB	H: (RAW) 21.00GB	M: (RAW) 37.11GB	400.83GB	5	1	Data MSSQL70_misc3 MSSQL70_cs3
5 458.93GB	I: (RAW) 21.00GB	N: (RAW) 37.11GB	400.83GB	6	1	Data MSSQL70_misc4 MSSQL70_cs4
6 458.93GB	J: (RAW) 21.00GB	O: (RAW) 37.11GB	400.83GB	7	1	Data MSSQL70_misc5 MSSQL70_cs5

Type of Database

A statement must be provided that describes:

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

Microsoft SQL Server, 2000, a relational database, was used in this benchmark. SQL Server stored procedures were used and invoked through DB-Library function calls embedded in C code.

Database Mapping

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

No partitioning or replication was used.

180-Days Space

Details of the 180 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.

The detail of 180-day space calculation is shown in Appendix D.

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 free space on the log file was queried using *DBCC sqlperf(logspace)*.
2. Transactions were run against the database with a full load of users.
3. The free space was again queried using *DBCC sqlperf(logspace)*.
4. The space used was calculated as the difference between the first and second query.
5. The number of NEW-ORDERS was verified from an RTE report covering the entire run.
6. The space used was divided by the number of NEW-ORDERS giving a spaceused per NEW-ORDER transaction.
7. The space used per transaction was multiplied by the measured tpmC rate times 480 minutes.

The results of the above steps yielded a requirement of 74.45 GB to sustain the log for 8 hours.

Space available on the transaction log volume was 84.56 GB (including mirror), indicating that enough storage was configured to sustain 8 hours of growth.

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

Clause 5 : Performance Metrics and Response Time Related Items

Throughput

Measured tpmC must be reported

Table 4 : Measured tpmC

31,764.37 tpmC

Response Times

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

Table 5 : Response Times (in seconds)

Type	Average	Maximum	90 th %
New-Order	0.23	5.17	0.30
Payment	0.15	4.73	0.19
Stock Level	0.95	5.43	1.49
Interactive Delivery	0.10	0.72	0.11
Deferred Delivery	0.31	3.56	0.52
Order Status	0.18	3.77	0.23
Menu	0.10	1.17	0.11

Keying and Think Times

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

Table 6 : Keying Times

Type	Minimum	Average	Maximum
New-Order	18.00	18.01	18.04
Payment	3.00	3.02	3.05
Stock Level	2.00	2.02	2.05
Interactive Delivery	2.00	2.02	2.04
Order Status	2.00	2.02	2.05

Table 7 : Think Times

Type	Minimum	Average	Maximum
New-Order	0.00	12.04	120.51
Payment	0.00	12.05	120.50
Stock Level	0.00	5.01	50.50
Interactive Delivery	0.00	5.06	50.50
Order Status	0.00	10.11	100.50

Response Time Frequency Distribution Curves and Other Graphs

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

Figure 2.1 : New Order Response Time Distribution

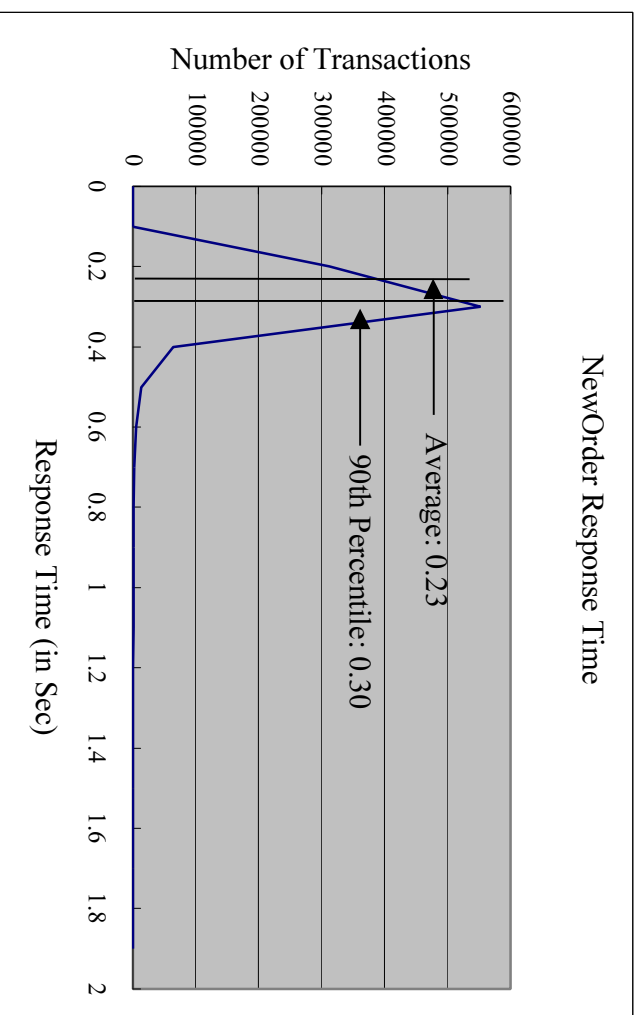


Figure 2.2 : Payment Response Time Distribution

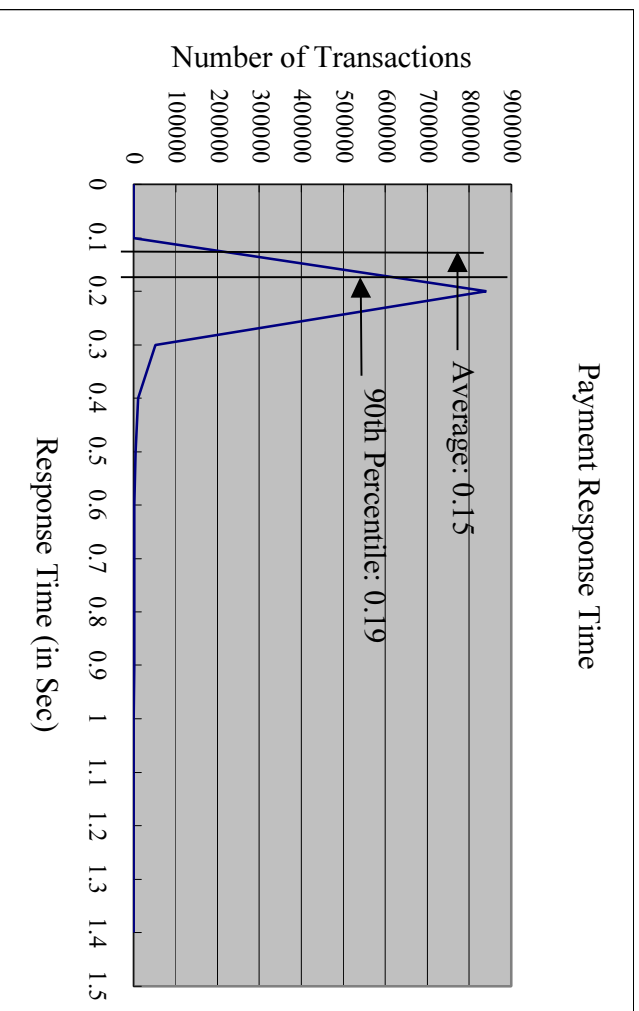


Figure 2.3 : Order Status Response Time Distribution

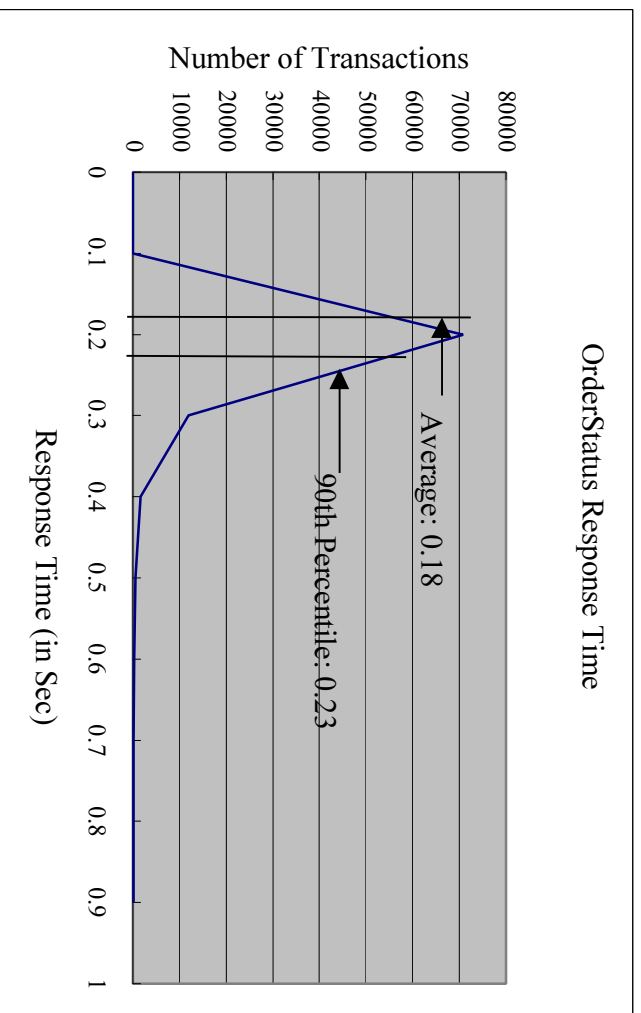


Figure 2.4 : Delivery Response Time Distribution

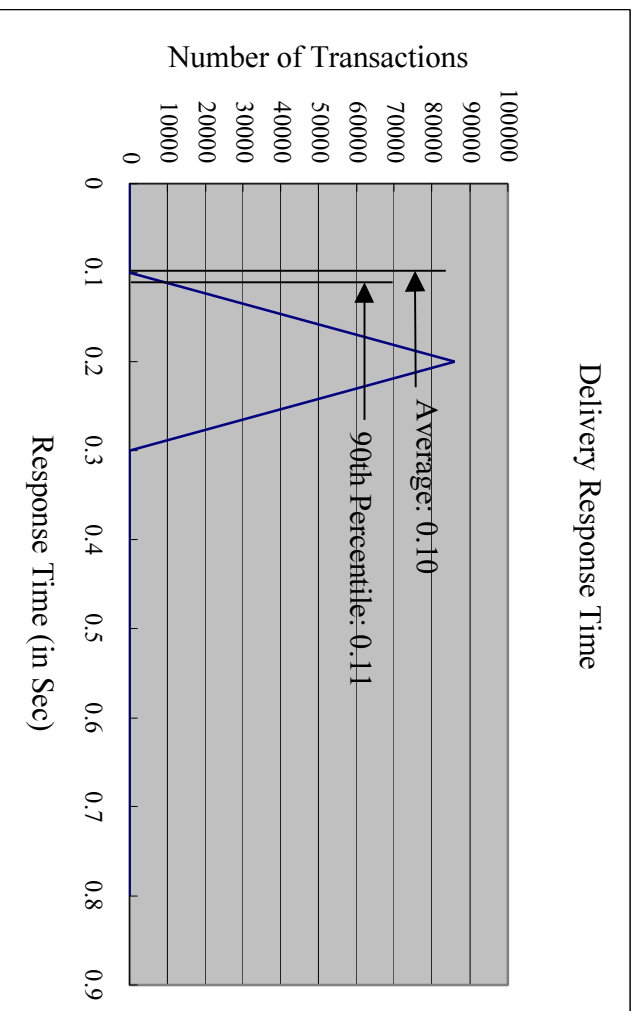
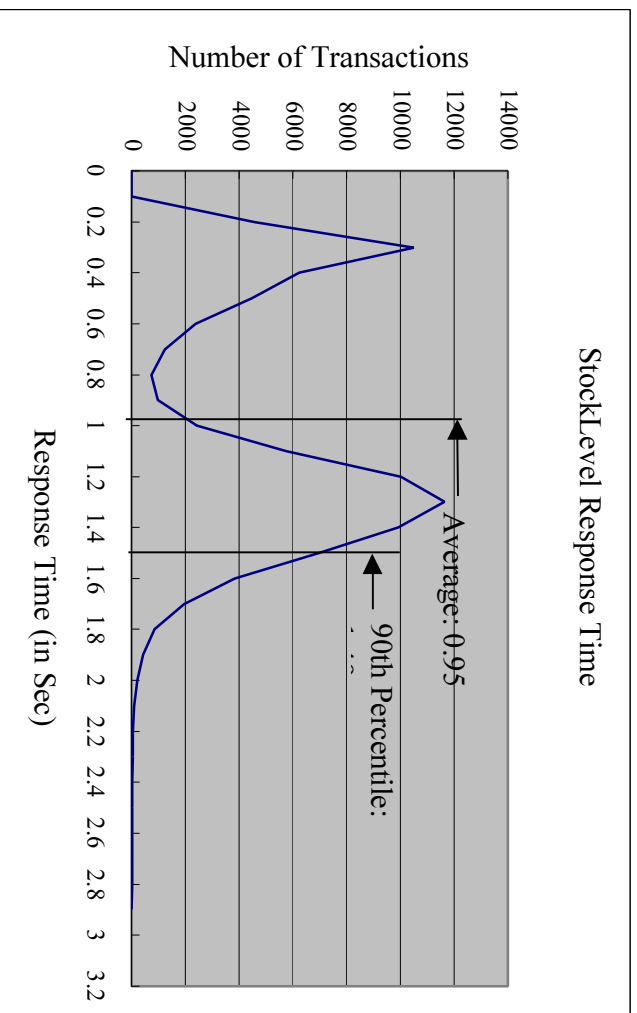


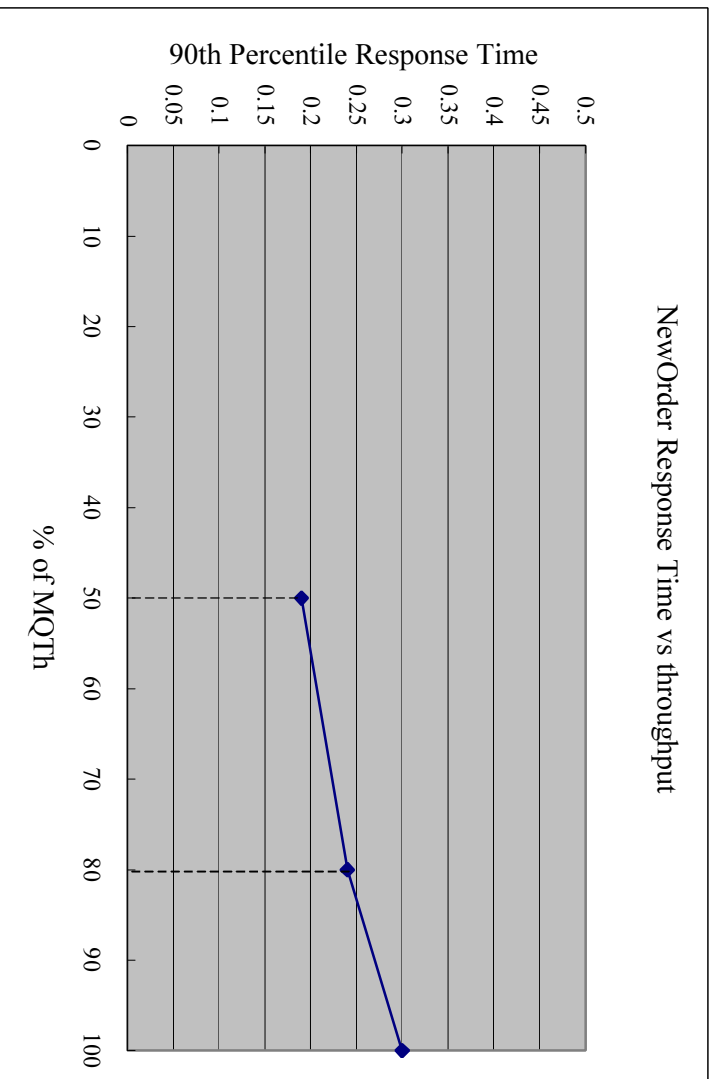
Figure 2.5 : Stock Level Response Time Distribution



Response time versus Throughput Performance Curve

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

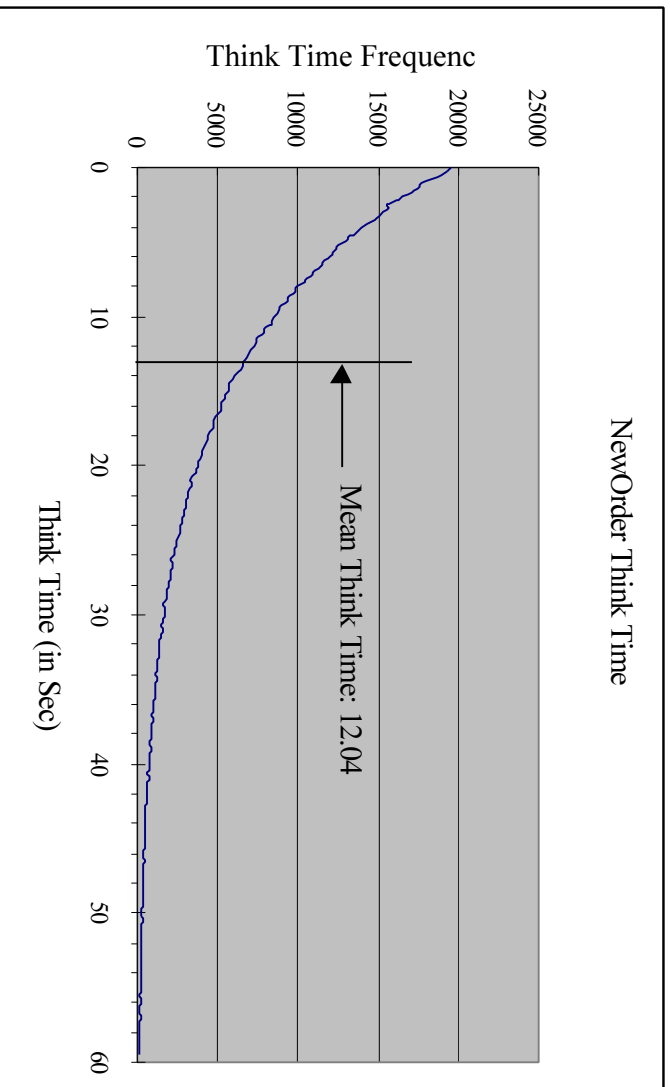
Figure 2.6 Response Time Performance vs. Throughput Curve



NEW-Order Think Time

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

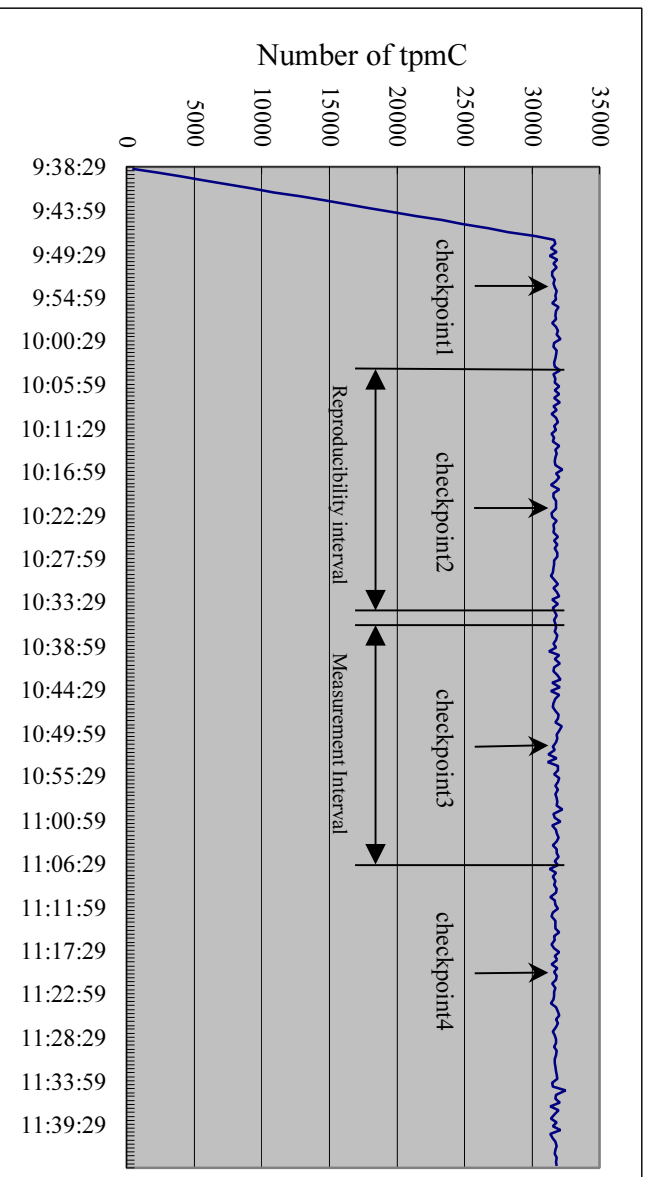
Figure 2.7 New-Order Think Time



New-Order Throughput vs. Elapsed Time

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

Figure 2.8 New Order Throughput vs. Time



Steady State

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

Steady state was confirmed by the throughput data collected during the run and graphed in Figure 2.8.

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 the maximum allowable value 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.

Reproducibility

A description of the method used to determine the reproducibility of the measurement results must be reported.

The reproducibility test result is taken from another, non-overlapping, measurement interval of the same duration as the reported interval. The throughput difference measured over that interval was within 0.13% of reported interval result.

Measurement Period Duration

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

The reported measured interval was exactly 30 minutes long.

Regulation of Transaction Mix

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

The RTE was given a weighted random distribution which 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 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.

Initial checkpoint was started 13 minutes after the start of ramp-up. Second checkpoint was started 30 minutes after the 1st checkpoint. And Third checkpoint was started 30 minutes after the 2nd checkpoint. The time from the start of the Measurement interval was 15.0 minutes after. In accord with Clause 5.5.22, there is no checkpoint within the "guard zones" 1800/4=450 seconds from the beginning and end of the measurement interval.

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

Descriptions of RTE

The RTE input parameters, code fragments, functions, etc. used to generate each transaction input field must be disclosed.

The RTE used was the Microsoft BenchCraft RTE System. The RTE input parameters are listed in Appendix C.

Emulated Components

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

AS configured for this test, the driver software emulates the traffic that would be observed from the users' PCs connected by Ethernet to the front-end clients using HTTP (HyperText Transfer Protocol) over TCP/IP. One tenth of a second (100 milli seconds) was added to each transaction time to compensate for the overhead of the Web browser.

Functional Diagrams and Detail of Driver System

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

The diagrams in figure 1.1 and 1.2 show the tested and priced benchmark configurations.

Network configurations and Driver system

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

Figure 1.1 and 1.2 in this report has the network configurations of both the tested system and the priced system.

The front-end clients were connected over one 100Mbps 100Base-T Ethernet segments to the back-end. Each front-end client were connected to the RTE over seven 10Mbps 10Base-T Ethernet segments.

The priced PCs are also connected using 10Mbps Ethernet to the front-end clients.

Network Bandwidth

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

The Ethernet used in the local area network (LAN) between the emulated terminals and the front-end system complies with the IEEE 802.3 standard and has a bandwidth of 10Mbps.

Operator Intervention

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

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

Clause 7 : Pricing Related Items

Hardware and Software Components

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

The total 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.

The detailed list of all hardware and software for the priced configuration is listed in the system pricing summary.

Availability

The committed delivery date for general availability (availability date) of products used in the price calculation must be reported. When the priced system included products with different availability dates, the reported availability date for the priced system must be the date at which all components are committed to be available. The single date must be reported on the first page of the Executive Summary. All availability dates whether for individual components or for the SUT as a whole, must be disclosed to a precision of one day.

All the components used in the priced system are currently available with the exception of:

- NEC Express5800/180Ra-7 will be available by August 15, 2000.
- NEC 3U Disk Expansion Unit(14 slots) ADR-0802-IN-00 will be available by September 29, 2000.
- Microsoft SQL Server 2000 will be available by August 1, 2000.

Throughput, and Price Performance

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

- Maximum Qualified Throughput 31,764.37 tpmC
- Price per tpmC : \$18.63 per tpmC
- Total 5-year cost of ownership

Country Specific Pricing

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

This system is being priced for the United States of America.

Usage Pricing

For any usage pricing, the sponsor must disclose:

- Usage level at which the component was priced.
- A statement of the company policy allowing such pricing.

None

Clause 9 : Audit Related Items

Auditor's Report

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

Next page contains the complete independent auditor's report by Francois Raab of Information Paradigm Inc. for the test described in this report.

Availability of the Full Disclosure Report

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

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

Transaction Processing Performance Council
c/o Shanley Public Relations
777 North First Street, Suite 6000
San Jose, CA 95112-6311
or your local NEC office.

Auditor's letter

INFO SIZING



Benchmark Sponsor: Eiichi Kenai
NEC Corporation
3rd Development Dept
3rd Computers Software Dept
Fuchu City Tokyo 183, Japan

June 15, 2000

I verified the TPC Benchmark™ C performance of the following Client Server configuration:

Platform: **NEC Express5800/180Ra-7 c/s**
Operating system: **Microsoft Windows 2000 Advanced Server**
Database Manager: **Microsoft SQL Server 2000 Enterprise Edition**
Transaction Manager: **Microsoft COM+**

The results were:

CPU's Speed	Memory	Disks	NewOrder 90% Response Time	tpmC
Server: NEC Express5800/180Ra-7				
4 x Pentium III Xeon (700 MHz)	8 GB Main 2 MB L2-cache/cpu	281 x 9 GB 10 x 18 GB	0.30 Seconds	31,764.37
Four (4) Clients: NEC Express5800/120Ld (Specification for each)				
2 x Pentium III Xeon (866 MHz)	512 MB Main 256 KB L2-cache/cpu	1 x 9 GB	n/a	n/a

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

- The database records were the proper size
- The database was properly scaled and populated

1373 North Franklin Street • Colorado Springs, CO 80903 • 2527 • Office: 719/473 -7555 • Fax: 719/47 3-7554

- The required ACID properties were met
- The transactions were correctly implemented
- Input data was generated according to the specified percentages
- The transaction cycle times included the required keying and think times
- The reported response times were correctly measured.
- All 90% response times were under the specified maximums
- At least 90% of all delivery transactions met the 80 Second completion time limit
- The reported measurement interval was 30 minutes (1800 seconds)
- The reported measurement interval was representative of steady state conditions
- One checkpoint was taken during the reported measurement interval
- The repeatability of the measured performance was verified
- The 180 day storage requirement was correctly computed
- The system pricing was verified for major components and maintenance

Additional Audit Notes:

The performance measurements were executed on an Express5800/140Hb system. This system has all the same active elements as the reported system, except that it is packaged for rack mount, rather than tower enclosure. It is my opinion that the performance of the priced configuration would be identical to the performance of the measured system.

Respectfully Yours,



François Raab
President

1373 North Franklin Street • Colorado Springs, CO 80903 • 2527 • Office: 719/473 -7555 • Fax: 719/473 -7554

Appendix A : Application Source Code

WEBCLNT.DSP

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

# TARGETTYPE "Win32 (x86) Application" 0x0101

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

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

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "\Release"
# PROP BASE Intermediate_Dir "\Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir "\Release"
# PROP Intermediate_Dir "\Release"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
"\_WINDOWS" /YX /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "\_WINDOWS"
/YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:i386

!ENDIF

# Begin Target

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

```
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:i386

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "\Debug"
# PROP BASE Intermediate_Dir "\Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir "\Debug"
# PROP Intermediate_Dir "\Debug"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D
"\_WINDOWS" /YX /c
# ADD CPP /nologo /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D
"\_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:i386

!ENDIF

# Begin Target

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

WEBCLNT.DSW

```
Microsoft Developer Studio Workspace File, FormatVersion 6.00
# WARNING: DO NOT EDIT OR DELETE THIS WORKSPACE FILE!

#####

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

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

#####

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

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

```
Package=<4>
{{{
}}}

#####

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

Package=<5>
{{{
}}}

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

#####

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

Package=<5>
{{{
}}}

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

#####
```

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

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

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

```
#####
#####
```

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

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

```
Package=<4>
{{{
}}}
```

```
#####
#####
```

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

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

```
Package=<4>
{{{
}}}
```

```
#####
#####
```

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

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

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

```
#####
#####
```

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

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

```
Package=<4>
{{{
}}}
```

```
#####
#####
```

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

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

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

```
#####
#####
```

Global:

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

```
Package=<3>
{{{
}}}
```

```
#####
#####
```

INSTALL.DSP

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

```
# TARGETTYPE "Win32 (x86) Application" 0x0101
```

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

```
# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
```

```
MTL=midl.exe
RSC=rc.exe
```

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

```
# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir ".\Release"
# PROP BASE Intermediate_Dir ".\Release"
# PROP BASE Target_Dir "."
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir "."
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
" _WINDOWS" /YX /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:i386
# ADD LINK32 version.lib comctl32.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib
odbc32.lib odbccp32.lib /nologo /subsystem:windows /machine:i386
/out:". \bin\install.exe"
```

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

```
# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir ".\Debug"
# PROP BASE Intermediate_Dir ".\Debug"
# PROP BASE Target_Dir "."
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir "."
# ADD BASE CPP /nobgo /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D
" _WINDOWS" /YX /c
# ADD CPP /nologo /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D
" _WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:i386
# ADD LINK32 version.lib comctl32.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib
odbc32.lib odbccp32.lib /nologo /subsystem:windows /debug /machine:i386
/out:". \bin\install.exe"
```

```
!ENDIF
```

```
# Begin Target
```

```

# Name "install - Win32 Release"
# Name "install - Win32 Debug"
# Begin Group "Source Files"

# PROP Default_Filter "cpp;c;cxx;rc;def;r;odl;hpp;bat;for;f90"
# Begin Source File

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

SOURCE=.\src\install.rc
# ADD BASE RSC /I 0x409 /i "src"
# ADD RSC /I 0x409 /i "src" /i ".\src"
# End Source File
# Begin Source File

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

# PROP Default_Filter "h;hpp;hxx;hm;inl;fi;fd"
# End Group
# Begin Group "Resource Files"

# PROP Default_Filter "ico;cur;bmp;dlg;rc2;rct;bin;cnt;rtf;gif;jpg;jpeg;jpe"
# Begin Source File

SOURCE=.\SRC\ICON1.ICO
# End Source File
# Begin Source File

SOURCE=.\SRC\ICON2.ICO
# End Source File
# End Group
# Begin Source File

SOURCE=.\SRC\LICENSE.TXT
# End Source File
# Begin Source File

SOURCE=.\isapi_dll\bin\tpcc.dll
# End Source File
# Begin Source File

SOURCE=.\tm_com_dll\bin\tpcc_com.dll
# End Source File
# Begin Source File

SOURCE=.\tpcc_com_all\bin\tpcc_com_all.dll
# End Source File
# Begin Source File

SOURCE=.\tpcc_com_ps\bin\tpcc_com_ps.dll
# End Source File
# Begin Source File

SOURCE=.\ldb_dblib_dll\bin\tpcc_dblib.dll
# End Source File
# Begin Source File

SOURCE=.\ldb_odbc_dll\bin\tpcc_odbc.dll
# End Source File
# Begin Source File

SOURCE=.\tm_tuxedo_dll\bin\tpcc_tuxedo.dll
# End Source File
# Begin Source File

SOURCE=.\tuxapp\bin\tuxapp.exe

```

```

# End Source File
# End Target
# End Project

```

ISAPI_DLL.DSP

```

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

# TARGETTYPE "Win32 (x86) DynamicLink Library" 0x0102

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

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

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
"\WINDOWS" /YX /FD /c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "NDEBUG" /D "WIN32" /D
"\WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /I 0x409 /d "NDEBUG"
# ADD RSC /I 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:1386
# ADD

```

```

LINK32 ..\common\txnlgl\lib\release\rtetime.lib ..\common\txnlgl\lib\release\spi
nlock.lib ..\common\txnlgl\lib\release\error.lib ..\common\txnlgl\lib\release\txnlo
g.lib wsock32.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:1386 /out:". \bin\tpcc.dll"

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG"
/D "\WINDOWS" /YX /FD /c
# ADD CPP /nologo /MDd /W3 /GX /ZI /Od /D "_DEBUG" /D "WIN32" /D
"\WINDOWS" /FR /YX /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /I 0x409 /d "_DEBUG"
# ADD RSC /I 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:1386 /pdbtype:sept
# ADD
LINK32 ..\common\txnlgl\lib\debug\rtetime.lib ..\common\txnlgl\lib\debug\spinlo
ck.lib ..\common\txnlgl\lib\debug\error.lib ..\common\txnlgl\lib\debug\txnlgl.lib
wsock32.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:1386 /out:". \bin\tpcc.dll"
/pdbtype:sept
# SUBTRACT LINK32 /profile /pdb:none

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "isapi_dll"
# PROP BASE Intermediate_Dir "isapi_dll"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /GX /ZI /Od /D "_DEBUG" /D "WIN32" /D
"\WINDOWS" /FR /YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /GX /ZI /O2 /D "NDEBUG" /D "ICECAP" /D
"WIN32" /D "\WINDOWS" /FR /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /I 0x409 /d "_DEBUG"
# ADD RSC /I 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:1386 /out:". \bin\tpcc.dll"

```

```

/pdbtype:sept
# SUBTRACT BASE LINK32 /profile /pdb:none
# ADD LINK32
icap.lib ..\common\txnolog\lib\release\rtetime.lib ..\common\txnolog\lib\release\spinc
lock.lib ..\common\txnolog\lib\release\error.lib ..\common\txnolog\lib\release\txnolog.
lib wsock32.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:i386 /out:". \bin\tpcc.dll"
/pdbtype:sept
# SUBTRACT LINK32 /profile /pdb:none /map

!ENDIF

# Begin Target

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

DB_DBLIB_DLL.DSP

```

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

# TARGETTYPE "Win32 (x86) DynamicLink Library" 0x0102

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

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

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
"_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
"_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib
/nologo /subsystem:windows /dll /machine:i386
# ADD LINK32 ntdll.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /machine:i386 /out:". \bin\tpcc_dblib.dll"

```

```

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MD /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG"
/D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D
"_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib
/nologo /subsystem:windows /dll /debug /machine:i386 /pdbtype:sept
# ADD LINK32 ntdll.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /debug /machine:i386 /out:". \bin\tpcc_dblib.dll"
/pdbtype:sept

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "db_dblib"
# PROP BASE Intermediate_Dir "db_dblib"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D
"NDEBUG" /D "_WINDOWS" /YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /Gm /GX /ZI /O2 /D "WIN32" /D "NDEBUG" /D
"_WINDOWS" /D "ICECAP" /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 ntdll.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /debug /machine:i386 /out:". \bin\tpcc_dblib.dll"
/pdbtype:sept

!ENDIF

# Begin Target

```

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

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

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

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

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

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

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

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

TM_COM_DLL.DSP

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

```
# TARGETTYPE "Win32 (x86) DynamicLink Library" 0x0102
```

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

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

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

```
# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
"_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
"_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:1386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:1386 /out:".bin\tpcc_com.dll"

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

```
# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG"
/D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D
"_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:1386 /pdbtype:sept
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:1386 /out:".bin\tpcc_com.dll"
/pdbtype:sept
```

```
!ENDIF
```

```
# Begin Target
```

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

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

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

TPCC_COM_ALL.DSP

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

```
# TARGETTYPE "Win32 (x86) DynamicLink Library" 0x0102
```

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

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

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

```
# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
"_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
"_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
```

```
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /system:windows /dll /machine:1386
# ADD LINK32 .ldb_dblib_dll\bin\tpcc_dblib.lib ..ldb_odbc_dll\bin\tpcc_odbc.lib
kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib
shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /machine:1386
```

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

```
# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG"
/D " _WINDOWS" /YX /FD /c
# ADD CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D
" _WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D " _DEBUG" /mktplib203 /o "NUL" /win32
# ADD MTL /nologo /D " _DEBUG" /mktplib203 /o "NUL" /win32
# ADD BASE RSC /I 0x409 /d " _DEBUG"
# ADD RSC /I 0x409 /d " _DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /system:windows /dll /debug /machine:1386 /pdbtype:sept
# ADD LINK32 .ldb_dblib_dll\bin\tpcc_dblib.lib ..ldb_odbc_dll\bin\tpcc_odbc.lib
kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib
shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /debug /machine:1386 /pdbtype:sept
```

```
!ENDIF
```

```
# Begin Target
```

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

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

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

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

```
SOURCE=.\src\tpcc_com_all.idl
```

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

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

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

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

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

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

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

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

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

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

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

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

```
!ENDIF
```

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

```
# PROP Default_Filter ""
# Begin Source File
```

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

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

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

TPCC_COM_PS.DSP

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

```
# TARGETTYPE "Win32 (x86) Application" 0x0101
```

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

```
!MESSAGE NMAKE /f "tpcc_com_ps.mak" CFG="tpcc_com_ps - Win32
Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpcc_com_ps - Win32 Release" (based on "Win32 (x86)
Application")
!MESSAGE "tpcc_com_ps - Win32 Debug" (based on "Win32 (x86)
Application")
!MESSAGE
```

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

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

```
# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
" _WINDOWS" /YX /FD /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D
_WIN32_WINNT=0x0400 /D "REGISTER_PROXY_DLL" /FD /c
# SUBTRACT CPP /YX
# ADD BASE MTL /nologo /D "NDEBUG" /mktplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktplib203 /o "NUL" /win32
# ADD BASE RSC /I 0x409 /d "NDEBUG"
# ADD RSC /I 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /system:windows /machine:1386
# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpctr4.lib oleaut32.lib uuid.lib
/nologo /entry:"DllMain" /system:windows /dll /pdb:none /machine:1386
/def:"..\src\tpcc_com_ps.def"
# Begin Custom Build - Copying tpcc_com_ps.h
InputPath=.\bin\tpcc_com_ps.dll
SOURCE="$(InputPath)"
```

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

```
# End Custom Build
```

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

```
# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
```

```

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

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

# End Custom Build

!ENDIF

# Begin Target

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

# PROP Default_Filter ""
# Begin Source File

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

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

SOURCE=. \src\tpcc_com_ps.idl

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

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

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

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

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

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

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

```

```

# End Custom Build

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

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

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

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

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

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

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

!ENDIF

# End Source File
# Begin Source File

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

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

```

install\src\install.c

```

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

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

#include "resource.h"

#define WM_INITTEXT WM_USER+100

```

```

HICON hIcon;
HINSTANCE hInst;

DWORD versionExeMS;
DWORD versionExeLS;
DWORD versionExeMM;
DWORD versionDIIMS;
DWORD versionDIILS;

// TPC-C registry settings
TPCCREGISTRYDATA Reg;

static int iPoolThreadLimit;
static int iThreadTimeout;
static int iListenBackLog;
static int iAcceptExOutstanding;

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

BOOL CALLBACK LicenseDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam);
BOOL CALLBACK UpdatedDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam);
BOOL CALLBACK MainDlgProc(HWND hwnd, UINT uMsg, WPARAM
wParam, LPARAM lParam);
BOOL CALLBACK CopyDlgProc(HWND hwnd, UINT uMsg, WPARAM
wParam, LPARAM lParam);
static void ProcessOK(HWND hwnd, char
*szDllPath);
static void ReadRegistrySettings(void);
static void WriteRegistrySettings(char *szDllPath);
static BOOL RegisterDLL(char *szFileName);
static int CopyFiles(HWND hDlg,
char *szDllPath);
static BOOL GetInstallPath(char *szDllPath);
static void GetVersionInfo(char *szDllPath, char
*szExePath);
static BOOL CheckWWWWebService(void);
static BOOL StartWWWWebService(void);
static BOOL StopWWWWebService(void);
static void UpdateDialog(HWND hDlg);

BOOL install_com(char *szDllPath);

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

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

hInst = hInstance;

InitCommonControls();

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

iRc = DialogBox(hInstance, MAKEINTRESOURCE(IDD_DIALOG4),
GetDesktopWindow(), LicenseDlgProc);
if ( iRc )
{
iRc = DialogBox(hInstance,
MAKEINTRESOURCE(IDD_DIALOG1), GetDesktopWindow(), MainDlgProc);
if ( iRc )
{
DialogBoxParam(hInstance,
MAKEINTRESOURCE(IDD_DIALOG2), GetDesktopWindow(), UpdatedDlgProc,
(LPARAM)iRc);
}
}
}

```

```

    }
    DestroyIcon(hIcon);
    return 0;
}

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

    switch(uMsg)
    {
        case WM_INITDIALOG:
            hFont = CreateFont(-12, 0, 0, 0, 400, 0,
0, 0, 0, 0, 0, 0, "Arial");
            SendMessage( GetDlgItem(hwnd,
IDR_LICENSE1), WM_SETFONT, (WPARAM)hFont, MAKELPARAM(0, 0) );
            PostMessage(hwnd, WM_INITTEXT,
(WPARAM)0, (LPARAM)0);
            return TRUE;
        case WM_INITTEXT:
            hResInfo = FindResource(hInst,
MAKEINTRESOURCE(IDR_LICENSE1), "LICENSE");
            dwSize = SizeofResource(hInst,
hResInfo);
            hRes = LoadResource(hInst, hResInfo );
            pSrc = (BYTE *)LockResource(hRes);
            pDst = (unsigned char
*)malloc(dwSize+1);
            if ( pDst )
            {
                memcpy(pDst, pSrc,
dwSize);
                pDst[dwSize] = 0;
                SetDlgItemText(hwnd,
IDC_LICENSE, (const char *)pDst);
            }
            else
                SetDlgItemText(hwnd,
IDC_LICENSE, (const char *)pSrc);
            return TRUE;
        case WM_DESTROY:
            DeleteObject(hFont);
            return TRUE;
        case WM_COMMAND:
            if ( wParam == IDOK )
                EndDialog(hwnd, TRUE);
            if ( wParam == IDCANCEL )
                EndDialog(hwnd, FALSE);
            default:
                break;
    }
    return FALSE;
}

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

```

```

                SetDlgItemText(hwnd, IDC_RESULTS, "TPGC Web Client
Installed");
                break;
            }
            return TRUE;
        case WM_COMMAND:
            if ( wParam == IDOK )
                EndDialog(hwnd, TRUE);
            default:
                break;
    }
    return FALSE;
}

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

    switch(uMsg)
    {
        case WM_INITDIALOG:
            GlobalMemoryStatus(&memoryStatus);
            iMaxPhysicalMemory =
(memoryStatus.dwTotalPhys/ 1048576);
            if ( GetInstallPath(szDllPath) )
            {
                MessageBox(hwnd, "Error
internet service inetsrv is not installed.", NULL, MB_ICONSTOP | MB_OK);
                EndDialog(hwnd, FALSE);
                return TRUE;
            }
            // set default values
            ZeroMemory( &Reg, sizeof(Reg) );
            Reg.dwNumberOfDeliveryThreads = 4;
            Reg.dwMaxConnections = 100;
            Reg.dwMaxPendingDeliveries
= 100;
            Reg.eDB_Protocol = DBLIB;
            Reg.eTxnMon = None;
            strcpy(Reg.szDbServer,
"" );
            strcpy(Reg.szDbName,
"tpcc");
            strcpy(Reg.szDbUser,
"sa");
            strcpy(Reg.szDbPassword,
"" );
            iPoolThreadLimit =
iThreadTimeout = 86400;
            iListenBackLog = 15;
            iAcceptExOutstanding = 40;
            ReadTPCCRegistrySettings( &Reg );
            ReadRegistrySettings();
            GetModuleFileName(hInst, szExePath,
sizeof(szExePath));
            GetVersionInfo(szDllPath, szExePath);
            wsprintf(szTmp,

```

```

"Version %d.%2.2d.%3.3d", versionExeMS, versionExeMM, versionExeLS);
            SetDlgItemText(hwnd, IDC_VERSION,
szTmp);
            SetDlgItemText(hwnd, IDC_PATH,
szDllPath);
            SetDlgItemText(hwnd,
ED_DB_SERVER, Reg.szDbServer);
            SetDlgItemText(hwnd,
ED_DB_USER_ID, Reg.szDbUser);
            SetDlgItemText(hwnd,
ED_DB_PASSWORD, Reg.szDbPassword);
            SetDlgItemText(hwnd, ED_DB_NAME,
Reg.szDbName);
            SetDlgItemInt(hwnd, ED_THREADS,
Reg.dwNumberOfDeliveryThreads, FALSE);
            SetDlgItemInt(hwnd,
ED_MAXCONNECTION, Reg.dwMaxConnections, FALSE);
            SetDlgItemInt(hwnd,
ED_MAXDELIVERIES, Reg.dwMaxPendingDeliveries, FALSE);
            SetDlgItemInt(hwnd,
ED_IIS_MAX_THREAD_POOL_LIMIT, iPoolThreadLimit, FALSE);
            SetDlgItemInt(hwnd,
ED_IIS_THREAD_TIMEOUT, iThreadTimeout, FALSE);
            SetDlgItemInt(hwnd,
ED_IIS_LISTEN_BACKLOG, iListenBackLog, FALSE);
            SetDlgItemInt(hwnd,
ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE, iAcceptExOutstanding,
FALSE);
            CheckDlgButton(hwnd, IDC_DBLIB, 0);
            CheckDlgButton(hwnd, IDC_ODBC, 0);
            if ( Reg.eDB_Protocol == DBLIB )
                CheckDlgButton(hwnd,
IDC_DBLIB, 1);
            else
                CheckDlgButton(hwnd,
IDC_ODBC, 1);
            // check OS version level for COM.
            Must be at least Windows 2000
            VI.dwOSVersionInfoSize = sizeof(VI);
            GetVersionEx( &VI );
            if (VI.dwMajorVersion < 5)
            {
                HWND hDlg =
GetDlgItem( hwnd, IDC_TM_MTS );
                EnableWindow( hDlg, 0 );
                // disable COM option
                if (Reg.eTxnMon == COM)
                    Reg.eTxnMon = None;
            }
            CheckDlgButton(hwnd, IDC_TM_NONE,
0);
            CheckDlgButton(hwnd,
IDC_TM_TUXEDO, 0);
            CheckDlgButton(hwnd, IDC_TM_MTS,
0);
            CheckDlgButton(hwnd,
IDC_TM_ENCINA, 0);
            switch (Reg.eTxnMon)
            {
                case None:
                    CheckDlgButton(hwnd,
IDC_TM_NONE, 1);
                    break;
                case TUXEDO:
                    CheckDlgButton(hwnd,

```



```

IDC_TM_TUXEDO, 1);
                                break;
                                case ENCINA:
                                CheckDlgButton(hwnd,
IDC_TM_ENCINA, 1);
                                break;
                                case COM:
                                CheckDlgButton(hwnd,
IDC_TM_MTS, 1);
                                break;
                                }
                                return TRUE;
                                case WM_PAINT:
                                if ( !IsIconic(hwnd) )
                                {
                                        BeginPaint(hwnd, &ps);
                                        DrawIcon(ps.hdc, 0, 0,
                                        hIcon);
                                        EndPaint(hwnd, &ps);
                                        return TRUE;
                                }
                                break;
                                case WM_COMMAND:
                                if ( HIWORD(wParam) ==
BN_CLICKED )
                                {
                                        switch( LOWORD(wParam) )
                                        {
                                                case
IDC_DBLIB:
                                                return TRUE;
                                                case
IDC_ODBC:
                                                return TRUE;
                                                ProcessOK(hwnd, szDllPath);
                                                return TRUE;
                                                IDCANCEL:
                                                EndDialog(hwnd, FALSE);
                                                return TRUE;
                                                return FALSE;
                                                }
                                }
                                default:
                                break;
                                }
                                return FALSE;
}

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

        char        szFullName[256];
        char        szErrMsg[128];

        // read settings from dialog
        Reg.dwNumberOfDeliveryThreads = GetDlgItemInt(hwnd,

```

```

ED_THREADS, &d, FALSE);
        Reg.dwMaxConnections = GetDlgItemInt(hwnd,
ED_MAXCONNECTION, &d, FALSE);
        Reg.dwMaxPendingDeliveries = GetDlgItemInt(hwnd,
ED_MAXDELIVERIES, &d, FALSE);

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

        if ( !IsDlgButtonChecked(hwnd, IDC_DBLIB) )
        {
                Reg.eDB_Protocol = DBLIB;
                rc = 1;
        }
        else if ( !IsDlgButtonChecked(hwnd, IDC_ODBC) )
        {
                Reg.eDB_Protocol = ODBC;
                rc = 2;
        }

        if ( !IsDlgButtonChecked(hwnd, IDC_TM_NONE) )
                Reg.eTxnMon = None;
        else if ( !IsDlgButtonChecked(hwnd, IDC_TM_TUXEDO) )
                Reg.eTxnMon = TUXEDO;
        else if ( !IsDlgButtonChecked(hwnd, IDC_TM_MTS) )
                Reg.eTxnMon = COM;
        else if ( !IsDlgButtonChecked(hwnd, IDC_TM_ENCINA) )
                Reg.eTxnMon = ENCINA;

        iPoolThreadLimit = GetDlgItemInt(hwnd,
ED_IIS_MAX_THREAD_POOL_LIMIT, &d, FALSE);
        iThreadTimeout = GetDlgItemInt(hwnd,
ED_IIS_THREAD_TIMEOUT, &d, FALSE);
        iListenBackLog = GetDlgItemInt(hwnd,
ED_IIS_LISTEN_BACKLOG, &d, FALSE);
        iAcceptExOutstanding = GetDlgItemInt(hwnd,
ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE, &d, FALSE);

        ShowWindow(hwnd, SW_HIDE);
        hDlg = CreateDialog(hInst, MAKEINTRESOURCE(IDD_DIALOG3),
hwnd, CopyDlgProc);
        ShowWindow(hDlg, SW_SHOWNA);
        UpdateDialog(hDlg);

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

                EndDialog(hwnd, 0);
                return;
        }

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

        UpdateDialog(hDlg);
        WriteRegistrySettings(szDllPath);

```

```

// register com proxy stub
strcpy(szFullName, szDllPath);
strcat(szFullName, "tpcc_com_ps.dll");
if (!RegisterDLL(szFullName))
{
        ShowWindow(hwnd, SW_SHOWNA);
        DestroyWindow(hDlg);
        strcpy( szErrMsg, "Error occurred when registering " );
        strcat( szErrMsg, szFullName );
        MessageBox(hwnd, szErrMsg, NULL, MB_ICONSTOP
| MB_OK);

        EndDialog(hwnd, 0);
        return;
}

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

        if (install_com(szDllPath))
        {
                ShowWindow(hwnd, SW_SHOWNA);
                DestroyWindow(hDlg);
                strcpy( szErrMsg, "Error occurred when
configuring COM settings." );
                MessageBox(hwnd, szErrMsg, NULL,
MB_ICONSTOP | MB_OK);

                EndDialog(hwnd, 0);
                return;
        }
        Sleep(100);

        ShowWindow(hwnd, SW_SHOWNA);
        DestroyWindow(hDlg);

        EndDialog(hwnd, rc);
        return;
}

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

        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\CurrentControlSet\Services\inetinfo\Parameters", 0, KEY_READ,
&hKey) == ERROR_SUCCESS )
        {
                size = sizeof(iPoolThreadLimit);
                if ( RegQueryValueEx(hKey, "PoolThreadLimit", 0,
&type, (char *)&iPoolThreadLimit, &size) == ERROR_SUCCESS )
                        if ( !iPoolThreadLimit )
                                iPoolThreadLimit =
iMaxPhysicalMemory * 2;

                size = sizeof(iThreadTimeout);
                if ( RegQueryValueEx(hKey, "ThreadTimeout", 0,
&type, (char *)&iThreadTimeout, &size) == ERROR_SUCCESS )
                        if ( !iThreadTimeout )
                                iThreadTimeout = 86400;

                size = sizeof(iListenBackLog);
                if ( RegQueryValueEx(hKey, "ListenBackLog", 0,

```

```

&type, (char *)&iListenBackLog, &size) == ERROR_SUCCESS )
    if ( !iListenBackLog )
        iListenBackLog = 15;

    RegCloseKey(hKey);
}

if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\CurrentControlSet\Services\inetinfo\Parameters", 0, KEY_READ,
&hKey) == ERROR_SUCCESS )
{
    size = sizeof(iAcceptExOutstanding);
    if ( RegQueryValueEx(hKey, "AcceptExOutstanding",
0, &type, (char *)&iAcceptExOutstanding, &size) == ERROR_SUCCESS )
        if ( !iAcceptExOutstanding )
            iAcceptExOutstanding =
40;

    RegCloseKey(hKey);
}

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

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

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

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

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

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

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

        RegFlushKey(hKey);
        RegCloseKey(hKey);
}

```

```

}

if ( (iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\CurrentControlSet\Services\inetinfo\Parameters", 0, NULL,
REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, NULL, &hKey,
&dwDisposition)) == ERROR_SUCCESS )
{
    RegSetValueEx(hKey, "PoolThreadLimit", 0,
REG_DWORD, (char *)&iPoolThreadLimit, sizeof(iPoolThreadLimit));
    RegSetValueEx(hKey, "ThreadTimeout", 0,
REG_DWORD, (char *)&iThreadTimeout, sizeof(iThreadTimeout));
    RegSetValueEx(hKey, "ListenBackLog", 0,
REG_DWORD, (char *)&iListenBackLog, sizeof(iListenBackLog));

    RegFlushKey(hKey);
    RegCloseKey(hKey);
}

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

    RegFlushKey(hKey);
    RegCloseKey(hKey);
}

return;

BOOL CALLBACK CopyDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam,
LPARAM lParam)
{
    if ( uMsg == WM_INITDIALOG )
    {
        SendDlgItemMessage(hwnd, IDC_PROGRESS1,
PBM_SETRANGE, 0, MAKELPARAM(0, 15));
        SendDlgItemMessage(hwnd, IDC_PROGRESS1,
PBM_SETSTEP, (WPARAM)1, 0);
        return TRUE;
    }
    return FALSE;
}

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

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

BOOL FileFromResource( char *szResourceName, int iResourceId, char
*szDllPath, char *szFileName )
{
    HGLOBAL    hDLL;
    HRSRC     hResInfo;
    HANDLE     hFile;

```

```

    DWORD      dwSize;
    BYTE       *pSrc;
    DWORD      d;
    char       szFullName[256];

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

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

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

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

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

    CloseHandle(hFile);

    UnlockResource(hDLL);
    FreeResource(hDLL);
    return TRUE;
}

static int CopyFiles(HWND hDlg, char *szDllPath)
{
    BOOL      bSvcRunning;

    bSvcRunning = CheckWWWWebService();
    if ( bSvcRunning )
    {
        SetDlgItemText(hDlg, IDC_STATUS, "Stopping Web
Service.");
        SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

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

    SetDlgItemText(hDlg, IDC_STATUS, "Copying Files...");
    SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0,
0);
    UpdateDialog(hDlg);

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

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

```

```

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

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

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

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

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

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

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

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

return 1;
}

```

```

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

    szDllPath[0] = 0;
    bRc = TRUE;
    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\Virtual Roots", 0,
KEY_ALL_ACCESS, &hKey) == ERROR_SUCCESS )
    {
        sv = sizeof(szData);
        iRc = RegQueryValueEx( hKey, "/", NULL, NULL,
szData, &sv ); // used by IIS 3.0
        if (iRc == ERROR_FILE_NOT_FOUND)
            iRc = RegQueryValueEx( hKey, "/",
NULL, NULL, szData, &sv ); // used by IIS 4.0
        if (iRc == ERROR_SUCCESS)
        {
            bRc = FALSE;
            strcpy(szDllPath, szData);
            if ( (ptr = strchr(szDllPath, ','))
                *ptr = 0;

            len = strlen(szDllPath);
            if ( szDllPath[len-1] != '\\')
            {
                szDllPath[len] = '\\';
                szDllPath[len+1] = 0;
            }
        }
        RegCloseKey(hKey);
    }
    return bRc;
}

static void GetVersionInfo(char *szDLLPath, char*szExePath)
{
    DWORD   d;
    DWORD   dwSize;
    DWORD   dwBytes;
    char    *vs;
    VS_FIXEDFILEINFO *vs;

    versionDIIMS = 0;
    versionDIILS = 0;
    if ( _access(szDLLPath, 00) == 0 )
    {
        dwSize = GetFileVersionInfoSize(szDLLPath, &d);
        if ( dwSize )
        {
            ptr = (char *)malloc(dwSize);
            GetFileVersionInfo(szDLLPath, 0,
dwSize, ptr);
            VerQueryValue(ptr, "1", &vs, &dwBytes);
            versionDIIMS = vs->dwProductVersionMS;
            versionDIILS = vs->dwProductVersionLS;
            free(ptr);

            versionExeMS = 0x7FFF;
            versionExeLS = 0x7FFF;
        }
    }
}

```

```

dwSize = GetFileVersionInfoSize(szExePath, &d);
if ( dwSize )
{
    ptr = (char *)malloc(dwSize);
    GetFileVersionInfo(szExePath, 0, dwSize, ptr);
    VerQueryValue(ptr, "1", &vs, &dwBytes);

    versionExeMS = vs->dwProductVersionMS;
    versionExeLS = LOWORD(vs->dwProductVersionLS);
    versionExeMM = HIWORD(vs->dwProductVersionLS);
    free(ptr);
}
return;

static BOOL CheckWWWWebService(void)
{
    SC_HANDLE schSCManager;
    SC_HANDLE schService;
    SERVICE_STATUS ssStatus;

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

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

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

    CloseServiceHandle(schService);
    return TRUE;
}

ServiceNotRunning:
{
    CloseServiceHandle(schService);
    return FALSE;
}

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

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

    if (! StartService(schService, 0, NULL) )
        goto StartWWWWebErr;
    //start Service pending, Check the status until the service is
running.
    if (! QueryServiceStatus(schService, &ssStatus) )
        goto StartWWWWebErr;
    while( ssStatus.dwCurrentState != SERVICE_RUNNING)
    {

```

```

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

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

        CloseServiceHandle(schService);
        return TRUE;

StartWWWWebErr:
        CloseServiceHandle(schService);
        return FALSE;
    }

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

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

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

        if ( !ControlService(schService, SERVICE_CONTROL_STOP,
        &ssStatus) )
            goto StopWWWWebErr;
        //start Service pending, Check the status until the service is
        running.
        if ( !QueryServiceStatus(schService, &ssStatus) )
            goto StopWWWWebErr;
        while( ssStatus.dwCurrentState == SERVICE_RUNNING)
        {
            dwOldCheckPoint = ssStatus.dwCheckPoint;
            //Save the current checkpoint.
            Sleep(ssStatus.dwWaitHint);
            //Wait for the
specified interval.
            if ( !QueryServiceStatus(schService, &ssStatus) )
                //Check the status again.
                break;
                if (dwOldCheckPoint >= ssStatus.dwCheckPoint)
                    //Break if the checkpoint has not been incremented.
                    break;
            }

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

            CloseServiceHandle(schService);
            return TRUE;

StopWWWWebErr:

```

```

        CloseServiceHandle(schService);
        return FALSE;
    }

    static void UpdateDialog(HWND hDlg)
    {
        MSG msg;

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

        Sleep(250);
        return;
    }
}

```

install\src\install.h

```

//{{(NO_DEPENDENCIES)}
// Microsoft Developer Studio generated include file.
// Used by install.rc
//

#define IDD_DIALOG1          101
#define IDI_ICON1           102
#define IDR_TPCCDLL         103
#define IDD_DIALOG2         105
#define IDI_ICON2           106
#define IDR_DELIVERY        107
#define IDD_DIALOG3         108

#define BN_LOG              1001
#define ED_KEEP              1002
#define ED_THREADS          1003
#define ED_THREADS2         1004
#define IDC_PATH            1007
#define IDC_VERSION         1009
#define IDC_RESULTS         1010
#define IDC_PROGRESS1       1011
#define IDC_STATUS          1012
#define IDC_BUTTON1         1013

#define ED_MAXCONNECTON     1014
#define ED_IIS_MAX_THREAD_POOL_LIMIT 1015
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE 1017
#define ED_IIS_THREAD_TIMEOUT 1018
#define ED_IIS_LISTEN_BACKLOG 1019

#define IDC_DBLIB           1021
#define IDC_ODBC            1022
#define IDC_CONNECT_POOL   1023
#define ED_USER_CONNECT_DELAY_TIME 1024

// Next default values for new objects
//

```

install\src\install.rc

```

//Microsoft Developer Studio generated resource script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS

```

```

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

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

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

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

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

IDD_DIALOG1 DIALOGEX 0, 0, 219, 351
STYLE DS_MODALFRAME | DS_CENTER | WS_MINIMIZEBOX | WS_POPUP |
WS_CAPTION |
WS_SYSMENU
CAPTION "TPCC Web Client Installation Utility"
FONT 8, "MS Sans Serif"
BEGIN
    EDITTEXT    ED_THREADS,164,45,34,12,ES_RIGHT | ES_NUMBER,
                WS_EX_RTLREADING
    EDITTEXT    ED_MAXDELIVERIES,164,59,34,2,ES_RIGHT |
    ES_NUMBER,
                WS_EX_RTLREADING
    EDITTEXT    ED_MAXCONNECTION,164,73,34,12,ES_RIGHT |
    ES_NUMBER,
                WS_EX_RTLREADING
    CONTROL     "None",IDC_TM_NONE,"Button",BS_AUTORADIOBUTTON |
                WS_GROUP | WS_TABSTOP,43,100,33,10
    CONTROL     "COM",IDC_TM_MTS,"Button",BS_AUTORADIOBUTTON |
                WS_TABSTOP,43,113,32,10
    CONTROL     "TUXEDO",IDC_TM_TUXEDO,"Button",BS_AUTORADIOBUTTON |
                WS_TABSTOP,106,100,46,10
    CONTROL     "ENCINA",IDC_TM_ENCINA,"Button",BS_AUTORADIOBUTTON |
                WS_DISABLED | WS_TABSTOP,106,113,43,10
    EDITTEXT    ED_DB_SERVER,131,152,67,12,ES_AUTOHSCROLL
    EDITTEXT    ED_DB_USER_ID,131,165,67,2,ES_AUTOHSCROLL
    EDITTEXT    ED_DB_PASSWORD,131,178,67,12,ES_AUTOHSCROLL
    EDITTEXT    ED_DB_NAME,131,191,67,12,ES_AUTOHSCROLL
    CONTROL     "DBLIB",IDC_DBLIB,"Button",BS_AUTORADIOBUTTON |
    WS_GROUP |
                WS_TABSTOP,45,219,39,12
    CONTROL     "ODBC",IDC_ODBC,"Button",BS_AUTORADIOBUTTON |
    WS_TABSTOP,
                91,219,39,12
    EDITTEXT    ED_IIS_MAX_THREAD_POOL_LIMIT,164,263,34,12,ES_RIGHT |
    ES_NUMBER,WS_EX_RTLREADING
    EDITTEXT    ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,164,277,34,12,ES_RIGHT |
    ES_NUMBER,WS_EX_RTLREADING
    EDITTEXT    ED_IIS_THREAD_TIMEOUT,164,291,34,12,ES_RIGHT |
    ES_NUMBER,
                WS_EX_RTLREADING
    EDITTEXT    ED_IIS_LISTEN_BACKLOG,164,305,34,12,ES_RIGHT |
    ES_NUMBER,
                WS_EX_RTLREADING
    DEFPPUSHBUTTON "OK",IDOK,53,331,50,14
    PUSHBUTTON    "Cancel",IDCANCEL,119,331,50,14
    EDITTEXT    IDC_PATH,106,26,91,13,ES_AUTOHSCROLL |

```

```

ES_READONLY
LTEXT "Number of Delivery Threads:",IDC_STATIC,35,45,115,12
LTEXT "Max Number of Connections:",IDC_STATIC,35,73,115,12
RTEXT "Version 4.11",IDC_VERSION,120,4,89,9
LTEXT "IIS Max Thread Pool Limit:",IDC_STATIC,36,263,115,12
LTEXT "Web Service Backlog Queue Size:",IDC_STATIC,36,277,115,
12
LTEXT "IIS Thread Timeout (seconds):",IDC_STATIC,36,291,115,12
LTEXT "IIS Listen Backlog:",IDC_STATIC,36,307,115,10
GROUPBOX "Database
Interface",IDC_STATIC,35,208,163,27,WS_GROUP
LTEXT "Installation directory:",IDC_STATIC,35,29,71,10
GROUPBOX "Transaction Monitor",IDC_STATIC,33,90,165,37
LTEXT "Server Name:",IDC_STATIC,35,155,56,8
LTEXT "User ID:",IDC_STATIC,35,168,60,8
LTEXT "User Password:",IDC_STATIC,35,181,83,8
LTEXT "Database Name:",IDC_STATIC,35,194,54,8
GROUPBOX "SQL Server Connection
Properties",IDC_STATIC,22,139,187,
102
GROUPBOX "Web Client Properties",IDC_STATIC,22,15,187,118
GROUPBOX "IIS Settings",IDC_STATIC,22,247,187,79
LTEXT "Max Pending Deliveries:",IDC_STATIC,35,59,115,12
END

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

IDD_DIALOG3 DIALOG DISCARDABLE 0, 0, 91, 40
STYLE DS_SYSDIALOG | DS_MODALFRAME | DS_3DLOOK | DS_CENTER |
WS_CAPTION
CAPTION "Installing TPCG Web Client"
FONT 12, "Arial Black"
BEGIN
CONTROL
"Progress1",IDC_PROGRESS1,"msctls_progress32",WS_BORDER,
7,20,77,13
CTEXT "Static",IDC_STATUS,77,77,12,SS_SUNKEN
END

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

////////////////////////////////////
//
// DESIGNINFO
//

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE

```

```

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

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

IDD_DIALOG3, DIALOG
BEGIN
LEFTMARGIN, 7
RIGHTMARGIN, 8
TOPMARGIN, 7
BOTTOMMARGIN, 33
END

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

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

1 TEXTINCLUDE DISCARDABLE
BEGIN
"resource.h"
END

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

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

#endif // APSTUDIO_INVOKED

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

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

```

```

IDI_ICON2 ICON DISCARDABLE "icon2.ico"

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

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

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

VS_VERSION_INFO VERSIONINFO
FILEVERSION 0,4,20,0
PRODUCTVERSION 0,4,20,0
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x40004L
FILETYPE 0x1L
FILESUBTYPE 0x0L
BEGIN
BLOCK "StringFileInfo"
BEGIN
BLOCK "040904b0"
BEGIN
VALUE "Comments", "TPC-C Web Client Installer"
VALUE "CompanyName", "Microsoft"
VALUE "FileDescription", "install"
VALUE "FileVersion", "0, 4, 20, 00"
VALUE "InternalName", "install"
VALUE "LegalCopyright", "Copyright © 1999"
VALUE "OriginalFilename", "install.exe"
VALUE "ProductName", "Microsoft install"
VALUE "ProductVersion", "0, 4, 20, 00"
END
END
END
BLOCK "VarFileInfo"
BEGIN
VALUE "Translation", 0x409, 1200
END
END
#endif // !_MAC

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

IDR_LICENSE1 LICENSE DISCARDABLE "license.txt"

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

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

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

```

```

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

//
// TUXEDO_APP
//

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

//
// TUXEDO_DLL
//

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

//
// COM_DLL
//

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

//
// COM_PS_DLL
//

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

//
// COM_ALL_DLL
//

IDR_COMALL_DLL   COM_ALL_DLL DISCARDABLE
"..\\..\\tpcc_com_all\\bin\\tpcc_com_all.dll"
#endif // English (U.S.) resources

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

#endif // not APSTUDIO_INVOKED

```

install\\src\\install_com.cpp

```

/*      FILE:          INSTALL_COM.CPP
 *      Microsoft TPC-C Kit Ver.
 *      4.20.000
 *      Copyright Microsoft, 1999
 *      All Rights Reserved
 *
 *      not audited
 *
 *      PURPOSE:  installation code for COM application for TPC-C Web

```

```

Kit
 *      Contact:      Charles Levine (clevine@microsoft.com)
 *
 *      Change history:
 *      4.20.000 - first version
 */

#define _WIN32_WINNT 0x0500

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

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

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

    ICatalogObject* pCatalogObjectApp =
    NULL;
    ICatalogObject* pCatalogObjectCo =
    NULL;
    ICatalogObject* pCatalogObjectTf =
    NULL;
    ICatalogObject* pCatalogObjectM =
    NULL;

    _bstr_t bstrTemp,
    bstrTemp2, bstrTemp3, bstrTemp4;
    _bstr_t bstrDllPath =
    szDllPath;

    _variant_t vTmp, vKey;
    IActProp,
    long bTmp;

    ICount, ICountCo, ICountTf, ICountMethod;
    bool

    CoInitializeEx(NULL, COINIT_MULTITHREADED);

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

    if (!SUCCEEDED(hr)) goto Error;

    bstrTemp = "Applications";

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

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

```

```

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

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

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

            if (wcsncmp(vTmp.bstrVal, L"TPC-C"))
            {
                ICount--;
                continue;
            }
            else
            {
                hr = pCatalogCollectionApp->Remove(ICount - 1);
                if (!SUCCEEDED(hr)) goto Error;
                break;
            }
        }

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

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

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

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

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

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

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

        pCatalogObjectApp->Release();
        pCatalogObjectApp = NULL;

        bstrTemp = "TPC-C";
        // app name
        bstrTemp2 = bstrDllPath + "tpcc_com_all.dll";
        // DLL
        bstrTemp3 = "";

```

```

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

hr = pCOMAdminCat->InstallComponent(bstrTemp,
                                     bstrTemp2,
                                     bstrTemp3,
                                     bstrTemp4);

if (!SUCCEEDED(hr)) goto Error;

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

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

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

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

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

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

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

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

    bstrTemp = "MaxPoolSize";
    vTmp.Clear(); // clear variant so it isn't
    // stored as a bool (_variant_t feature)
    vTmp = (long)30;
    hr = pCatalogObjectCo->put_Value(bstrTemp, vTmp);
    if (!SUCCEEDED(hr)) goto Error;

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

    // save key to get the InterfacesForComponent
    collection
    hr = pCatalogObjectCo->get_Key(&vKey);

```

```

if (!SUCCEEDED(hr)) goto Error;

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

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

hr = pCatalogCollectionIf->get_Count(&CountIf);
if (!SUCCEEDED(hr)) goto Error;

// iterate through interfaces in component
while (ICountIf > 0)
{
    hr = pCatalogCollectionIf-
>get_Item(ICountIf - 1, (IDispatch**) &pCatalogObjectIf);
    if (!SUCCEEDED(hr)) goto Error;

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

    bstrTemp = "MethodsForInterface";
    hr = pCatalogCollectionIf-
>GetCollection(bstrTemp, vKey, (IDispatch**) &pCatalogCollectionMethod);
    if (!SUCCEEDED(hr)) goto Error;

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

    hr = pCatalogCollectionMethod-
>get_Count(&CountMethod);
    if (!SUCCEEDED(hr)) goto Error;

    // iterate through methods of interface
    while (ICountMethod > 0)
    {
        hr =
pCatalogCollectionMethod->get_Item(ICountMethod - 1, (IDispatch**)
&pCatalogObjectMethod);
        if (!SUCCEEDED(hr)) goto
Error;

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

        pCatalogObjectMethod-
>Release();
        NULL;

        ICountMethod--;

        // save changes
        hr = pCatalogCollectionMethod-
>SaveChanges(&IActProp);
        if (!SUCCEEDED(hr)) goto Error;

        pCatalogObjectIf->Release();
        pCatalogObjectIf = NULL;
    }
}

```

```

ICountIf--;
}

pCatalogObjectCo->Release();
pCatalogObjectCo = NULL;
ICountCo--;
}

// save changes
hr = pCatalogCollectionCo->SaveChanges(&IActProp);
if (!SUCCEEDED(hr)) goto Error;

pCatalogCollectionApp->Release();
pCatalogCollectionApp = NULL;

pCatalogCollectionCo->Release();
pCatalogCollectionCo = NULL;

pCatalogCollectionIf->Release();
pCatalogCollectionIf = NULL;

pCatalogCollectionMethod->Release();
pCatalogCollectionMethod = NULL;

Error:
CoUninitialize();

if (!SUCCEEDED(hr))
{
    LPTSTR lpBuf;
    DWORD dwRes =
FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER |
FORMAT_MESSAGE_FROM_SYSTEM,
                NULL,
                hr,
                MAKELANGID(LANG_NEUTRAL,
                SUBLANG_DEFAULT),
                (LPTSTR) &lpBuf,
                0,
                NULL);
    // _tprintf(_T("Error adding components. HRESULT:
0x%x\n%s"), hr, lpBuf);
    return TRUE;
}
else
    return FALSE;
}

install\src\RESOURCE.H

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

```

```

#define IDD_DIALOG2          105
#define IDI_ICON2           106
#define IDR_DELIVERY        107
#define IDD_DIALOG3        108
#define IDR_LICENSE1        112
#define IDD_DIALOG4        113
#define IDR_TPCCOBJ1        117
#define IDR_TPCCSTUB1       118
#define IDR_DBLIB_DLL       122
#define IDR_ODBC_DLL        123
#define IDR_TUXEDO_APP      124
#define IDR_TUXEDO_DLL      125
#define IDR_COM_DLL         126
#define IDR_COMPS_DLL       127
#define IDR_COMALL_DLL      128
#define BN_LOG               1001
#define ED_KEEP              1002
#define ED_THREADS           1003
#define ED_THREADS2         1004
#define IDC_PATH             1007
#define IDC_VERSION         1009
#define IDC_RESULTS         1010
#define IDC_PROGRESS1       1011
#define IDC_STATUS          1012
#define IDC_BUTTON1         1013
#define ED_MAXCONNECTION    1014
#define ED_IIS_MAX_THREAD_POOL_LIMIT 1015
#define ED_MAXDELIVERIES    1016
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE 1017
#define ED_IIS_THREAD_TIMEOUT 1018
#define ED_IIS_LISTEN_BACKLOG 1019
#define IDC_DBLIB           1021
#define IDC_LICENSE         1022
#define IDC_ODBC            1022
#define IDC_CONNECT_POOL    1023
#define ED_DB_SERVER        1023
#define ED_USER_CONNECT_DELAY_TIME 1024
#define ED_DB_USER_ID       1024
#define IDC_MTS              1025
#define IDC_TM_MTS          1025
#define IDC_TM_TUXEDO       1026
#define IDC_TM_NONE         1027
#define ED_DB_PASSWORD      1028
#define ED_DB_NAME          1029
#define IDC_TM_ENCINA       1030

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE 129
#define _APS_NEXT_COMMAND_VALUE 40001
#define _APS_NEXT_CONTROL_VALUE 1024
#define _APS_NEXT_SYMED_VALUE 101
#endif
#endif

```

isapi_dllsrc\tpcc.def

LIBRARY TPCC.DLL

EXPORTS

```

GetExtensionProc @1
HttpExtensionProc @2
TerminateExtension @3

```

isapi_dllsrc\tpcc.rc

```

//Microsoft Developer Studio generated resource script.
//
#include "resource.h"

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

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

#ifdef _WIN32
//
// Version
//
VS_VERSION_INFO VERSIONINFO
FILEVERSION 0,4,0,0
PRODUCTVERSION 0,4,0,0
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x40004L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904b0"
        BEGIN
            VALUE "Comments", "TPC-C HTML DLL Server (DBLIB)"
            VALUE "CompanyName", "Microsoft"
            VALUE "FileDescription", "TPC-C HTML DLL Server (DBLIB)"
            VALUE "FileVersion", "0, 4, 0, 0"
            VALUE "InternalName", "tpcc0"
            VALUE "LegalCopyright", "Copyright © 1997"
            VALUE "OriginalFilename", "tpcc.dll"
            VALUE "ProductName", "Microsoft tpcc0"
            VALUE "ProductVersion", "0, 4, 0, 0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END
#endif // !_MAC

#ifdef APSTUDIO_INVOKED
//

```

```

//
// TEXTINCLUDE
//
1 TEXTINCLUDE DISCARDABLE
BEGIN
    "resource.h"
END
2 TEXTINCLUDE DISCARDABLE
BEGIN
    "#include \"afxres.h\"\n\n"
    "\0"
END
3 TEXTINCLUDE DISCARDABLE
BEGIN
    "\n\n"
    "\0"
END
#endif // APSTUDIO_INVOKED

//
// Dialog
//
IDD_DIALOG1 DIALOG DISCARDABLE 0, 0, 186, 95
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Dialog"
FONT 8, "MS Sans Serif"
BEGIN
    DEFPUSHBUTTON "OK",IDOK,129,7,50,14
    PUSHBUTTON "Cancel",IDCANCEL,129,24,50,14
END

//
// DESIGNINFO
//
#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
    IDD_DIALOG1, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 179
        TOPMARGIN, 7
        BOTTOMMARGIN, 88
    END
END
#endif // English (U.S.) resources

#ifdef APSTUDIO_INVOKED
//
// Generated from the TEXTINCLUDE3 resource.
//
//
// not APSTUDIO_INVOKED

```


isapi_dll\src\tpcc.h

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

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

#define TP_MAX_RETRIES 50

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

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

//This structure defines the data necessary to keep distinct for each terminal or
client connection.
typedef struct _CLIENTDATA
{
    int iNextFree;
    //index of next free element or -1 if this entry in use.
    int w_id;
    //warehouse id assigned at welcome
    form
    int d_id;
    //district id assigned at welcome form

    int iSynclId;
    //synchronization id
}
```

```
int iTickCount;
//time of last access;

CTPCC_BASE *pTxn;

} CLIENTDATA, *PCLIENTDATA;

//This structure is used to define the operational interface for terminal id support
typedef struct _TERM
{
    int iNumEntries;
    //total allocated terminal array entries
    int iFreeList;
    //next available terminal
    array element or -1 if none
    int iMasterSynclId;
    //synchronization id
    CLIENTDATA *pClientData;
    //pointer to allocated client data
} TERM;

typedef TERM *PTERM;
//pointer to terminal structure type

enum WEBERROR
{
    NO_ERR,
    ERR_COMMAND_UNDEFINED,
    ERR_D_ID_INVALID,
    ERR_DELIVERY_CARRIER_ID_RANGE,
    ERR_DELIVERY_CARRIER_INVALID,
    ERR_DELIVERY_MISSING_OCD_KEY,
    ERR_DELIVERY_THREAD_FAILED,
    ERR_GETPROCADDR_FAILED,
    ERR_HTML_ILL_FORMED,
    ERR_INVALID_SYNC_CONNECTION,
    ERR_INVALID_TERMID,
    ERR_LOADDLL_FAILED,
    ERR_MAX_CONNECTIONS_EXCEEDED,
    ERR_MEM_ALLOC_FAILED,
    ERR_MISSING_REGISTRY_ENTRIES,
    ERR_NEWORDER_CUSTOMER_INVALID,
    ERR_NEWORDER_CUSTOMER_KEY,
    ERR_NEWORDER_DISTRICT_INVALID,
    ERR_NEWORDER_FORM_MISSING_DID,
    ERR_NEWORDER_ITEMID_INVALID,
    ERR_NEWORDER_ITEMID_RANGE,
    ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
    ERR_NEWORDER_MISSING_IID_KEY,
    ERR_NEWORDER_MISSING_QTY_KEY,
    ERR_NEWORDER_MISSING_SUPPW_KEY,
    ERR_NEWORDER_NOITEMS_ENTERED,
    ERR_NEWORDER_QTY_INVALID,
    ERR_NEWORDER_QTY_RANGE,
    ERR_NEWORDER_QTY_WITHOUT_SUPPW,
    ERR_NEWORDER_SUPPW_INVALID,
    ERR_NO_SERVER_SPECIFIED,
    ERR_ORDERSTATUS_CID_AND_CLT,
    ERR_ORDERSTATUS_CID_INVALID,
    ERR_ORDERSTATUS_CLT_RANGE,
    ERR_ORDERSTATUS_DID_INVALID,
    ERR_ORDERSTATUS_MISSING_CID_CT,
    ERR_ORDERSTATUS_MISSING_CID_KEY,
    ERR_ORDERSTATUS_MISSING_CLT_KEY,
    ERR_ORDERSTATUS_MISSING_DID_KEY,
    ERR_PAYMENT_CDI_INVALID,
    ERR_PAYMENT_CID_AND_CLT,
    ERR_PAYMENT_CUSTOMER_INVALID,
    ERR_PAYMENT_CWI_INVALID,
    ERR_PAYMENT_DISTRICT_INVALID,
}
```

```
ERR_PAYMENT_HAM_INVALID,
ERR_PAYMENT_HAM_RANGE,
ERR_PAYMENT_LAST_NAME_TO_LONG,
ERR_PAYMENT_MISSING_CDI_KEY,
ERR_PAYMENT_MISSING_CID_CLT,
ERR_PAYMENT_MISSING_CID_KEY,
ERR_PAYMENT_MISSING_CLT,
ERR_PAYMENT_MISSING_CLT_KEY,
ERR_PAYMENT_MISSING_CWI_KEY,
ERR_PAYMENT_MISSING_DID_KEY,
ERR_PAYMENT_MISSING_HAM_KEY,
ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
ERR_STOCKLEVEL_THRESHOLD_INVALID,
ERR_STOCKLEVEL_THRESHOLD_RANGE,
ERR_VERSION_MISMATCH,
ERR_W_ID_INVALID
};

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

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

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

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

    int ErrorType() {return ERR_TYPE_WEBDLL;};
    int ErrorNum() {return m_Error;};
    char *ErrorText();
};

//These constants have already been defined in engstut.h, but since we do
//not want to include it in the delisrv executable
#define TXN_EVENT_START 2
#define TXN_EVENT_STOP 4
#define TXN_EVENT_WARNING 6 //used to
record a warning into the log

//function prototypes

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call,
LPVOID lpReserved);
```

```

void WriteMessageToEventLog(LPTSTR lpszMsg);
void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd,
int *pFormId, int *pTermId, int *pSynclId);
void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId);
void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId);
void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int iError, int
iErrorType, char *szMsg, int iTermId);
void GetKeyValue(char **pQueryString, char *pKey, char *pValue, int iMax,
WEBERROR err);
int GetIntKeyValue(char **pQueryString, char *pKey, WEBERROR NoKeyErr,
WEBERROR NotIntErr);
void TermInit(void);
void TermDeleteAll(void);
int TermAdd(void);
void TermDelete(int id);
void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int
iErrorNum, int iTermId, int iSyncl, char *szErrorText, char *szBuffer);
void MakeMainMenuForm(int iTermId, int iSyncl, char *szForm);
void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA
*pStockLevelData, BOOL blnput, char *szForm);
void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA *pNewOrderData,
BOOL blnput, char *szForm);
void MakePaymentForm(int iTermId, PAYMENT_DATA *pPaymentData, BOOL
blnput, char *szForm);
void MakeOrderStatusForm(int iTermId, ORDER_STATUS_DATA
*pOrderStatusData, BOOL blnput, char *szForm);
void MakeDeliveryForm(int iTermId, DELIVERY_DATA *pDeliveryData, BOOL
blnput, char *szForm);
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId,
char *szBuffer);
void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA
*pNewOrderData);
void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA
*pPaymentData);
void GetOrderStatusData(LPSTR lpszQueryString, ORDER_STATUS_DATA
*pOrderStatusData);
BOOL PostDeliveryInfo(short w_id, short o_carrier_id);
BOOL IsNumeric(char *ptr);
BOOL IsDecimal(char *ptr);
void DeliveryWorkerThread(void *ptr);

```

isapi_dll\src\tpcc.cpp

```

/* FILE: TPCC.C Microsoft TPC-C Kit Ver.
* 4.20.000 Copyright Microsoft, 1999
* All Rights Reserved
* Version 4.10.000 audited
* by Richard Gimarc, Performance Metrics, 3/17/99
* PURPOSE: Main module for TPCC.DLL which is an ISAPI service
dll.
* Contact: Charles Levine (clevine@microsoft.com)
*

```

```

* Change history:
* 4.20.000 - reworked error handling; added options for
COM and Encina txn monitors
*/

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

#include <sqltypes.h>

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

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

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

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

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

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

#define LEN_ERR_STRING 256

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

char szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

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

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

```

```

// of the RTE and web client.
#define WEBCLIENT_VERSION "410"

static CRITICAL_SECTION TermCriticalSection;

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

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

// For deferred Delivery txns:

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

HANDLE hWorkerSemaphore = INVALID_HANDLE_VALUE;
HANDLE hDoneEvent;

HANDLE *pDeliHandles = NULL;

// configuration settings from registry
TPCCREGISTRYDATA Reg;

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

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

DWORD dwDelBuffFreeCount; // number of buffers free

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

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

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

```

```

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

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

                DisableThreadLibraryCalls((HMODULE)hModule);
                InitializeCriticalSection(&TermCriticalSection);

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

                dwDelBuffSize =
min( Reg.dwMaxPendingDeliveries, 10000 ); // min with 10000 as a sanity
constraint
                dwNumDeliveryThreads =
min( Reg.dwNumberOfDeliveryThreads, 100 ); // min with 100 as a sanity
constraint

                TermInit();

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

                    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_CTPEC_TUXEDO*)
GetProcAddress(hLibInstanceTm,"CTPCC_TUXEDO_new");
                    if
                    (pCTPCC_TUXEDO_new == NULL)
                        throw new CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED,
szDllName, GetLastError() );
                    }
                    else if (Reg.eTxnMon ==
ENCINA)
                }
            }
}

```

```

{
    strcpy( szDllName, Reg.szPath );
    strcat( szDllName, "tpcc_encina.dll");
    hLibInstanceTm = LoadLibrary( szDllName );

    if
    (hLibInstanceTm == NULL)
        throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );

    // get
    function pointer to wrapper for class constructor
    pCTPCC_ENCINA_new = (TYPE_CTPCC_ENCINA*)
GetProcAddress(hLibInstanceTm,"CTPCC_ENCINA_new");
    pCTPCC_ENCINA_post_init = (TYPE_CTPCC_ENCINA*)
GetProcAddress(hLibInstanceTm,"CTPCC_ENCINA_post_init");
    if
    (pCTPCC_ENCINA_new == NULL)
        throw new CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED,
szDllName, GetLastError() );
    }
    else if (Reg.eTxnMon ==
COM)
    {
        strcpy( szDllName, Reg.szPath );
        strcat( szDllName, "tpcc_com.dll");

        hLibInstanceTm = LoadLibrary( szDllName );

        if
        (hLibInstanceTm == NULL)
            throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );

        // get
        function pointer to wrapper for class constructor
        pCTPCC_COM_new = (TYPE_CTPCC_COM*)
GetProcAddress(hLibInstanceTm,"CTPCC_COM_new");
        if
        (pCTPCC_COM_new == NULL)
            throw new CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED,
szDllName, GetLastError() );
        }
        // load DLL for database
        if ((Reg.eTxnMon ==
None) || (dwNumDeliveryThreads > 0))
        {
            if
            (Reg.eDB_Protocol == DBLIB)
            {
                strcpy( szDllName, Reg.szPath );
                strcat( szDllName, "tpcc_dblib.dll");

                hLibInstanceDb = LoadLibrary( szDllName );

                if (hLibInstanceDb == NULL)
                    throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );
            }
        }
    }
}

```

```

// get function pointer to wrapper for class constructor
pCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_new");
if (pCTPCC_DBLIB_new == NULL)
    throw new CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED,
szDllName, GetLastError() );
}
else if
(Reg.eDB_Protocol == ODBC)
{
    strcpy( szDllName, Reg.szPath );
    strcat( szDllName, "tpcc_odbc.dll");

    hLibInstanceDb = LoadLibrary( szDllName );

    if (hLibInstanceDb == NULL)
        throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );

    // get function pointer to wrapper for class constructor
    pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*)
    GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
    if (pCTPCC_ODBC_new == NULL)
        throw new CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED,
szDllName, GetLastError() );
    }
    if
    (dwNumDeliveryThreads
    {
        // for deferred
        delivery txns:
        = CreateEvent( NULL, TRUE /* manual reset */, FALSE /* initially not signalled */,
NULL );

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

        dwDelBuffFreeCount = dwDelBuffSize;

        InitJulianTime(NULL);

        // create
        unique log file name based on delilog -yymmdd-hhmm.log
        SYSTEMTIME Time;
        GetLocalTime( &Time );

        wsprintf( szLogFile, "%sdelivery-%2.2d%2.2d%2.2d-
%2.2d%2.2d.log",
            Reg.szPath, Time.wYear % 100, Time.wMonth, Time.wDay,
            Time.wHour, Time.wMinute );

        txnDelilog =

```

```

new CTxnLog(szLogFile, TXN_LOG_WRITE);

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

// allocate
structures for delivery buffers and thread mgmt
pDeliHandles = new HANDLE[dwNumDeliveryThreads];
pDelBuff =
new DELIVERY_TRANSACTION[dwDelBuffSize];
// launch
DeliveryWorkerThread to perform actual delivery txns
for(i=0;
i<dwNumDeliveryThreads; i++)
{
pDeliHandles[i] = (HANDLE) _beginthread( DeliveryWorkerThread,
0, NULL );
if (pDeliHandles[i] == INVALID_HANDLE_VALUE)
throw new
CWEBCLNTErr( ERR_DELIVERY_THREAD_FAILED );
}
break;
case DLL_PROCESS_DETACH:
if
(dwNumDeliveryThreads)
{
if
(txnDelilog != NULL)
{
//write event into txn log for STOP
txxDelilog->WriteCtrlRecToLog(TXN_EVENT_STOP,
szMyComputerName, sizeof(szMyComputerName));

// This will do a clean shutdown of the delivery log file
CTxnLog *txnDelilogLocal = txnDelilog;
txnDelilog= NULL;
delete txnDelilogLocal;
}
delete []
pDeliHandles;
delete []
pDelBuff;

CloseHandle( hWorkerSemaphore );
CloseHandle( hDoneEvent );
DeleteCriticalSection(&DelBuffCriticalSection);
}
DeleteCriticalSection(&TermCriticalSection);
if (hLibInstanceTm !=

```

```

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

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

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

return TRUE;
}

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

```

```

*/
BOOL WINAPI TerminateExtension( DWORD dwFlags )
{
if (pDeliHandles)
{
SetEvent( hDoneEvent );
for(DWORD i=0; i<dwNumDeliveryThreads; i++)
WaitForSingleObject( pDeliHandles[i],
INFINITE );
}
TermDeleteAll();
return TRUE;
}

/* FUNCTION: HttpExtensionProc
*
* PURPOSE: This function is the main entry point for the TPCC
DLL. The internet service
calls this function passing in the http
string.
*
* ARGUMENTS: EXTENSION_CONTROL_BLOCK *pECB
structure pointer to passed in internet
service information.
*
* RETURNS: DWORD HSE_STATUS_SUCCESS
connection
can be dropped if error
HSE_STATUS_SUCCESS_AND_KEEP_CONN keep
connect valid comment sent
*
* COMMENTS: None
*/
DWORD WINAPI HttpExtensionProc(EXTENSION_CONTROL_BLOCK *pECB)
{
int iCmd, FormId, TermId,
iSyncld;
char szBuffer[4096];
int lpbSize;
static char szHeader[] = "200 Ok";
DWORD dwSize = 6; // initial value
is strlen(szHeader)
char szHeader1[4096];

#ifdef ICECAP
StartCAP();
#endif

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

if (TermId != 0)
{
if ( TermId < 0 || TermId >=
Term.iNumEntries || Term.pClientData[TermId].iNextFree !=1 )
{
// debugging...
char szTmp[128];
wsprintf( szTmp, "Invalid

```

```

term ID; TermId = %d", TermId );
    WriteMessageToEventLog( szTmp );

    throw new
CWEBCLNTER( ERR_INVALID_TERMID );
}

//must have a valid syncid here since
termid is valid
Term.pClientData[TermId].iSyncid
if (iSyncid !=
throw new
CWEBCLNTER( ERR_INVALID_SYNC_CONNECTION );

//set use time
Term.pClientData[TermId].iTickCount =
GetTickCount();
}

switch(iCmd)
{
case 0:
    WelcomeForm(pECB, szBuffer);
    break;

case 1:
    switch( FormId )
    {
        case WELCOME_FORM:
            MAIN_MENU_FORM:
                break;

        case
            NEW_ORDER_FORM:
                ProcessNewOrderForm(pECB, TermId, szBuffer);
                break;
                case PAYMENT_FORM:
                    ProcessPaymentForm(pECB, TermId, szBuffer);
                    break;
                    case DELIVERY_FORM:
                        ProcessDeliveryForm(pECB, TermId, szBuffer);
                        break;
                        case
                            ORDER_STATUS_FORM:
                                ProcessOrderStatusForm(pECB, TermId, szBuffer);
                                break;
                                case
                                    STOCK_LEVEL_FORM:
                                        ProcessStockLevelForm(pECB, TermId, szBuffer);
                                        break;
                                        }
                    case 2:
                        // new-order selected from menu;
                        display new-order input form
                        INPUT_FORM, szBuffer);
                        MakeNewOrderForm(TermId, NULL,
                        break;

                    case 3:
                        // payment selected from menu; display
                        payment input form
                        INPUT_FORM, szBuffer);
                        MakePaymentForm(TermId, NULL,
                        break;

                    case 4:
                        // delivery selected from menu; display

```

```

delivery input form
INPUT_FORM, szBuffer);
    MakeDeliveryForm(TermId, NULL,
    break;

    case 5:
        // order-status selected from menu;
        display order-status input form
        INPUT_FORM, szBuffer);
        MakeOrderStatusForm(TermId, NULL,
        break;

    case 6:
        // stock-level selected from menu;
        display stock-level input form
        INPUT_FORM, szBuffer);
        MakeStockLevelForm(TermId, NULL,
        break;

    case 7:
        // ExitCmd
        TermDelete(TermId);
        WelcomeForm(pECB, szBuffer);
        break;

    case 8:
        SubmitCmd(pECB, szBuffer);
        break;

    case 9:
        // menu
        MakeMainMenuForm(TermId,
        Term.pClientData[TermId].iSyncid, szBuffer);
        break;

    case 10:
        // CMD=Clear
        // resets all connections; should only be
        used when no other connections are active
        TermDeleteAll();
        Terminate();
        WelcomeForm(pECB, szBuffer);
        break;

    case 11:
        // CMD=Stats
        StatsCmd(pECB, szBuffer);
        break;
    }
} catch (CBaseErr *e)
{
    ErrorForm( pECB, e->ErrorType(), e->ErrorNum(),
    TermId, iSyncid, e->ErrorText(), szBuffer );
    delete e;
}
catch (...)
{
    ErrorForm( pECB, ERR_TYPE_WEBDLL, 0, TermId,
    iSyncid, "Error: Unhandled exception in Web Client.", szBuffer );
}

#ifdef ICECAP
    StopCAP();
#endif

    lpbSize = strlen(szBuffer);
    wsprintf(szHeader1,
        "Content-Type: text/html\r\n"
        "Content-Length: %d\r\n"
        "Connection: Keep-Alive\r\n\r\n",
    lpbSize);
    strcat( szHeader1, szBuffer );
    ("pECB->ServerSupportFunction)(pECB->ConnID,
    HSE_REQ_SEND_RESPONSE_HEADER, szHeader(LPDWORD) &dwSize,
    (LPDWORD)szHeader1);

    //finish up and keep connection

```

```

pECB->dwHttpStatusCode = 200;
return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}

void WriteMessageToEventLog(LPTSTR lpszMsg)
{
    TCHAR szMsg[256];
    HANDLE hEventSource;
    LPTSTR lpszStrings[2];

    // Use event logging to log the error.
    //
    hEventSource = RegisterEventSource(NULL, TEXT("TPCC.DLL"));

    _stprintf(szMsg, TEXT("Error in TPCC.DLL: "));
    lpszStrings[0] = szMsg;
    lpszStrings[1] = lpszMsg;

    if (hEventSource != NULL)
    {
        ReportEvent(hEventSource, // handle of event source
        EVENTLOG_ERROR_TYPE, // event type
        0, // event category
        0, // event ID
        NULL, // current user's SID
        2, // strings in lpszStrings
        0, // no bytes of raw data
        (LPCTSTR *)lpszStrings, // array of error strings
        NULL); // no raw data

        (VOID) DeregisterEventSource(hEventSource);
    }
}

/* FUNCTION: DeliveryWorkerThread
*
* PURPOSE: This function processes deferred delivery txns. There
are typically several threads running this routine. The
number of threads is determined by an entry
read from the registry. The thread waits
for work by waiting on semaphore. When a delivery txn is posted, the
semaphore is released. After processing
the delivery txn, information is logged to
record the txn status and execution
time.
*/

/*static*/ void DeliveryWorkerThread(void *ptr)
{
    CTPCC_BASE *pTxn = NULL;
    DELIVERY_TRANSACTION delivery;
    PDELIVERY_DATA
    pDeliveryData;
    TXN_RECORD_TPCC_DELIV_DEF txnDeliRec;

    DWORD
    index;
    HANDLE
    handles[2];

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

    assert(txnDeliLog != NULL);

```

```

try
{
    if (Reg.eDB_Protocol == ODBC)
        pTxn =
        pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
        szMyComputerName, Reg.szDbName );
    else if (Reg.eDB_Protocol == DBLIB)
        pTxn =
        pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
        szMyComputerName, Reg.szDbName );
    pDeliveryData = pTxn->BuffAddr_Delivery();
}
catch (CBaseErr *e)
{
    char szTmp[1024];
    sprintf( szTmp, "Error in Delivery Txn thread. Could
not connect to database. "
User=%s, Password=%s, Database=%s",
"%s. Server=%s,
e->ErrorText(),
Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
    WriteMessageToEventLog( szTmp );
    delete e;
    goto ErrorExit;
}
catch (...)
{
    WriteMessageToEventLog(TEXT("Unhandled
exception caught in DeliveryWorkerThread.));
    goto ErrorExit;
}

while (TRUE)
{
    try
    {
        //while delivery thread running, i.e. user
        has not requested termination

        while (TRUE)
        {
            // need to wait for multiple
            objects: program exit or worker semaphore;

            handles[0] = hDoneEvent;
            handles[1] =

            hWorkerSemaphore;

            WaitForMultipleObjects( 2, &handles[0], FALSE, INFINITE );
            if (index ==

            WAIT_OBJECT_0)
            goto

            ErrorExit;

            ZeroMemory(&txnDeliRec,
            sizeof(txnDeliRec));
            txnDeliRec.TxnType =

            TXN_REC_TYPE_TPCC_DELIV_DEF;

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

            EnterCriticalSection(&DelBuffCriticalSection);
            delivery =

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

            dwDelBuffBusyIndex = 0;

            LeaveCriticalSection(&DelBuffCriticalSection);

```

```

        pDeliveryData->w_id =
        pDeliveryData-
        txnDeliRec.w_id =
        txnDeliRec.o_carrier_id =
        pDeliveryData->o_carrier_id;
        txnDeliRec.TxnStartT0 =
        Get64BitTime(&delivery.queue);

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

        GetLocalTime( &trans_end );
        //log txn
        txnDeliRec.TxnStatus =
        ERR_SUCCESS;
        for (int i=0; i<10; i++)
            // wrap-around if at end of buffer
            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];
        sprintf( szTmp, "Error in Delivery Txn
thread. %s", e->ErrorText());
        WriteMessageToEventLog( szTmp );

        // log the error txn
        txnDeliRec.TxnStatus = e->ErrorType();
        if (txnDeliLog != NULL)
            txnDeliLog-
            >WriteToLog(&txnDeliRec);
        delete e;
    }
    catch (...)
    {
        // unhandled exception; shouldn't
        happen; not much we can do...

        WriteMessageToEventLog(TEXT("Unhandled exception caught in
DeliveryWorkerThread.));
    }
}
ErrorExit:
    delete pTxn;
    _endthread();
}

/* FUNCTION: PostDeliveryInfo
*
* PURPOSE: This function enters the delivery txn into the deferred
delivery buffer.
*
* RETURNS: BOOL FALSE delivery
information posted successfully

```

```

*
TRUE error cannot post delivery info
*/
BOOL PostDeliveryInfo(short w_id, short o_carrier_id)
{
    BOOL bError;

    EnterCriticalSection(&DelBuffCriticalSection);
    if (dwDelBuffFreeCount > 0)
    {
        bError = FALSE;
        (pDelBuff+dwDelBuffFreeIndex)->w_id
        = w_id;
        (pDelBuff+dwDelBuffFreeIndex)->o_carrier_id
        = o_carrier_id;
        GetLocalTime(&(pDelBuff+dwDelBuffFreeIndex)-
        >queue);
        dwDelBuffFreeCount--;
        dwDelBuffFreeIndex++;
        if (dwDelBuffFreeIndex == dwDelBuffSize)
            dwDelBuffFreeIndex = 0;
    }
    else
        // No free buffers. Return an error, which indicates
        that the delivery buffer is full.
        // Most likely, the number of delivery worker threads
        needs to be increased to keep up
        // with the txn rate.
        bError = TRUE;
    LeaveCriticalSection(&DelBuffCriticalSection);

    if (!bError)
        // increment worker semaphore to wake up a worker
        thread
        ReleaseSemaphore( hWorkerSemaphore, 1, NULL );
    return bError;
}

/* FUNCTION: ProcessQueryString
*
* PURPOSE: This function extracts the relevent information out of
the http command passed in from
the browser.
*
* COMMENTS: If this is the initial connection i.e. client is at welcome
screen then
there will not be a terminal
id or current form id. If this is the case
then the pTermid and
pFormid return values are undefined.
*/

void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd,
int *pFormid, int *pTermid, int *pSynclid)
{
    char *ptr = pECB->lpszQueryString;
    char szBuffer[25];
    int i;

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

    *pCmd = 0;
    // default is the login

```

```

screen
    *pTermId = 0;

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

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

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

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

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

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

        "<B><BIG>Microsoft TPGC 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='TERMIID' 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>"

```

```

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

    "Txn Monitor

    "Database

    "Max

    "Connections = <B>%d</B><BR>"

    "# of Delivery

    "Threads = <B>%d</B><BR>"

    "Max

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

    , szTxnMonNames[Reg.eTxnMon],
    szDBNames[Reg.eDB_Protocol],
    Reg.dwMaxConnections,
    dwNumDeliveryThreads, dwDelBuffSize );
    strcat( szBuffer, szTmp);

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

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

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

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

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

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

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

        "</PRE></font>"

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

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

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

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

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

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

        "</PRE></font>"

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

    sprintf( szTmp, "Please enter your Warehouse and

```

```

District for this session:<BR>"

    "<font

    face='Courier New' color='blue'><PRE>" );
    strcat( szBuffer, szTmp);
    strcat( szBuffer, "Warehouse ID = <INPUT
NAME='w_id' SIZE=4><BR>"

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

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

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

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

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

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

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

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

    if (Reg.eTxnMon == None)
    {
        // parse Server name
        GetKeyValue(&ptr, "db_server", szServer,
sizeof(szServer), ERR_NO_SERVER_SPECIFIED);
        // parse User name
        GetKeyValue(&ptr, "db_user", szUser, sizeof(szUser),
NO_ERR);
        // parse Password
        GetKeyValue(&ptr, "db_passwd", szPassword,
sizeof(szPassword), NO_ERR);
        // parse Database name
        GetKeyValue(&ptr, "db_name", szDatabase,
sizeof(szDatabase), NO_ERR);
    }

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

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

```

```

iNewTerm = TermAdd();
Term.pClientData[iNewTerm].w_id = w_id;
Term.pClientData[iNewTerm].d_id = d_id;

try
{
    if (Reg.eTxnMon == TUXEDO)
        Term.pClientData[iNewTerm].pTxn =
pCTPCC_TUXEDO_new();
    else if (Reg.eTxnMon == ENCINA)
        Term.pClientData[iNewTerm].pTxn =
pCTPCC_ENCINA_new();
    else if (Reg.eTxnMon == COM)
        Term.pClientData[iNewTerm].pTxn =
pCTPCC_COM_new( Reg.bCOM_SinglePool );
    else if (Reg.eDB_Protocol == ODBC)
        Term.pClientData[iNewTerm].pTxn =
pCTPCC_ODBC_new( szServer, szUser, szPassword, szMyComputerName,
szDatabase );
    else if (Reg.eDB_Protocol == DBLIB)
        Term.pClientData[iNewTerm].pTxn =
pCTPCC_DBLIB_new( szServer, szUser, szPassword, szMyComputerName,
szDatabase );
}
catch (...)
{
    TermDelete(iNewTerm);
    throw; // pass exception upward
}

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

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

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

    EnterCriticalSection(&TermCriticalSection);

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

    LeaveCriticalSection(&TermCriticalSection);

    wsprintf( szBuffer,
"Client Stats</TITLE></HEAD>"
" <HTML><HEAD><TITLE>TPGC Web
Connections: %d </BIG></B><BR></BODY></HTML>"
, iTot );
}

char *CWEBCINT_ERR::ErrorText()
{
    static SERRORMSG errorMsgs[] =

```

```

{
    {
        ERR_COMMAND_UNDEFINED,
        "Command undefined."
    },
    {
        ERR_D_ID_INVALID,
        "Invalid District ID Must be
1 to 10."
    },
    {
        ERR_DELIVERY_CARRIER_ID_RANGE,
        "Delivery Carrier IDout of range must be 1 - 10."
    },
    {
        ERR_DELIVERY_CARRIER_INVALID,
        "Delivery Carrier ID invalid must be numeric 1- 10."
    },
    {
        ERR_DELIVERY_MISSING_OCD_KEY,
        "Delivery missing Carrier ID key\"OCD\"."
    },
    {
        ERR_DELIVERY_THREAD_FAILED,
        "Could not start delivery worker thread."
    },
    {
        ERR_GETPROCADDR_FAILED,
        "Could not map proc in
DLL. GetProcAddr error. DLL="
    },
    {
        ERR_HTML_ILL_FORMED,
        "Required key field is
missing from HTML string."
    },
    {
        ERR_INVALID_SYNC_CONNECTION,
        "Invalid Terminal Sync ID."
    },
    {
        ERR_INVALID_TERMID,
        "Invalid Terminal ID."
    },
    {
        ERR_LOADDLL_FAILED,
        "Load of DLL failed.
DLL="
    },
    {
        ERR_MAX_CONNECTIONS_EXCEEDED,
        "No connections available. MaxConnections is probably too
low." },
    {
        ERR_MISSING_REGISTRY_ENTRIES,
        "Required registry entries are missing. ReRun
INSTALL to correct."
    },
    {
        ERR_NEWORDER_CUSTOMER_INVALID,
        "New Order customer id invalid data type, range = 1 to 3000."
    },
    {
        ERR_NEWORDER_CUSTOMER_KEY,
        "New Order missing Customer key
\"CID\"."
    },
    {
        ERR_NEWORDER_DISTRICT_INVALID,
        "New Order District ID Invalid range 1- 10."
    },
    {
        ERR_NEWORDER_FORM_MISSING_DID,
        "New Order missing District key\"DID\"."
    },
    {
        ERR_NEWORDER_ITEMID_INVALID,
        "New Order Item Id is wrong data type, must be
numeric."
    },
    {
        ERR_NEWORDER_ITEMID_RANGE,
        "New Order Item Id is out of range.

```

```

Range = 1 to 999999."
    },
    {
        ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
        "New Order Item_Id field entered without a corresponding
Supp_W."
    },
    {
        ERR_NEWORDER_MISSING_IID_KEY,
        "New Order missing Item Id key\"IID\"."
    },
    {
        ERR_NEWORDER_MISSING_QTY_KEY,
        "New Order Missing Qty key\"Qty#\"."
    },
    {
        ERR_NEWORDER_MISSING_SUPPW_KEY,
        "New Order missing Supp_W key\"SP#\"."
    },
    {
        ERR_NEWORDER_NOITEMS_ENTERED,
        "New Order No order lines entered."
    },
    {
        ERR_NEWORDER_QTY_INVALID,
        "New Order Qty invalid must be numeric
range 1 - 99."
    },
    {
        ERR_NEWORDER_QTY_RANGE,
        "New Order Qty is out of
range. Range = 1 to 99."
    },
    {
        ERR_NEWORDER_QTY_WITHOUT_SUPPW,
        "New Order Qty field entered without a corresponding Supp_W."
    },
    {
        ERR_NEWORDER_SUPPW_INVALID,
        "New Order Supp_W invalid data type
must be numeric."
    },
    {
        ERR_NO_SERVER_SPECIFIED,
        "No Server name specified."
    },
    {
        ERR_ORDERSTATUS_CID_AND_CLT,
        "Order Status Only Customer ID or Last Name may
be entered, not both."
    },
    {
        ERR_ORDERSTATUS_CID_INVALID,
        "Order Status Customer ID invalid, range must be
numeric 1 - 3000."
    },
    {
        ERR_ORDERSTATUS_CLT_RANGE,
        "Order Status Customer last name
longer than 16 characters."
    },
    {
        ERR_ORDERSTATUS_DID_INVALID,
        "Order Status District invalid, value must be numeric
1 - 10."
    },
    {
        ERR_ORDERSTATUS_MISSING_CID_CLT,
        "Order Status Either Customer ID or Last Name must be entered."
    },
    {
        ERR_ORDERSTATUS_MISSING_CID_KEY,
        "Order Status missing Customer key \"CID\"."
    },
    {
        ERR_ORDERSTATUS_MISSING_CLT_KEY,
        "Order Status missing Customer Last Name key \"CLT\"."
    },
    {
        ERR_ORDERSTATUS_MISSING_DID_KEY,
        "Order Status missing District key \"DID\"."
    },
    {
        ERR_PAYMENT_CDI_INVALID,
        "Payment Customer district invalid must
be numeric."
    },
    {
        ERR_PAYMENT_CID_AND_CLT,

```



```

    "Payment Only Customer ID or Last
Name may be entered, not both."
    {
        ERR_PAYMENT_CUSTOMER_INVALID,
        "Payment Customer data type invalid, must be numeric."
    },
    {
        ERR_PAYMENT_CWI_INVALID,
        "Payment Customer Warehouse invalid,
must be numeric."
    },
    {
        ERR_PAYMENT_DISTRICT_INVALID,
        "Payment District ID is invalid, must be 1-10."
    },
    {
        ERR_PAYMENT_HAM_INVALID,
        "Payment Amount invalid data type
must be numeric."
    },
    {
        ERR_PAYMENT_HAM_RANGE,
        "Payment Amount out of
range, 0 - 9999.99."
    },
    {
        ERR_PAYMENT_LAST_NAME_TO_LONG,
        "Payment Customer last name longer than 16 characters."
    },
    {
        ERR_PAYMENT_MISSING_CDI_KEY,
        "Payment missing Customer district key \"CDI\"."
    },
    {
        ERR_PAYMENT_MISSING_CID_CLT,
        "Payment Either Customer ID or Last Name must be
entered."
    },
    {
        ERR_PAYMENT_MISSING_CID_KEY,
        "Payment missing Customer Key \"CID\"."
    },
    {
        ERR_PAYMENT_MISSING_CLT_KEY,
        "Payment missing Customer Last Name key
V\"CLT\"."
    },
    {
        ERR_PAYMENT_MISSING_CWI_KEY,
        "Payment missing Customer Warehouse key
V\"CWI\"."
    },
    {
        ERR_PAYMENT_MISSING_DID_KEY,
        "Payment missing District Key \"DID\"."
    },
    {
        ERR_PAYMENT_MISSING_HAM_KEY,
        "Payment missing Amount key \"HAM\"."
    },
    },
    {
        ERR_STOCKLEVEL_MISSING_THRESHOLDKEY, "Stock Level;
missing Threshold key \"TT\"."
    },
    {
        ERR_STOCKLEVEL_THRESHOLD_INVALID,
        "Stock Level; Threshold value must be in the range = 1-99."
    },
    {
        ERR_STOCKLEVEL_THRESHOLD_RANGE,
        "Stock Level Threshold out of range, range must be 1 - 99."
    },
    {
        ERR_VERSION_MISMATCH,
        "Invalid version field. RTE
and Web Client are probably out of sync."
    },
    {
        ERR_W_ID_INVALID,
        "Invalid Warehouse ID."
    },
    {
        0,
    },
    ""
}

```

```

};
char szTmp[256];
int i = 0;
while (TRUE)
{
    if (errorMsgs[i].szMsg[0] == 0)
    {
        strcpy( szTmp, "Unknown error
number.");
        break;
    }
    if (m_Error == errorMsgs[i].iError)
    {
        strcpy( szTmp, errorMsgs[i].szMsg );
        break;
    }
    i++;
}
if (m_szTextDetail)
    strcat( szTmp, m_szTextDetail );
if (m_SystemErr)
    sprintf( szTmp+strlen(szTmp), " Error=%d",
m_SystemErr );
m_szErrorText = new char[strlen(szTmp)+1];
strcpy( m_szErrorText, szTmp );
return m_szErrorText;
}
/* FUNCTION: GetKeyValue
*
* PURPOSE: This function parses a http formatted string for
specific key values.
*
* ARGUMENTS: char pQueryString http string from client browser
*             char pKey key value to look for
*             char pValue character array into which
to place key's value
*             int iMax maximum
length of key value array.
*             WEBERROR error value to
throw
*
* RETURNS: nothing.
*
* ERROR: if (the pKey value is not found) then
if (err == 0)
return (empty string)
else
throw CWEBCLNT_ERR(err)
*
* COMMENTS: http keys are formatted either KEY=value& or
KEY=value0. This DLL formats
TPC-C input fields in such
a manner that the keys can be extracted in the
above manner.
*/
void GetKeyValue(char **pQueryString, char *pKey, char *pValue, int iMax,
WEBERROR err)
{
    char *ptr;

```

```

if ( !(ptr=strstr("pQueryString, pKey)) )
    goto ErrorExit;
ptr += strlen(pKey);
if ( *ptr != '=' )
    goto ErrorExit;
ptr++;
iMax--; // one position is for terminating null
while( *ptr && *ptr != ' ' && iMax )
{
    *pValue++ = *ptr++;
    iMax--;
}
*pValue = 0; // terminating null
*pQueryString = ptr;
return;
ErrorExit:
if (err != NO_ERR)
    throw new CWEBCLNT_ERR( err );
*pValue = 0; // return empty result string
}
/* FUNCTION: GetIntKeyValue
*
* PURPOSE: This function parses a http formatted string for a
specific key value.
*
* ARGUMENTS: char pQueryString http string from client browser
*             char pKey key value to look for
*             NoKeyErr error value to throw if key not found
*             WEBERROR
*             NotIntErr error value to throw if value not numeric
*
* RETURNS: integer
*
* ERROR: if (the pKey value is not found) then
if
(NoKeyErr != NO_ERR)
throw CWEBCLNT_ERR(err)
else
return 0
else if (non-numeric char
found) then
if
(NotIntErr != NO_ERR) then
throw CWEBCLNT_ERR(err)
else
return 0
*
* COMMENTS: http keys are formatted either KEY=value& or
KEY=value0. This DLL formats
TPC-C input fields in such
a manner that the keys can be extracted in the
above manner.
*/
int GetIntKeyValue(char **pQueryString, char *pKey, WEBERROR NoKeyErr,
WEBERROR NotIntErr)
{
    char *ptr0;
    char *ptr;

```

```

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

ptr0 = ptr; // remember starting point
// scan string until a terminator (null or &) or a non-digit
while( *ptr && *ptr != '\0' && isdigit(*ptr) )
    ptr++;

// make sure we stopped scanning for the right reason
if ((ptr0 == ptr) || (*ptr && *ptr != '\0'))
{
    if (NotIntErr != NO_ERR)
        throw new
CWEBCLNT_ERR( NoKeyErr );
    return 0;
}

*pQueryString = ptr;
return atoi(ptr0);

ErrorNoKey:
if (NoKeyErr != NO_ERR)
    throw new CWEBCLNT_ERR(NoKeyErr );
return 0;
}

/* FUNCTION: TermInit
*
* PURPOSE: This function initializes the client terminal structure; it
is called when the TPCC.DLL
*
* ARGUMENTS: none
*
* RETURNS: None
*/

void TermInit(void)
{
    EnterCriticalSection(&TermCriticalSection);

    Term.iMasterSynclid = 1;
    Term.iNumEntries = Reg.dwMaxConnections+1;

    Term.pClientData = NULL;
    Term.pClientData =
(PCLIENTDATA)malloc(Term.iNumEntries * sizeof(CLIENTDATA));
    if (Term.pClientData == NULL)
    {
        LeaveCriticalSection(&TermCriticalSection);
        throw new
CWEBCLNT_ERR( ERR_MEM_ALLOC_FAILED );
    }

    ZeroMemory( Term.pClientData, Term.iNumEntries *
sizeof(CLIENTDATA) );

    Term.iFreeList = Term.iNumEntries-1;
    // build free list
    // note: Term.pClientData[0].iNextFree gets set to -1, which marks
it as "in use".
    // This is intentional, as the zero entry is used as an anchor and
never
    // allocated as an actual terminal.
    for(int i=0; i<Term.iNumEntries; i++)
        Term.pClientData[i].iNextFree = i+1;

    LeaveCriticalSection(&TermCriticalSection);
}

/* FUNCTION: TermDeleteAll

```

```

*
* PURPOSE: This function frees allocated resources associated
with the terminal structure.
*
* ARGUMENTS: none
*
* RETURNS: None
*
* COMMENTS: This function is called only when the inet service
unloads the TPCC.DLL
*/

void TermDeleteAll(void)
{
    EnterCriticalSection(&TermCriticalSection);

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

    Term.iFreeList = 0;
    Term.iNumEntries = 0;
    if ( Term.pClientData )
        free(Term.pClientData);
    Term.pClientData = NULL;

    LeaveCriticalSection(&TermCriticalSection);
}

/* FUNCTION: TermAdd
*
* PURPOSE: This function assigns a terminal id which is used to
identify a client browser.
*
* RETURNS: int assigned
*
* ARGUMENTS: terminal id
*/

int TermAdd(void)
{
    DWORD i;
    int iNewTerm, iTickCount;

    if (Term.iNumEntries == 0)
        return -1;

    EnterCriticalSection(&TermCriticalSection);
    if (Term.iFreeList != 0)
    {
        // position is available
        iNewTerm = Term.iFreeList;
        Term.iFreeList =
Term.pClientData[iNewTerm].iNextFree;
        Term.pClientData[iNewTerm].iNextFree = -1; //
indicates this position is in use
    }
    else
    {
        // no open slots, so find the slot that hasn't been used
in the longest time and reuse it
        for(iNewTerm=1, i=1, iTickCount=0x7FFFFFFF;
i<Reg.dwMaxConnections; i++)
        {
            if (iTickCount >
Term.pClientData[i].iTickCount)
            {
                iTickCount =
Term.pClientData[i].iTickCount;
                iNewTerm = i;
            }
        }
    }
}

```

```

}
// if oldest term is less than one minute old, it
probably means that more connections
// are being attempted than were specified as "Max
Connections" at install. In this case,
// do not bump existing connection; instead, return
error to requestor.
if ((GetTickCount()- iTickCount) < 60000)
{
    LeaveCriticalSection(&TermCriticalSection);
    throw new
CWEBCLNT_ERR( ERR_MAX_CONNECTIONS_EXCEEDED );
}

Term.pClientData[iNewTerm].iTickCount = GetTickCount();
Term.pClientData[iNewTerm].iSynclid = Term.iMasterSynclid++;
Term.pClientData[iNewTerm].pTxn = NULL;

LeaveCriticalSection(&TermCriticalSection);
return iNewTerm;
}

/* FUNCTION: TermDelete
*
* PURPOSE: This function makes a terminal entry in the Term
array available for reuse.
*
* ARGUMENTS: int id Terminal id of client exiting
*/

void TermDelete(int id)
{
    if ( id > 0 && id < Term.iNumEntries )
    {
        delete Term.pClientData[id].pTxn;

        // put onto free list
        EnterCriticalSection(&TermCriticalSection);

        Term.pClientData[id].iNextFree = Term.iFreeList;
        Term.iFreeList = id;

        LeaveCriticalSection(&TermCriticalSection);
    }
}

/* FUNCTION: MakeErrorForm
*/

void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int
iErrorNum, int iTermId, int iSynclid, char *szErrorText, char *szBuffer )
{
    wsprintf(szBuffer,
"<HTML><HEAD><TITLE>TPGC
Error</TITLE></HEAD><BODY>"
"<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
"VALUE=\"%d\">" "<INPUT TYPE=\"hidden\" NAME=\"ERROR\""
"VALUE=\"%d\">" "<INPUT TYPE=\"hidden\" NAME=\"FORMID\""
"VALUE=\"%d\">" "<INPUT TYPE=\"hidden\" NAME=\"TERMINID\""
"VALUE=\"%d\">" "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\""
"VALUE=\"%d\">" "<BOLD>An Error Occurred</BOLD><BR><BR>"

```

```

"%s"
"<BR><BR><HR>"
"<INPUT TYPE='submit' NAME='CMD'"
VALUE= "..NewOrder..\">"
"<INPUT TYPE='submit' NAME='CMD'"
VALUE= "..Payment..\">"
"<INPUT TYPE='submit' NAME='CMD'"
VALUE= "..Delivery..\">"
"<INPUT TYPE='submit' NAME='CMD'"
VALUE= "..Order-Status..\">"
"<INPUT TYPE='submit' NAME='CMD'"
VALUE= "..Stock-Level..\">"
"<INPUT TYPE='submit' NAME='CMD'"
VALUE= "..Exit..\">"

"</FORM></BODY></HTML>"
, iType, iErrorNum, MAIN_MENU_FORM, iTermId,
iSyncl, szErrorText ;
}

/* FUNCTION: MakeMainMenuForm
*/

void MakeMainMenuForm(int iTermId, int iSyncl, char *szForm)
{
    sprintf(szForm,
"<HTML><HEAD><TITLE>TPGC Main
Menu</TITLE></HEAD><BODY>"
"Select Desired Transaction.<BR><HR>"
"<FORM ACTION='tpcc.dll' METHOD='GET'>"
"<INPUT TYPE='hidden' NAME='STATUSID'"
VALUE="0\">"
"<INPUT TYPE='hidden' NAME='ERROR'"
VALUE="0\">"
"<INPUT TYPE='hidden' NAME='FORMID'"
VALUE="%d\">"
"<INPUT TYPE='hidden' NAME='TERMI'"
VALUE="%d\">"
"<INPUT TYPE='hidden' NAME='SYNCID'"
VALUE="\">"
"<INPUT TYPE='submit' NAME='CMD'"
VALUE= "..NewOrder..\">"
"<INPUT TYPE='submit' NAME='CMD'"
VALUE= "..Payment..\">"
"<INPUT TYPE='submit' NAME='CMD'"
VALUE= "..Delivery..\">"
"<INPUT TYPE='submit' NAME='CMD'"
VALUE= "..Order-Status..\">"
"<INPUT TYPE='submit' NAME='CMD'"
VALUE= "..Stock-Level..\">"
"<INPUT TYPE='submit' NAME='CMD'"
VALUE= "..Exit..\">"
"</FORM></BODY></HTML>"
, MAIN_MENU_FORM, iTermId, iSyncl);
}

/* FUNCTION: MakeStockLevelForm
*
* PURPOSE: This function constructs the Stock Level HTML page.
*
* COMMENTS: The internal client buffer is created when the terminal
id is assigned and should not be freed except when the
client terminal id is no longer needed.
*/

void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA
*pStockLevelData, BOOL binput, char *szForm)
{
    int c;

    c = sprintf(szForm,
"<HTML><HEAD><TITLE>TPGC Stock

```

```

Level</TITLE></HEAD><FORM ACTION='tpcc.dll' METHOD='GET'>"
"<INPUT TYPE='hidden' NAME='STATUSID'"
VALUE="0\">"
"<INPUT TYPE='hidden' NAME='ERROR'"
VALUE="0\">"
"<INPUT TYPE='hidden' NAME='FORMID'"
VALUE="%d\">"
"<INPUT TYPE='hidden' NAME='TERMI'"
VALUE="%d\">"
"<INPUT TYPE='hidden' NAME='SYNCID'"
VALUE="%d\">"
"<PRE><font face='Courier'"
Stock-Level<BR>"
"Warehouse: %4.4d District: %2.2d<BR> <BR> ",
STOCK_LEVEL_FORM, iTermId,
Term.pClientData[iTermId].iSyncl,
Term.pClientData[iTermId].w_id,
Term.pClientData[iTermId].d_id);

    if ( binput )
    {
        strcpy(szForm+c,
"Stock Level Threshold: <INPUT
NAME='TT'" SIZE=2><BR> <BR>"
"low stock: </font><BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR>"
"<BR></PRE><HR>"
"<INPUT TYPE='submit'"
NAME="CMD\" VALUE="Process\">"
"<INPUT TYPE='submit'"
NAME="CMD\" VALUE="Menu\">"
"</FORM></HTML> ");
    }
    else
    {
        sprintf(szForm+c,
"Stock Level Threshold: %2.2d<BR>
<BR>"
"low stock: %3.3d</font> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR>"
"<BR> <BR></PRE><HR>"
"<INPUT TYPE='submit'"
NAME="CMD\" VALUE="NewOrder..\">"
"<INPUT TYPE='submit'"
NAME="CMD\" VALUE="Payment..\">"
"<INPUT TYPE='submit'"
NAME="CMD\" VALUE="Delivery..\">"
"<INPUT TYPE='submit'"
NAME="CMD\" VALUE="Order-Status..\">"
"<INPUT TYPE='submit'"
NAME="CMD\" VALUE="Stock-Level..\">"
"<INPUT TYPE='submit'"
NAME="CMD\" VALUE="Exit..\">"
"</FORM></HTML>"
, pStockLevelData->low_stock);
    }
}

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

void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA *pNewOrderData,
BOOL binput, char *szForm)
{

```

```

int i, c;
BOOL bValid;
static char szBR[] = " <BR> <BR> <BR> <BR> <BR> <BR> <BR> ";

if (binput)
    assert( pNewOrderData->exec_status_code == eOK
|| pNewOrderData->exec_status_code == eInvalidItem );

bValid = (binput || (pNewOrderData->exec_status_code == eOK));

c = sprintf(szForm,
"<HTML><HEAD><TITLE>TPGC New
Order</TITLE></HEAD><BODY>"
"<FORM ACTION='tpcc.dll' METHOD='GET'>"
"<INPUT TYPE='hidden' NAME='STATUSID'"
VALUE="%d\">"
"<INPUT TYPE='hidden' NAME='ERROR'"
VALUE="0\">"
"<INPUT TYPE='hidden' NAME='FORMID'"
VALUE="%d\">"
"<INPUT TYPE='hidden' NAME='TERMI'"
VALUE="%d\">"
"<INPUT TYPE='hidden' NAME='SYNCID'"
VALUE="\">"
"<PRE><font face='Courier'"
New Order<BR>"
, bValid ? 0 : ERR_BAD_ITEM_ID,
NEW_ORDER_FORM, iTermId, Term.pClientData[iTermId].iSyncl);

    if ( binput )
    {
        c += sprintf(szForm+c, "Warehouse: %4.4d ",
Term.pClientData[iTermId].w_id);

        strcpy( szForm+c,
"District: <INPUT NAME='DID'"
SIZE=1>
Date: <BR>"
"Customer: <INPUT NAME='CID'"
SIZE=4> Name: Credit: %Disc: <BR>"
"Order Number: Number of Lines:
W_tax: D_tax: <BR> <BR>"
"Supp_W Item_Id Item Name
Qty Stock B/G Price Amount<BR>"
"<INPUT NAME='IID00'" SIZE=6>
SIZE=1><BR>"
"<INPUT NAME='SP00'" SIZE=4>
"<INPUT NAME='Qty00'"
"<INPUT NAME='IID01'" SIZE=6>
SIZE=1><BR>"
"<INPUT NAME='SP01'" SIZE=4>
"<INPUT NAME='Qty01'"
"<INPUT NAME='IID02'" SIZE=6>
SIZE=1><BR>"
"<INPUT NAME='SP02'" SIZE=4>
"<INPUT NAME='Qty02'"
"<INPUT NAME='IID03'" SIZE=6>
SIZE=1><BR>"
"<INPUT NAME='SP03'" SIZE=4>
"<INPUT NAME='Qty03'"
"<INPUT NAME='IID04'" SIZE=6>
SIZE=1><BR>"
"<INPUT NAME='SP04'" SIZE=4>
"<INPUT NAME='Qty04'"
"<INPUT NAME='IID05'" SIZE=6>
SIZE=1><BR>"
"<INPUT NAME='SP05'" SIZE=4>
"<INPUT NAME='Qty05'"
"<INPUT NAME='IID06'" SIZE=6>
SIZE=1><BR>"
"<INPUT NAME='SP06'" SIZE=4>
"<INPUT NAME='Qty06'"
"<INPUT NAME='IID07'" SIZE=6>
SIZE=1><BR>"
"<INPUT NAME='SP07'" SIZE=4>
"<INPUT NAME='Qty07'"
"<INPUT NAME='IID08'" SIZE=6>
SIZE=1><BR>"
"<INPUT NAME='SP08'" SIZE=4>
"<INPUT NAME='Qty08'"
"<INPUT NAME='IID09'" SIZE=6>
SIZE=1><BR>"
"<INPUT NAME='SP09'" SIZE=4>

```

```

<INPUT NAME="IID09" SIZE=6>
SIZE=1><BR>"
" <INPUT NAME="SP10" SIZE=4>
<INPUT NAME="IID10" SIZE=6>
SIZE=1><BR>"
" <INPUT NAME="SP11" SIZE=4>
<INPUT NAME="IID11" SIZE=6>
SIZE=1><BR>"
" <INPUT NAME="SP12" SIZE=4>
<INPUT NAME="IID12" SIZE=6>
SIZE=1><BR>"
" <INPUT NAME="SP13" SIZE=4>
<INPUT NAME="IID13" SIZE=6>
SIZE=1><BR>"
" <INPUT NAME="SP14" SIZE=4>
<INPUT NAME="IID14" SIZE=6>
SIZE=1><BR>"
"Execution Status:
" </font></PRE><HR>"
" <INPUT TYPE="submit">
" <INPUT TYPE="submit">
" </FORM></HTML>"
);
}
else
{
c += sprintf(szForm+c, "Warehouse: %4.4d
District: %2.2d
Date: ",
pNewOrderData->w_id,
pNewOrderData->d_id);
if ( bValid )
{
c += sprintf(szForm+c, "%2.2d-%2.2d-
%4.4d %2.2d:%2.2d:%2.2d",
pNewOrderData-
>o_entry_d.day,
pNewOrderData-
>o_entry_d.month,
pNewOrderData-
>o_entry_d.year,
pNewOrderData-
>o_entry_d.hour,
pNewOrderData-
>o_entry_d.minute,
pNewOrderData-
>o_entry_d.second);
}
c += sprintf(szForm+c, "<BR>Customer: %44d
Name: %-16s Credit: %-2s ",
pNewOrderData->c_id,
pNewOrderData->c_last, pNewOrderData->c_credit);
if ( bValid )
{
c += sprintf(szForm+c,
"%%Disc: %5.2f <BR>"
"Order Number: %8.8d Number of Lines: %2.2d W_tax: %5.2f
D_tax: %5.2f <BR> <BR>"
" Supp_W Item_Id Item Name Qty Stock B/G Price
Amount<BR>",
100.0*pNewOrderData-
>c_discount,
pNewOrderData->o_id,
pNewOrderData->o_ol_cnt,

```

```

100.0 * pNewOrderData-
>w_tax,
100.0 * pNewOrderData-
>d_tax);
for(i=0; i<pNewOrderData->o_ol_cnt;
i++)
{
c += sprintf(szForm+c,
"%4.4d %6.6d %2.2s %2.2d %3.3d %1.1s $%6.2f $%7.2f <BR>",
pNewOrderData->OL[i].ol_supply_w_id,
pNewOrderData->OL[i].ol_i_id,
pNewOrderData->OL[i].ol_i_name,
pNewOrderData->OL[i].ol_quantity,
pNewOrderData->OL[i].ol_stock,
pNewOrderData->OL[i].ol_brand_generic,
pNewOrderData->OL[i].ol_i_price,
pNewOrderData->OL[i].ol_amount );
}
else
{
c += sprintf(szForm+c,
"%Disc:<BR>"
"Order Number: %8.8d
" Supp_W Item_Id Item
Name Qty Stock B/G Price Amount<BR>"
, pNewOrderData->o_id);
i = 0;
strncpy( szForm+c, szBR, (15-i)*5 );
c += (15-i)*5;
if ( bValid )
c += sprintf(szForm+c, "Execution
Total: $%8.2f ",
pNewOrderData-
>total_amount);
else
c += sprintf(szForm+c, "Execution
Total:");
strcpy(szForm+c,
" <BR></font></PRE><HR>"
" <INPUT TYPE="submit">
NAME="CMD" VALUE="..NewOrder..">"
" <INPUT TYPE="submit">
NAME="CMD" VALUE="..Payment..">"
" <INPUT TYPE="submit">
NAME="CMD" VALUE="..Delivery..">"
" <INPUT TYPE="submit">
NAME="CMD" VALUE="..Order-Status..">"
" <INPUT TYPE="submit">
NAME="CMD" VALUE="..Stock-Level..">"
" <INPUT TYPE="submit">
NAME="CMD" VALUE="..Exit..">"
" </FORM></HTML>"
);
}
}

```

```

/* FUNCTION: MakePaymentForm
*
* COMMENTS: The internal client buffer is created when the terminal
id is assigned and should not
be freed except when the
client terminal id is no longer needed.
*/
void MakePaymentForm(int iTermId, PAYMENT_DATA *pPaymentData, BOOL
bInput, char *szForm)
{
int c;
c = sprintf(szForm,
"HTML<<HEAD><TITLE>TPGC
Payment</TITLE></HEAD><BODY>"
"FORM ACTION="tpcc.dll" METHOD="GET">"
"INPUT TYPE="hidden" NAME="STATUSID"
VALUE="0">"
"INPUT TYPE="hidden" NAME="ERROR"
VALUE="0">"
"INPUT TYPE="hidden" NAME="FORMID"
VALUE="%d">"
"INPUT TYPE="hidden" NAME="TERMID"
VALUE="%d">"
"INPUT TYPE="hidden" NAME="SYNCID"
VALUE="%d">"
"PRE><font face="Courier">
Payment<BR>"
"Date: "
, PAYMENT_FORM, iTermId,
Term.pClientData[iTermId].iSyncId);
if ( !bInput )
{
c += sprintf(szForm+c, "%2.2d-%2.2d-
%4.4d %2.2d:%2.2d:%2.2d",
pPaymentData->h_date.day,
pPaymentData->h_date.month,
pPaymentData->h_date.year,
pPaymentData->h_date.hour,
pPaymentData->h_date.minute,
pPaymentData->h_date.second);
}
if ( !bInput )
{
c += sprintf(szForm+c,
"<BR> <BR>Warehouse: %4.4d"
" District: <INPUT
NAME="DID" SIZE=1><BR> <BR> <BR> <BR> <BR>"
"Customer: <INPUT NAME="CID"
SIZE=4>"
"Cust-Warehouse: <INPUT
NAME="CWI" SIZE=4> "
"Cust-District: <INPUT NAME="CDI"
SIZE=1><BR>"
"Name: <INPUT
NAME="CLT" SIZE=16>
Since:<BR>"
"
Credit:<BR>"
"
Disc:<BR>"
"
Phone:<BR> <BR>"
"Amount Paid: $<INPUT
NAME="HAM" SIZE=7> New Cust-Balance:<BR>"
"Credit Limit:<BR> <BR>Cust-Data:
<BR> <BR> <BR> <BR> <BR></font></PRE><HR>"
" <INPUT TYPE="submit">
NAME="CMD" VALUE="Process"><INPUT TYPE="submit" NAME="CMD"
VALUE="Menu">"

```

```
                "</BODY></FORM></HTML>"
                , Term.pClientData[iTermId].w_id);
    }
    else
    {
        c += sprintf(szForm+c,
            "<BR><BR>Warehouse: %4.4d
District: %2.2d<BR>"
            "%-20s            %-20s<BR>"
            "%-20s            %-20s<BR>"
            "%-20s %-2s %5.5s-%4.4s  %-
20s %-2s %5.5s-%4.4s<BR> <BR>"
            "Customer: %4.4d Cust-
Warehouse: %4.4d Cust-District: %2.2d<BR>"
            "Name: %-16s %-2s %-16s
Since: %2.2d-%2.2d-%4.4d<BR>"
            "    %-20s            Credit: %-
2s<BR>"
            , Term.pClientData[iTermId].w_id,
            pPaymentData->d_id,
            pPaymentData->d_street_1,
            pPaymentData->d_street_2,
            pPaymentData->w_city,
            pPaymentData->w_state, pPaymentData->w_zip, pPaymentData->w_zip+5,
            pPaymentData->d_city,
            pPaymentData->d_state, pPaymentData->d_zip, pPaymentData->d_zip+5,
            pPaymentData->c_id, pPaymentData-
>c_w_id,  pPaymentData->c_d_id,
            pPaymentData->c_first,
            pPaymentData->c_middle, pPaymentData->c_last,
            pPaymentData->c_since.day,
            pPaymentData->c_since.month,  pPaymentData->c_since.year,
            pPaymentData->c_street_1,
            pPaymentData->c_credit
            );
        c += sprintf(szForm+c,
            "    %-
20s            %Disc: %5.2f<BR>",
            pPaymentData->c_street_2,
            100.0*pPaymentData->c_discount);
        c += sprintf(szForm+c,
            "    %-20s %-2s %5.5s-%4.4s
Phone: %6.6s-%3.3s-%3.3s-%4.4s<BR> <BR>"
            , pPaymentData->c_city, pPaymentData-
>c_state, pPaymentData->c_zip, pPaymentData->c_zip+5,
            pPaymentData->c_phone,
            pPaymentData->c_phone+6, pPaymentData->c_phone+9, pPaymentData-
>c_phone+12);
        c += sprintf(szForm+c,
            "Amount Paid:    $%7.2f  New
Cust-Balance: $%14.2f<BR>"
            "Credit Limit:  $%13.2f<BR> <BR>"
            , pPaymentData->h_amount,
            pPaymentData->c_balance,
            pPaymentData->c_credit_lim
            );
        if ( pPaymentData->c_credit[0] == 'B' &&
            pPaymentData->c_credit[1] == 'C' )
            c += sprintf(szForm+c,
                "Cust-
Data: %-50.50s<BR>    %-50.50s<BR>    %-50.50s<BR>    %-
50.50s<BR>",
                pPaymentData->c_data, pPaymentData->c_data+50,
                pPaymentData->c_data+100, pPaymentData->c_data+150 );
    }
}
```

```
    else
        strcpy(szForm+c, "Cust-Data: <BR>
<BR> <BR> <BR>");
        strcpy(szForm,
            "
            "<INPUT TYPE='submit' NAME='CMDI'
VALUE='..NewOrder..'\>"
            "<INPUT TYPE='submit' NAME='CMDI'
VALUE='..Payment..'\>"
            "<INPUT TYPE='submit' NAME='CMDI'
VALUE='..Delivery..'\>"
            "<INPUT TYPE='submit' NAME='CMDI' VALUE='..Order-
Status..'\>"
            "<INPUT TYPE='submit' NAME='CMDI' VALUE='..Stock-
Level..'\>"
            "<INPUT TYPE='submit' NAME='CMDI' VALUE='..Exit..'\>"
            );
    }
    /* FUNCTION: MakeOrderStatusForm
    *
    * COMMENTS:      The internal client buffer is created when the terminal
id is assigned and should not
    *                               be freed except when the
client terminal id is no longer needed.
    */
    void MakeOrderStatusForm(int iTermId, ORDER_STATUS_DATA
*pOrderStatusData, BOOL bInput, char *szForm)
    {
        int            i, c;
        static char szBR[] = " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>";
        c = sprintf(szForm,
            "<HTML><HEAD><TITLE>TPGC Order-
Status</TITLE></HEAD><BODY>"
            "<FORM ACTION='tpcc.dll' METHOD='GET'\>"
            "<INPUT TYPE='hidden' NAME='STATUSID'"
            VALUE='0'\>"
            "<INPUT TYPE='hidden' NAME='ERRORI'"
            VALUE='0'\>"
            "<INPUT TYPE='hidden' NAME='FORMID'"
            VALUE='%d'\>"
            "<INPUT TYPE='hidden' NAME='TERMID'"
            VALUE='%d'\>"
            "<INPUT TYPE='hidden' NAME='SYNCDID'"
            VALUE='%d'\>"
            "<PRE><font face='Courier'>
Order-Status<BR>"
            "Warehouse: %4.4d ",
            ORDER_STATUS_FORM, iTermId,
            Term.pClientData[iTermId].iSyncd, Term.pClientData[iTermId].w_id);
        if ( bInput )
        {
            strcpy(szForm+c,
                "District: <INPUT NAME='DID'"
                SIZE=1><BR>"
                "Customer: <INPUT NAME='CID'"
                SIZE=4> Name:    <INPUT NAME='CLT'"
                SIZE=23><BR>"
                "Cust-Balance: <BR> <BR>"
            );
        }
    }
}
```

```
    "Order-Number:    Entry-Date:
Carrier-Number:<BR>"
            "Supply-W Item-Id Qty Amount
DeliveryDate<BR> <BR> <BR> <BR> <BR>
" <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR></font></PRE>"
            "<HR><INPUT TYPE='submit'"
NAME='CMDI' VALUE='Process'\><INPUT TYPE='submit'"
NAME='CMDI' VALUE='Menu'\>"
            "</BODY></FORM></HTML> ");
    }
    else
    {
        c += sprintf(szForm+c,
            "District: %2.2d<BR>"
            "Customer: %4.4d Name: %-16s %-
2s %-16s<BR>",
            pOrderStatusData->d_id,
            pOrderStatusData->c_id,
            pOrderStatusData->c_first,
            pOrderStatusData->c_middle, pOrderStatusData->c_last);
        c += sprintf(szForm+c, "Cust-Balance: $%9.2f<BR>
<BR>",
            pOrderStatusData->c_balance);
        c += sprintf(szForm+c,
            "Order-Number: %8.8d Entry-
Date: %2.2d-%2.2d-%4.4d %2.2d-%2.2d-%2.2d-%4.4d<BR>"
            "Supply-W Item-Id Qty Amount
DeliveryDate<BR>",
            pOrderStatusData->o_id,
            pOrderStatusData->o_entry_d.day,
            pOrderStatusData->o_entry_d.month,
            pOrderStatusData->o_entry_d.year,
            pOrderStatusData->o_entry_d.hour,
            pOrderStatusData->o_entry_d.minute,
            pOrderStatusData->o_entry_d.second,
            pOrderStatusData->o_carrier_id);
        for(i=0; i< pOrderStatusData->o_o_cnt; i++)
        {
            c += sprintf(szForm+c,
                "%4.4d %6.6d %2.2d $%8.2f
%2.2d-%2.2d-%4.4d<BR>",
                pOrderStatusData-
>OL[i].ol_supply_w_id,
                pOrderStatusData-
>OL[i].ol_i_id,
                pOrderStatusData-
>OL[i].ol_quantity,
                pOrderStatusData-
>OL[i].ol_amount,
                pOrderStatusData-
>OL[i].ol_delivery_d.day,
                pOrderStatusData-
>OL[i].ol_delivery_d.month,
                pOrderStatusData-
>OL[i].ol_delivery_d.year);
        }
        strcpy( szForm+c, szBR, (15-i)*5 );
        c += (15-i)*5;
        strcpy(szForm+c,
            "</font></PRE><HR><INPUT
TYPE='submit' NAME='CMDI' VALUE='..NewOrder..'\>"
            "<INPUT TYPE='submit'"
NAME='CMDI' VALUE='..Payment..'\>"
            "<INPUT TYPE='submit'"
NAME='CMDI' VALUE='..Delivery..'\>"
            "<INPUT TYPE='submit'"
NAME='CMDI' VALUE='..Order-Status..'\>"
        );
    }
}
```

```

        "<INPUT TYPE='submit'"
NAME="CMD1" VALUE="..Stock-Level..!>"
        "<INPUT TYPE='submit'"
NAME="CMD1" VALUE="..Exit..!>"
    "</BODY></FORM></HTML>");
}

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

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

    c = sprintf(szForm,
        "<HTML><HEAD><TITLE>TPGC
Delivery</TITLE></HEAD><BODY>"
        "<FORM ACTION='tpcc.dll' METHOD='GET'>"
        "<INPUT TYPE='hidden' NAME='STATUSID'"
VALUE="%"d!>"
        "<INPUT TYPE='hidden' NAME='ERROR'"
VALUE="0!>"
        "<INPUT TYPE='hidden' NAME='FORMID'"
VALUE="%"d!>"
        "<INPUT TYPE='hidden' NAME='TERMID'"
VALUE="%"d!>"
        "<INPUT TYPE='hidden' NAME='SYNCDID'"
VALUE="%"d!>"
        "<PRE><font face='Courier'>"
        "Warehouse: %4.4d<BR> <BR> ",
        (bInput && (pDeliveryData->exec_status_code !=
eOK)) ? ERR_TYPE_DELIVERY_POST : 0,
        DELIVERY_FORM, iTermId,
        Term.pClientData[iTermId].iSyncl, Term.pClientData[iTermId].w_id);

    if ( bInput )
    {
        strcpy( szForm+c,
            "Carrier Number: <INPUT
NAME='OCD' SIZE=1><BR> <BR>"
            "Execution Status: <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR>"
            "<BR> <BR> </font></PRE><HR>"
            "<INPUT TYPE='submit'"
NAME="CMD1" VALUE="Process!>"
            "<INPUT TYPE='submit'"
NAME="CMD1" VALUE="Menu!>"
            "</BODY></FORM></HTML>");
    }
    else
    {
        sprintf( szForm+c,
            "Carrier Number: %2.2d<BR> <BR>"
            "Execution Status: %s <BR> <BR>"
            "<BR> <BR> <BR> <BR> <BR> <BR>"
            "<BR> <BR> </font></PRE>"
            "<HR><INPUT TYPE='submit'"
NAME="CMD1" VALUE="..NewOrder..!>"
            "<INPUT TYPE='submit'"
NAME="CMD1" VALUE="..Payment..!>"
            "<INPUT TYPE='submit'"
NAME="CMD1" VALUE="..Delivery..!>"

```

```

        "<INPUT TYPE='submit'"
NAME="CMD1" VALUE="..Order-Status..!>"
        "<INPUT TYPE='submit'"
NAME="CMD1" VALUE="..Stock-Level..!>"
        "<INPUT TYPE='submit'"
NAME="CMD1" VALUE="..Exit..!>"
    "</BODY></FORM></HTML>"
        , pDeliveryData->o_carrier_id,
        (pDeliveryData->exec_status_code ==
eOK) ? "Delivery has been queued." : "Delivery Post Failed "
    );
}

/* FUNCTION: ProcessNewOrderForm
*
* PURPOSE:      This function gets and validates the input data from
the new order form
*
*                filling in the required input variables. It
then calls the SQLNewOrder
*
*                transaction, constructs the output form
and writes it back to client
*
*                browser.
*/

void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer)
{
    PNEW_ORDER_DATA      pNewOrder;

    pNewOrder = Term.pClientData[iTermId].pTxn
>BuffAddr_NewOrder();

    ZeroMemory(pNewOrder, sizeof(NEW_ORDER_DATA));
    pNewOrder->w_id = Term.pClientData[iTermId].w_id;
    GetNewOrderData(pECB->lpszQueryString, pNewOrder);

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

    pNewOrder = Term.pClientData[iTermId].pTxn
>BuffAddr_NewOrder();
    MakeNewOrderForm(iTermId, pNewOrder, OUTPUT_FORM,
szBuffer);
}

/* FUNCTION: void ProcessPaymentForm
*
* PURPOSE:      This function gets and validates the input data from
the payment form
*
*                filling in the required input variables. It
then calls the SQLPayment
*
*                transaction, constructs the output form
and writes it back to client
*
*                browser.
*
* ARGUMENTS:    EXTENSION_CONTROL_BLOCK *pECB
                passed in structure pointer from inetsrv.
                int iTermId client
                browser terminal id
*/

void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer)
{
    PPAYMENT_DATA      pPayment;

    pPayment = Term.pClientData[iTermId].pTxn
>BuffAddr_Payment();
    ZeroMemory(pPayment, sizeof(PAYMENT_DATA));

```

```

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

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

    pPayment = Term.pClientData[iTermId].pTxn
>BuffAddr_Payment();
    MakePaymentForm(iTermId, pPayment, OUTPUT_FORM,
szBuffer);
}

/* FUNCTION: ProcessOrderStatusForm
*
* PURPOSE:      This function gets and validates the input data from
the Order Status
*
*                form filling in the required input
variables. It then calls the
*
*                SQLOrderStatus transaction, constructs
the output form and writes it
*
*                back to client browser.
*
* ARGUMENTS:    EXTENSION_CONTROL_BLOCK *pECB
                passed in structure pointer from inetsrv.
                int iTermId client
                browser terminal id
*/

void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer)
{
    PORDER_STATUS_DATA pOrderStatus;

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

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

    pOrderStatus = Term.pClientData[iTermId].pTxn
>BuffAddr_OrderStatus();
    MakeOrderStatusForm(iTermId, pOrderStatus, OUTPUT_FORM,
szBuffer);
}

/* FUNCTION: ProcessDeliveryForm
*
* PURPOSE:      This function gets and validates the input data from
the delivery form
*
*                filling in the required input variables. It
then calls the PostDeliveryInfo
*
*                Api, The client is then informed that the
transaction has been posted.
*
* ARGUMENTS:    EXTENSION_CONTROL_BLOCK *pECB
                passed in structure pointer from inetsrv.
                int iTermId client
                browser terminal id
*/

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

    PDELIVERY_DATA      pDelivery;

```

```

pDelivery = Term.pClientData[iTermId].pTxn->BuffAddr_Delivery();
ZeroMemory(pDelivery, sizeof(DELIVERY_DATA));
pDelivery->w_id = Term.pClientData[iTermId].w_id;

pDelivery->o_carrier_id = GetIntKeyValue(&ptr, "OCD",
ERR_DELIVERY_MISSING_OCD_KEY, ERR_DELIVERY_CARRIER_INVALID);
if ( pDelivery->o_carrier_id > 10 || pDelivery->o_carrier_id < 1 )
    throw new
CWEBCLNT_ERR( ERR_DELIVERY_CARRIER_ID_RANGE );

if (dwNumDeliveryThreads)
{
    //post delivery info
    if ( PostDeliveryInfo(pDelivery->w_id, pDelivery-
>o_carrier_id )
        pDelivery->exec_status_code =
eDeliveryFailed;
    else
        pDelivery->exec_status_code = eOK;
}
else // delivery is done synchronously if no delivery threads
configured
    Term.pClientData[iTermId].pTxn->Delivery();

pDelivery = Term.pClientData[iTermId].pTxn->BuffAddr_Delivery();
MakeDeliveryForm(iTermId, pDelivery, OUTPUT_FORM, szBuffer);
}

/* FUNCTION: ProcessStockLevelForm
*
* PURPOSE: This function gets and validates the input data from
the Stock Level form filling in the required input
variables. It then calls the SQLStockLevel transaction, constructs
the output form and writes it back to client browser.
*
* ARGUMENTS: EXTENSION_CONTROL_BLOCK *pECB
passed in structure pointer from inetrv. int iTermId client
browser terminal id
*/

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

    PSTOCK_LEVEL_DATA pStockLevel;

    pStockLevel = Term.pClientData[iTermId].pTxn
->BuffAddr_StockLevel();
    ZeroMemory( pStockLevel, sizeof(STOCK_LEVEL_DATA) );

    pStockLevel->w_id = Term.pClientData[iTermId].w_id;
    pStockLevel->d_id = Term.pClientData[iTermId].d_id;

    pStockLevel->threshold = GetIntKeyValue(&ptr, "TT",
ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
ERR_STOCKLEVEL_THRESHOLD_INVALID);
    if ( pStockLevel->threshold >= 100 || pStockLevel->threshold < 0 )
        throw new
CWEBCLNT_ERR( ERR_STOCKLEVEL_THRESHOLD_RANGE );

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

    pStockLevel = Term.pClientData[iTermId].pTxn
->BuffAddr_StockLevel();
    MakeStockLevelForm(iTermId, pStockLevel, OUTPUT_FORM,

```

```

szBuffer);
}

/* FUNCTION: GetNewOrderData
*
* PURPOSE: This function extracts and validates the new order
form data from an http command string.
*
* ARGUMENTS: LPSTR lpszQueryString client browser http
command string NEW_ORDER_DATA
pointer to new order data
*
* structure *pNewOrderData
*/

void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA
*pNewOrderData)
{
    char szTmp[26];
    int i;
    short items;
    int ol_i_id, ol_quantity;
    char *ptr = lpszQueryString;

    static char szSP[MAX_OL_NEW_ORDER_ITEMS][6] =
{ "SP00", "SP01", "SP02", "SP03", "SP04",
"SP05", "SP06", "SP07", "SP08", "SP09",
"SP10", "SP11", "SP12", "SP13", "SP14" };
    static char szIID[MAX_OL_NEW_ORDER_ITEMS][7] =
{ "IID00", "IID01", "IID02", "IID03", "IID04",
"IID05", "IID06", "IID07", "IID08", "IID09",
"IID10", "IID11", "IID12", "IID13", "IID14" };
    static char szQty[MAX_OL_NEW_ORDER_ITEMS][7] =
{ "Qty00", "Qty01", "Qty02", "Qty03", "Qty04",
"Qty05", "Qty06", "Qty07", "Qty08", "Qty09",
"Qty10", "Qty11", "Qty12", "Qty13", "Qty14" };

    pNewOrderData->d_id = GetIntKeyValue(&ptr, "DID",
ERR_NEWORDER_FORM_MISSING_DID,
ERR_NEWORDER_DISTRICT_INVALID);
    pNewOrderData->c_id = GetIntKeyValue(&ptr, "CID",
ERR_NEWORDER_CUSTOMER_KEY,
ERR_NEWORDER_CUSTOMER_INVALID);

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

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

            ol_quantity = pNewOrderData-
>OL[items].ol_quantity =
                GetIntKeyValue(&ptr,
szQty[i], ERR_NEWORDER_MISSING_QTY_KEY,
ERR_NEWORDER_QTY_INVALID);

```

```

            if ( ol_quantity > 99 || ol_quantity < 1 )
                throw new
CWEBCLNT_ERR( ERR_NEWORDER_QTY_RANGE );
        }
        items++;
    }
    else
    {
        // nothing entered for supply warehouse,
        so item id and qty must also be blank
        GetKeyValue(&ptr, szIID[i], szTmp,
sizeof(szTmp), ERR_NEWORDER_MISSING_IID_KEY);
        if ( szTmp[0] )
            throw new
CWEBCLNT_ERR( ERR_NEWORDER_ITEMID_WITHOUT_SUPPW );

        GetKeyValue(&ptr, szQty[i], szTmp,
sizeof(szTmp), ERR_NEWORDER_MISSING_QTY_KEY);
        if ( szTmp[0] )
            throw new
CWEBCLNT_ERR( ERR_NEWORDER_QTY_WITHOUT_SUPPW );
    }
    if ( items == 0 )
        throw new
CWEBCLNT_ERR( ERR_NEWORDER_NOITEMS_ENTERED );

    pNewOrderData->o_ol_cnt = items;
}

/* FUNCTION: GetPaymentData
*
* PURPOSE: This function extracts and validates the payment form
data from an http command string.
*
* ARGUMENTS: LPSTR lpszQueryString client browser http
command string PAYMENT_DATA
pointer to payment data
*
* structure *pPaymentData
*/

void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA
*pPaymentData)
{
    char szTmp[26];
    char *ptr = lpszQueryString;
    BOOL bCustIdBlank;

    pPaymentData->d_id = GetIntKeyValue(&ptr, "DID",
ERR_PAYMENT_MISSING_DID_KEY, ERR_PAYMENT_DISTRICT_INVALID);

    GetKeyValue(&ptr, "CID", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_CID_KEY);
    if ( szTmp[0] == 0 )
    {
        bCustIdBlank = TRUE;
        pPaymentData->c_id = 0;
    }
    else
    {
        // parse customer id and verify that last name was
        bCustIdBlank = FALSE;
        if ( !IsNumeric(szTmp) )
            throw new
CWEBCLNT_ERR( ERR_PAYMENT_CUSTOMER_INVALID );
        pPaymentData->c_id = atoi(szTmp);
    }

    pPaymentData->c_w_id = GetIntKeyValue(&ptr, "CWI",
ERR_PAYMENT_MISSING_CWI_KEY, ERR_PAYMENT_CWI_INVALID);
    pPaymentData->c_d_id = GetIntKeyValue(&ptr, "CDI",

```

```

ERR_PAYMENT_MISSING_CDI_KEY, ERR_PAYMENT_CDI_INVALID);
    if ( bCustIdBlank )
    {
        // customer id is blank, so last name must be entered
        GetKeyValue(&ptr, "CLT", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_CLT_KEY);
        if ( szTmp[0] == 0 )
            throw new
CWEBCLNT_ERR( ERR_PAYMENT_MISSING_CID_CLT );
        _strupr( szTmp );
        if ( strlen(pPaymentData->c_last) >
LAST_NAME_LEN )
            throw new
CWEBCLNT_ERR( ERR_PAYMENT_LAST_NAME_TO_LONG );
        strcpy(pPaymentData->c_last, szTmp);
    }
    else
    {
        // parse customer id and verify that last name was
NOT entered
        GetKeyValue(&ptr, "CLT", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_CLT_KEY);
        if ( szTmp[0] != 0 )
            throw new
CWEBCLNT_ERR( ERR_PAYMENT_CID_AND_CLT );
    }

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

/* FUNCTION: GetOrderStatusData
*
* PURPOSE: This function extracts and validates the payment form
data from an http command string.
*/
void GetOrderStatusData(LPSTR lpszQueryString, ORDER_STATUS_DATA
*pOrderStatusData)
{
    char szTmp[26];
    char *ptr = lpszQueryString;

    pOrderStatusData->d_id = GetIntKeyValue(&ptr, "DID",
ERR_ORDERSTATUS_MISSING_DID_KEY,
ERR_ORDERSTATUS_DID_INVALID);

    GetKeyValue(&ptr, "CID", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CID_KEY);
    if ( szTmp[0] == 0 )
    {
        // customer id is blank, so last name must be entered
        pOrderStatusData->c_id = 0;
        GetKeyValue(&ptr, "CLT", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CLT_KEY);
        if ( szTmp[0] == 0 )
            throw new
CWEBCLNT_ERR( ERR_ORDERSTATUS_MISSING_CID_CLT );
        _strupr( szTmp );
        if ( strlen(pOrderStatusData->c_last) >
LAST_NAME_LEN )
            throw new
CWEBCLNT_ERR( ERR_ORDERSTATUS_CLT_RANGE );
        strcpy(pOrderStatusData->c_last, szTmp);
}

```

```

}
else
{
    // parse customer id and verify that last name was
NOT entered
    if (!IsNumeric(szTmp))
        throw new
CWEBCLNT_ERR( ERR_ORDERSTATUS_CID_INVALID );
    pOrderStatusData->c_id = atoi(szTmp);
    GetKeyValue(&ptr, "CLT", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CLT_KEY);
    if ( szTmp[0] != 0 )
        throw new
CWEBCLNT_ERR( ERR_ORDERSTATUS_CID_AND_CLT );
}

}

/* FUNCTION: BOOL IsNumeric(char *ptr)
*
* PURPOSE: This function determines if a string is numeric. It fails
if any characters other
*
* ARGUMENTS: char *ptr
pointer to string to check.
*
* RETURNS: BOOL FALSE if string is not
all numeric
*
TRUE if string contains only numeric characters i.e. '0' - '9'
*/
BOOL IsNumeric(char *ptr)
{
    if ( *ptr == 0 )
        return FALSE;

    while( *ptr && isdigit(*ptr) )
        ptr++;
    return ( !*ptr );
}

/* FUNCTION: BOOL IsDecimal(char *ptr)
*
* PURPOSE: This function determines if a string is a non-negative
decimal value.
*
* It fails if any characters other than a series of numbers followed by
a decimal point, another series of
*
* ARGUMENTS: char *ptr
pointer to string to check.
*
* RETURNS: BOOL FALSE if string is not
a valid non-negative decimal value
*
TRUE if string is OK
*/
BOOL IsDecimal(char *ptr)
{
    char *dotptr;
    BOOL bValid;

    if ( *ptr == 0 )
        return FALSE;

    // find decimal point
    dotptr = strchr( ptr, '.' );
    if ( dotptr == NULL )
        // no decimal point, so just check for numeric

```

```

return IsNumeric(ptr);
*dotptr = 0; // temporarily replace decimal with a terminator

if ( *ptr != 0 )
    bValid = IsNumeric(ptr);
// string starts with decimal point
else if (*(dotptr+1) == 0)
    return FALSE; // nothing but a decimal point is bad
else
    bValid = TRUE;

if (*(dotptr+1) != 0)
    // check text after decimal point
    bValid &= IsNumeric(dotptr+1);

*dotptr = '.'; // replace decimal point
return bValid;
}

```

isapi_dll/src/resource.h

```

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

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

```

db_dblib_dll/src/tpcc_dblib.h

```

/* FILE: TPCC_DBLIB.H
* Microsoft TPC-C Kit Ver.
4.2.000
* 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:
* 4.2.000 - updated rev number to match kit
*/
#pragma once

#ifndef PDBPROCESS
#define DBPROCESS void // dbprocess structure type
typedef DBPROCESS * PDBPROCESS;
#endif

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifndef DDECL
#define DDECL __declspec( dllimport )
#endif

```



```

class CSQLERR : public CBaseErr
{
public:
    CSQLERR(void)
    {
        m_msgno = 0;
        m_msgstate = 0;
        m_severity = 0;
        m_msgtext = NULL;
    };

    ~CSQLERR()
    {
        delete [] m_msgtext;
    };

    int m_msgno;
    int m_msgstate;
    int m_severity;
    char *m_msgtext;

    int ErrorType() {return ERR_TYPE_SQL;};
    int ErrorNum() {return m_msgno;};
    char *ErrorText() {return m_msgtext;};
};

class CDBLIBERR : public CBaseErr
{
public:
    enum ACTION
    {
        eNone,
        eUnknown,
        eLogin,
        // error from dblogin
        eDbOpen,
        // error from dbopen
        eDbUse,
        // error from dbuse
        eDbSqlExec,
        // error from dbsqlexec
        eDbSet,
        // error from one of the dbset* routines
        eDbNextRow,
        // error from dbnextrow
        eWrongRowCount,
        // more or less rows returned than expected
        eWrongNumCols,
        // more or less columns returned than expected
        eDbResults,
        // error from dbresults
        eDbRpcExec,
        // error from dbrpcexec
        eDbSetMaxProcs,
        // error from dbsetmaxprocs
        eDbProcHandler,
        // error from either dbprocerrhandle or 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 ErrorType() {return ERR_TYPE_DBLIB;};
int ErrorNum() {return m_dberror;};
char *ErrorText() {return m_dberrstr;};
};

class CTPCC_DBLIB_ERR : public CBaseErr
{
public:
    enum CTPCC_DBLIB_ERRS
    {
        ERR_WRONG_SP_VERSION = 1,
        // "Wrong version of stored procs on database server"
        ERR_INVALID_CUST,
        // "Invalid Customer id,name."
        ERR_NO_SUCH_ORDER,
        // "No orders found for customer."
        ERR_RETRIED_TRANS,
        // "Retries before transaction succeeded."
    };

    CTPCC_DBLIB_ERR( int iErr ) { m_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 ErrorType() {return ERR_TYPE_TPCC_DBLIB;};
    int ErrorNum() {return m_erno;};

    char *ErrorText();
};

class DIIDecl CTPCC_DBLIB : public CTPCC_BASE
{
private:
    // declare variables and private functions here...
    PDBPROCESS m_dbproc;
    CDBLIBERR *m_DbLibErr;
    // not allocated until needed (maybe never)
    CSQLERR *m_SqlErr;
    // not allocated until needed (maybe never)
    int m_MaxRetries; // retry count on deadlock

    void DiscardNextRows(int iExpectedCount);
    void DiscardNextResults(int iExpectedCount);
    void ThrowError( CDBLIBERR::ACTION eAction );
    void ResetError();

    union
    {
        NEW_ORDER_DATA
    };
};

```

```

Payment; PAYMENT_DATA
Delivery; DELIVERY_DATA
}
public:
    CTPCC_DBLIB(LPCSTR szServer, LPCSTR szUser,
LPCSTR szPassword, LPCSTR szHost, LPCSTR szDatabase );
    ~CTPCC_DBLIB(void);

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

    void NewOrder ();
    void Payment ();
    void Delivery ();
    void StockLevel ();
    void OrderStatus ();

    // these are public because they must be called from
the dblib_err_handler and msg_hangler
    // outside of the class
    void SetDbLibError(int severity, int dberr, int oserr,
LPCSTR dberrstr, LPCSTR oserrstr);
    void SetSqlError( int msgno, int msgstate, int severity,
LPCSTR msgtext );
};

extern "C" DIIDecl 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);

```

db_dblib_dll\src\tpcc_dblib.cpp

```

/* FILE: TPCC_DBLIB.CPP
Microsoft TPC-C Kit Ver.
4.20.000 Copyright Microsoft, 1999
All Rights Reserved
Version 4.10.000 audited
by Richard Gimarc, Performance Metrics, 3/17/99
PURPOSE: Implements dblib calls for TPC-C txns.
Contact: Charles Levine (clevine@microsoft.com)
Change history:
4.20.000 - updated rev number to match kit
4.10.001 - not deleting error class in catch handler on
deadlock retry;
not a functional bug, but

```

```

a memory leak
*
- had to tweak some
declarations to compile with latest SDK; no functional change
*/

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

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

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

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

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

#define DEFCLPACKSIZE 4096

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

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

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

BOOL APIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call,
LPVOID lpReserved)
{
    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:
            DisableThreadLibraryCalls(hModule);
            dbinit(); // initialize
            dblib
            break;

        case DLL_PROCESS_DETACH:
            dbexit(); // close all
            dblib structures/connections
            break;

        default:
            /* nothing */;
    }
    return TRUE;
}

int err_handler(DBPROCESS *dbproc, int severity, int dberr, int oserr, LPCSTR
dberrstr, LPCSTR oserrstr)
{
    CTPCC_DBLIB *pConn;

    assert(dbproc != NULL);
    pConn = (CTPCC_DBLIB*)dbgetuserdata(dbproc);

    if (pConn != NULL)
    {
        pConn->SetDbLibError( severity, dberr, oserr, dberrstr,

```

```

oserrstr );
    }
    return INT_CANCEL;
}

/* FUNCTION: int msg_handler(DBPROCESS *dbproc, DBINT msgno, int
msgstate, int severity, char *msgtext)
*
* PURPOSE: This function handles DB-Library SQL Server error
messages
*
* ARGUMENTS: DBPROCESS *dbproc
DBPROCESS id pointer
DBINT msgno message number
int msgstate message state
int severity message severity
char *msgtext printable message description
*
* RETURNS: int
INT_CONTINUE continue if error is SQLETIME else
INT_CANCEL action
INT_CANCEL cancel
operation
*
* COMMENTS: This function also sets the dead lock dbproc variable
if necessary.
*/

// typedef INT (SQLAPI *DBMSGHANDLE_PROC)(PDBPROCESS, DBINT, INT,
INT, LPCSTR, LPCSTR, LPCSTR, DBUSMALLINT);

int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int severity,
LPCSTR msgtext,
LPCSTR srvname, LPCSTR procname, DBUSMALLINT line)
{
    CTPCC_DBLIB *pConn;

    assert(dbproc != NULL);
    pConn = (CTPCC_DBLIB*)dbgetuserdata(dbproc);

    if (pConn != NULL)
    {
        pConn->SetSqlError( msgno, msgstate, severity,
msgtext );
    }
    return 0;
}

/* FUNCTION: void UtilStrCpy(char * pDest, char * pSrc, int n)
*
* PURPOSE: This function copies n characters from string pSrc to
pDst and places a
*
* ARGUMENTS: char *pDest
destination string pointer
char *pSrc source string pointer
int n number of characters to
copy
*
* RETURNS: None
*/

```

```

* COMMENTS: Unlike strncpy this function ensures that the result
string is
*
* always null terminated.
*/

inline static void UtilStrCpy(char * pDest, const BYTE * pSrc, int n)
{
    strncpy(pDest, (char *)pSrc, n);
    pDest[n] = '\0';

    return;
}

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
*
*/
char* CTPCC_DBLIB_ERR::ErrorText(void)
{
    int i;

    static SERRORMSG errorMsgs[] =
    {
        { ERR_WRONG_SP_VERSION,
        "Wrong version of stored procs on database server" },
        { ERR_INVALID_CUST,
        "Invalid Customer id,name." },
        { ERR_NO_SUCH_ORDER,
        "No orders found for customer." },
        { ERR_RETRIED_TRANS,
        "Retries
before transaction succeeded." },
        { 0,
        "" }
    };

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

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

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

CTPCC_DBLIB::CTPCC_DBLIB (
LPCSTR szServer, // name of SQL server
LPCSTR szUser, // user name

```

```

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

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

deadlock
    m_MaxRetries = 10;          // how many retries on

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

        ThrowError(CDBLIBERR::eDbSetMaxProcs);
    }

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

    // register error and message handler functions
    if (dbprocerrhandle(login, err_handler) == NULL)
        ThrowError(CDBLIBERR::eDbProcHandler);

    if (dbprocmsghandle(login, msg_handler) == NULL)
        ThrowError(CDBLIBERR::eDbProcHandler);

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

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

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

    m_dbproc = dbopen(login, szServer);

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

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

handler
    // save address of class instance so that the message and error

    // can get to data.
    dbsetuserdata(m_dbproc, (LPVOID)this);

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

    dbcmd(m_dbproc, "set nocount on");

```

```

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

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

DiscardNextResults(2);

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

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

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

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

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

DiscardNextRows(0);
DiscardNextResults(0);
}

CTPCC_DBLIB::~~CTPCC_DBLIB( void )
{
    // close db connection and deallocate resources
    dbclose(m_dbproc);
    InterlockedDecrement( &iConnectionCount );
    if (m_DbLibErr != NULL)
        delete m_DbLibErr;
    if (m_SqlErr != NULL)
        delete m_SqlErr;
}

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

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

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

```

```

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

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

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

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

    // check for SQL Server error first; if yes, throw it and ignore any
DBLib error.
    if (m_SqlErr != NULL)
    {
        CSQLERR *pSqlErr;
        pSqlErr = m_SqlErr;
        m_SqlErr = NULL; // clear our pointer to
instance; catch handler will delete
        throw pSqlErr;
    }

    CDBLIBERR *pDbLibErr;
    if (m_DbLibErr == NULL)
        // this case isn't expected to happen, since it means
that an error was returned
        // but the error handlers were not called.
        pDbLibErr = new CDBLIBERR(eAction);
    else
    {
        pDbLibErr = m_DbLibErr;
        pDbLibErr->m_eAction = eAction;
        m_DbLibErr = NULL; // clear our
pointer to instance; catch handler will delete
    }

    throw pDbLibErr;
}

// Read and discard rows until no more. Throw an exception if number of rows
read doesn't
// match number of rows expected. The row count will be ignored if the
expected count value
// passed in is negative. A typical use of this routine is to verify that there are no
more
// rows to be read.
void CTPCC_DBLIB::DiscardNextRows(int iExpectedCount)
{
    int RETCODE rc;
    int iRowsRead = 0;

    while (TRUE)
    {
        rc = dbnextrow(m_dbproc);
        if (rc == NO_MORE_ROWS)
            break;
        if (rc == FAIL)
            {

```

```

        if (iExpectedCount >= 0)
            ThrowError(CDBLIBERR::eDbNextRow);
        else
            break;
    }
    iRowsRead++;
}

if ((iExpectedCount >= 0) &&
    (iExpectedCount != iRowsRead))
    ThrowError(CDBLIBERR::eWrongRowCount);
}

// Read and discard results until no more. Throw an exception if number of
// result sets read doesn't
// match number expected. The result set count will be ignored if the expected
// count value
// passed in is negative. A typical use of this routine is to verify that there are no
// more
// result sets to be read.
void CTPCC_DBLIB::DiscardNextResults(int iExpectedCount)
{
    int 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::eDbResults);
            else
                break;
        }
        DiscardNextRows(1);
        iResultsRead++;
    }

    if ((iExpectedCount >= 0) &&
        (iExpectedCount != iResultsRead))
        ThrowError(CDBLIBERR::eWrongRowCount);
}

void CTPCC_DBLIB::StockLevel()
{
    int const BYTE *pData;
    int iTryCount = 0;

    ResetError();

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

            dbrpcparam(m_dbproc, NULL, 0,
SQLINT2, -1, -1, (BYTE *) &m_txn.StockLevel.w_id); // @w_id
smallint
            dbrpcparam(m_dbproc, NULL, 0,
SQLINT1, -1, -1, (BYTE *) &m_txn.StockLevel.d_id); // @d_id
tinyint
            dbrpcparam(m_dbproc, NULL, 0,
SQLINT2, -1, -1, (BYTE *) &m_txn.StockLevel.threshold); // @threshold
smallint

```

```

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

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

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

        if (pData=dbdata(m_dbproc, 1))
            m_txn.StockLevel.low_stock = *((long *) pData);

        DiscardNextRows(0);
        DiscardNextResults(0);

        m_txn.StockLevel.exec_status_code =
eOK;

        return;
    }
    catch (CSQLERR *e)
    {
        if ((e->m_msgno == 1205 ||
            (e->m_msgno ==
iErrOleDbProvider &&
sErrTimeoutExpired) != NULL)) &&
            (++iTryCount <=
iMaxRetries))
        {
            // hit deadlock; backoff for
            increasingly longer period
            delete e;
            Sleep(10 * iTryCount);
        }
        else
            throw;
    }
} // while (TRUE)

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

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

    int iTryCount = 0;
    const BYTE *pData;

    ResetError();

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

            dbrpcparam(m_dbproc, NULL, 0,
SQLINT2, -1, -1, (BYTE *) &m_txn.NewOrder.w_id);

```

```

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

            // check whether any order lines are for
            a remote warehouse
            m_txn.NewOrder.o_all_local = 1;
            for (i = 0; i < m_txn.NewOrder.o_o_cnt;
i++)
            {
                if
                (m_txn.NewOrder.OL[i].ol_supply_w_id != m_txn.NewOrder.w_id)
                {
                    m_txn.NewOrder.o_all_local = 0; // at least one remote warehouse
                    break;
                }
            }
            dbrpcparam(m_dbproc, NULL, 0,
SQLINT1, -1, -1, (BYTE *) &m_txn.NewOrder.o_all_local);

            for (i = 0; i < m_txn.NewOrder.o_o_cnt;
i++)
            {
                dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.NewOrder.OL[i].ol_i_id);
                dbrpcparam(m_dbproc,
NULL, 0, SQLINT2, -1, -1, (BYTE *) &m_txn.NewOrder.OL[i].ol_supply_w_id);
                dbrpcparam(m_dbproc,
NULL, 0, SQLINT2, -1, -1, (BYTE *) &m_txn.NewOrder.OL[i].ol_quantity);
            }

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

            // Get order line results
            m_txn.NewOrder.total_amount = 0;
            for (i = 0; i < m_txn.NewOrder.o_o_cnt;
i++)
            {
                if (dbresults(m_dbproc) !=
SUCCEED)
                    ThrowError(CDBLIBERR::eDbResults);

                if
                (dbnumcols(m_dbproc) != 5)
                    ThrowError(CDBLIBERR::eWrongNumCols);

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

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

                if (pData=dbdata(m_dbproc, 2))
                    m_txn.NewOrder.OL[i].ol_stock = *((DBSMALLINT *) pData);

                if (pData=dbdata(m_dbproc, 3))

```

```

        UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_generic, pData,
        dbdatlen(m_dbproc, 3));
        if(pData=dbdata(m_dbproc, 4))
            dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
            dbdatlen(m_dbproc,4),
            SQLFLT8,
            (BYTE *)&m_txn.NewOrder.OL[i].ol_i_price, 8);
        if(pData=dbdata(m_dbproc, 5))
            dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
            dbdatlen(m_dbproc,5),
            SQLFLT8,
            (BYTE *)&m_txn.NewOrder.OL[i].ol_amount, 8);

        m_txn.NewOrder.total_amount = m_txn.NewOrder.total_amount +
        m_txn.NewOrder.OL[i].ol_amount;
    }
    DiscardNextRows(0);

    // get remaining values for w_tax, d_tax,
    o_id, c_last, c_discount, c_credit, o_entry_d, commit_flag
    if (dbresults(m_dbproc) != SUCCEED)
        ThrowError(CDBLIBERR::eDbResults);
    if (dbnextrow(m_dbproc) != REG_ROW)
        ThrowError(CDBLIBERR::eDbNextRow);
    if (dbnumcols(m_dbproc) != 8)
        ThrowError(CDBLIBERR::eWrongNumCols);
    if (pData=dbdata(m_dbproc, 1))
        dbconvert(m_dbproc,
        SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,1), SQLFLT8, (BYTE
        *)&m_txn.NewOrder.w_tax, 8)
    if (pData=dbdata(m_dbproc, 2))
        dbconvert(m_dbproc,
        SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,2), SQLFLT8, (BYTE
        *)&m_txn.NewOrder.d_tax, 8);
    if (pData=dbdata(m_dbproc, 3))
        m_txn.NewOrder.o_id =
    *((DBINT *) pData);
    if (pData=dbdata(m_dbproc, 4))
        UtilStrCpy(m_txn.NewOrder.c_last, pData, dbdatlen(m_dbproc, 4));
    if (pData=dbdata(m_dbproc, 5))
        dbconvert(m_dbproc,
        SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,5), SQLFLT8, (BYTE
        *)&m_txn.NewOrder.c_discount, 8);
    if (pData=dbdata(m_dbproc, 6))
        UtilStrCpy(m_txn.NewOrder.c_credit, pData, dbdatlen(m_dbproc,
        6));
    if (pData=dbdata(m_dbproc, 7))
    {
        datetime =
    *((DBDATETIME *) pData);
        dbdatecrack(m_dbproc,

```

```

        &daterec, &datetime);
        m_txn.NewOrder.o_entry_d.year = daterec.year;
        m_txn.NewOrder.o_entry_d.month = daterec.month;
        m_txn.NewOrder.o_entry_d.day = daterec.day;
        m_txn.NewOrder.o_entry_d.hour = daterec.hour;
        m_txn.NewOrder.o_entry_d.minute = daterec.minute;
        m_txn.NewOrder.o_entry_d.second = daterec.second;
    }
    if (pData=dbdata(m_dbproc, 8))
        commit_flag =
    DiscardNextRows(0);
    DiscardNextResults(0);
    if (commit_flag == 1)
    {
        m_txn.NewOrder.total_amount *= ((1 + m_txn.NewOrder.w_tax +
        m_txn.NewOrder.d_tax) * (1 - m_txn.NewOrder.c_discount));
        m_txn.NewOrder.exec_status_code = eOK;
    }
    else
        m_txn.NewOrder.exec_status_code = eInvalidItem;
    return;
}
catch (CSQLERR *e)
{
    if ((e->m_msgno == 1205 ||
    (e->m_msgno ==
    iErrOleDbProvider &&
    sErrTimeoutExpired) != NULL) &&
    (++iTryCount <=
    iMaxRetries))
    {
        // hit deadlock; backoff for
        delete e;
        Sleep(10 * iTryCount);
    }
    else
        throw;
}
} while (TRUE)

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

void CTPCC_DBLIB::Payment()
{
    DBDATETIME datetime;
    DBDATERECEC daterec;

    int iTryCount = 0;
    const BYTE *pData;
    ResetError();

```

```

        while (TRUE)
        {
            try
            {
                dbrpcinit(m_dbproc, "tpcc_payment", 0);
                dbrpcparam(m_dbproc, NULL, 0,
                SQLINT2,-1,-1, (BYTE *) &m_txn.Payment.w_id);
                dbrpcparam(m_dbproc, NULL, 0,
                SQLINT2,-1,-1, (BYTE *) &m_txn.Payment.c_w_id);
                dbrpcparam(m_dbproc, NULL, 0,
                SQLFLT8,-1,-1, (BYTE *) &m_txn.Payment.h_amount);
                dbrpcparam(m_dbproc, NULL, 0,
                SQLINT1,-1,-1, (BYTE *) &m_txn.Payment.d_id);
                dbrpcparam(m_dbproc, NULL, 0,
                SQLINT1,-1,-1, (BYTE *) &m_txn.Payment.c_d_id);
                dbrpcparam(m_dbproc, NULL, 0,
                SQLINT4,-1,-1, (BYTE *) &m_txn.Payment.c_id);

                // if customer id is zero, then payment is
                by name
                if (m_txn.Payment.c_id == 0)
                    dbrpcparam(m_dbproc,
                NULL, 0, SQLCHAR,-1, strlen(m_txn.Payment.c_last), (unsigned char
                *)&m_txn.Payment.c_last);
                if (dbrpcexec(m_dbproc) == FAIL)
                    ThrowError(CDBLIBERR::eDbRpcExec);
                if (dbresults(m_dbproc) != SUCCEED)
                    ThrowError(CDBLIBERR::eDbResults);
                if (dbnextrow(m_dbproc) != REG_ROW)
                    ThrowError(CDBLIBERR::eDbNextRow);
                if (dbnumcols(m_dbproc) != 27)
                    ThrowError(CDBLIBERR::eWrongNumCols);
                if (pData=dbdata(m_dbproc, 1))
                    m_txn.Payment.c_id =
                *((DBINT *) pData);
                if (pData=dbdata(m_dbproc, 2))
                    UtilStrCpy(m_txn.Payment.c_last, pData, dbdatlen(m_dbproc, 2));
                if (pData=dbdata(m_dbproc, 3))
                {
                    datetime =
                dbdatecrack(m_dbproc,
                &daterec, &datetime);
                    m_txn.Payment.h_date.year = daterec.year;
                    m_txn.Payment.h_date.month = daterec.month;
                    m_txn.Payment.h_date.day = daterec.day;
                    m_txn.Payment.h_date.hour = daterec.hour;
                    m_txn.Payment.h_date.minute = daterec.minute;
                    m_txn.Payment.h_date.second = daterec.second;
                }
                if (pData=dbdata(m_dbproc, 4))
                    UtilStrCpy(m_txn.Payment.w_street_1, pData, dbdatlen(m_dbproc,
                4));
            }
        }
    }

```

```

        if (pData=dbdata(m_dbproc, 5))
5)); UtilStrCpy(m_txn.Payment.w_street_2, pData, dbdatlen(m_dbproc,
        if (pData=dbdata(m_dbproc, 6))
UtilStrCpy(m_txn.Payment.w_city, pData, dbdatlen(m_dbproc, 6));
        if (pData=dbdata(m_dbproc, 7))
7)); UtilStrCpy(m_txn.Payment.w_state, pData, dbdatlen(m_dbproc,
        if (pData=dbdata(m_dbproc, 8))
UtilStrCpy(m_txn.Payment.w_zip, pData, dbdatlen(m_dbproc, 8));
        if (pData=dbdata(m_dbproc, 9))
9)); UtilStrCpy(m_txn.Payment.d_street_1, pData, dbdatlen(m_dbproc,
        if (pData=dbdata(m_dbproc, 10))
10)); UtilStrCpy(m_txn.Payment.d_street_2, pData, dbdatlen(m_dbproc,
        if (pData=dbdata(m_dbproc, 11))
UtilStrCpy(m_txn.Payment.d_city, pData, dbdatlen(m_dbproc, 11));
        if (pData=dbdata(m_dbproc, 12))
12)); UtilStrCpy(m_txn.Payment.d_state, pData, dbdatlen(m_dbproc,
        if (pData=dbdata(m_dbproc, 13))
UtilStrCpy(m_txn.Payment.d_zip, pData, dbdatlen(m_dbproc, 13));
        if (pData=dbdata(m_dbproc, 14))
UtilStrCpy(m_txn.Payment.c_first, pData, dbdatlen(m_dbproc, 14));
        if (pData=dbdata(m_dbproc, 15))
15)); UtilStrCpy(m_txn.Payment.c_middle, pData, dbdatlen(m_dbproc,
        if (pData=dbdata(m_dbproc, 16))
16)); UtilStrCpy(m_txn.Payment.c_street_1, pData, dbdatlen(m_dbproc,
        if (pData=dbdata(m_dbproc, 17))
17)); UtilStrCpy(m_txn.Payment.c_street_2, pData, dbdatlen(m_dbproc,
        if (pData=dbdata(m_dbproc, 18))
UtilStrCpy(m_txn.Payment.c_city, pData, dbdatlen(m_dbproc, 18));
        if (pData=dbdata(m_dbproc, 19))
19)); UtilStrCpy(m_txn.Payment.c_state, pData, dbdatlen(m_dbproc,
        if (pData=dbdata(m_dbproc, 20))
UtilStrCpy(m_txn.Payment.c_zip, pData, dbdatlen(m_dbproc, 20));
        if (pData=dbdata(m_dbproc, 21))
21)); UtilStrCpy(m_txn.Payment.c_phone, pData, dbdatlen(m_dbproc,
        if (pData=dbdata(m_dbproc, 22))
{
        datetime =
*((DBDATETIME *) pData);
        dbdatecrack(m_dbproc,
&daterec, &datetime);
        m_txn.Payment.c_since.year = daterec.year;
        m_txn.Payment.c_since.month = daterec.month;

```

```

        m_txn.Payment.c_since.day = daterec.day;
        m_txn.Payment.c_since.hour = daterec.hour;
        m_txn.Payment.c_since.minute = daterec.minute;
        m_txn.Payment.c_since.second = daterec.second;
        }
        if(pData=dbdata(m_dbproc, 23))
23)); UtilStrCpy(m_txn.Payment.c_credit, pData, dbdatlen(m_dbproc,
        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))
27)); UtilStrCpy(m_txn.Payment.c_data, pData, dbdatlen(m_dbproc,
        DiscardNextRows(0);
        DiscardNextResults(0);
        if (m_txn.Payment.c_id == 0)
        throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
        else
        m_txn.Payment.exec_status_code = eOK;
        return;
        }
        catch (CSQLERR *e)
        {
        if ((e->m_msgno == 1205 ||
(e->m_msgno ==
iErrOleDbProvider &&
sErrTimeoutExpired) != NULL) &&
strstr(e->m_msgtext,
(++iTryCount <=
iMaxRetries))
{
        // hit deadlock; backoff for
        increasingly longer period
        delete e;
        Sleep(10 * iTryCount);
        }
        else
        throw;
        }
        // while (TRUE)
        // if (iTryCount)
        // throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
        }
        void CTPCC_DBLIB::OrderStatus()
        {

```

```

        int
        DBDATETIME          datetime;
        DBDATEREK          daterec;
        int
        RETCODE            rc;
        const BYTE *pData;
        iTryCount = 0;
        ResetError();
        while (TRUE)
        {
        try
        {
        dbrpcinit(m_dbproc, "tpcc_orderstatus",
0);
        dbrpcparam(m_dbproc, NULL, 0,
SQLINT2, -1, -1, (BYTE *) &m_txn.OrderStatus.w_id);
        dbrpcparam(m_dbproc, NULL, 0,
SQLINT1, -1, -1, (BYTE *) &m_txn.OrderStatus.d_id);
        dbrpcparam(m_dbproc, NULL, 0,
SQLINT4, -1, -1, (BYTE *) &m_txn.OrderStatus.c_id);
        // if customer id is zero, then order
        status is by name
        if (m_txn.OrderStatus.c_id == 0)
        dbrpcparam(m_dbproc,
NULL, 0, SQLCHAR, -1, strlen(m_txn.OrderStatus.c_last), (unsigned char
*)&m_txn.OrderStatus.c_last);
        if (dbrpcexec(m_dbproc) == FAIL)
        ThrowError(CDBLIBERR::eDbRpcExec);
        // Get order lines
        if (dbresults(m_dbproc) != SUCCEEDED)
        {
        if ((m_DbLibErr == NULL)
&& (m_SqlErr == NULL))
        throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
        else
        ThrowError(CDBLIBERR::eDbResults);
        }
        if (dbnumcols(m_dbproc) != 5)
        ThrowError(CDBLIBERR::eWrongNumCols);
        i = 0;
        while (TRUE)
        {
        rc =
dbrpcrow(m_dbproc);
        if (rc ==
NO_MORE_ROWS)
        break;
        if (rc != REG_ROW)
        ThrowError(CDBLIBERR::eDbNextRow);
        if (pData=dbdata(m_dbproc, 1))
        m_txn.OrderStatus.OL[i].ol_supply_w_id = (*DBSMALLINT *)
pData);
        if (pData=dbdata(m_dbproc, 2))

```

```

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

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

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

if (dbnumcols(m_dbproc) != 8)
ThrowError(CDBLIBERR::eWrongNumCols);

if(pData=dbdata(m_dbproc, 1))
m_txn.OrderStatus.c_id =
*((DBINT *) pData);
if(pData=dbdata(m_dbproc, 2))
UtilStrCpy(m_txn.OrderStatus.c_last, pData,
dbdatlen(m_dbproc,2));
if(pData=dbdata(m_dbproc, 3))
UtilStrCpy(m_txn.OrderStatus.c_first, pData,
dbdatlen(m_dbproc,3));
if(pData=dbdata(m_dbproc, 4))
UtilStrCpy(m_txn.OrderStatus.c_middle, pData,
dbdatlen(m_dbproc, 4));
if(pData=dbdata(m_dbproc, 5))
{
datetime =
*((DBDATETIME *) pData);
dbdatecrack(m_dbproc,
&daterec, &datetime);
m_txn.OrderStatus.o_entry_d.year = daterec.year;

```

```

m_txn.OrderStatus.o_entry_d.month = daterec.month;
m_txn.OrderStatus.o_entry_d.day = daterec.day;
m_txn.OrderStatus.o_entry_d.hour = daterec.hour;
m_txn.OrderStatus.o_entry_d.minute = daterec.minute;
m_txn.OrderStatus.o_entry_d.second = daterec.second;
}
if(pData=dbdata(m_dbproc, 6))
m_txn.OrderStatus.o_carrier_id = *((DBSMALLINT *) pData);
if(pData=dbdata(m_dbproc, 7))
dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData,dbdatlen(m_dbproc,7),
SQLFLT8, (BYTE *)&m_txn.OrderStatus.c_balance, 8);
if(pData=dbdata(m_dbproc, 8))
m_txn.OrderStatus.o_id =
*((DBINT *) pData);

DiscardNextRows(0);
DiscardNextResults(0);

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

return;
}
catch (CSQLERR *e)
{
if ((e->m_msgno == 1205 ||
(e->m_msgno ==
iErrOleDbProvider &&
sErrTimeoutExpired) != NULL) &&
(++iTryCount <=
iMaxRetries))
{
// hit deadlock; backoff for
increasingly longer period
delete e;
Sleep(10 * iTryCount);
}
else
throw;
}
}
// while (TRUE)
{
if (iTryCount)
throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}
}

void CTPCC_DBLIB::Delivery()
{
int i;
int iTryCount = 0;
const BYTE *pData;

```

```

ResetError();
while (TRUE)
{
try
{
dbrpcinit(m_dbproc, "tpcc_delivery", 0);
dbrpcparam(m_dbproc, NULL, 0,
SQLINT2, -1, -1, (BYTE *) &m_txn.Delivery.w_id);
dbrpcparam(m_dbproc, NULL, 0,
SQLINT1, -1, -1, (BYTE *) &m_txn.Delivery.o_carrier_id);
if (dbrpcexec(m_dbproc) == FAIL)
ThrowError(CDBLIBERR::eDbRpcExec);
if (dbresults(m_dbproc) != SUCCEED)
ThrowError(CDBLIBERR::eDbResults);
if (dbnextrow(m_dbproc) != REG_ROW)
ThrowError(CDBLIBERR::eDbNextRow);
if (dbnumcols(m_dbproc) != 10)
ThrowError(CDBLIBERR::eWrongNumCols);
for (i=0; i<10; i++)
{
if (pData =
dbdata(m_dbproc, i+1))
m_txn.Delivery.o_id[i] = *((DBINT *)pData);
}
DiscardNextRows(0);
DiscardNextResults(0);
m_txn.Delivery.exec_status_code =
eOK;
return;
}
catch (CSQLERR *e)
{
if ((e->m_msgno == 1205 ||
(e->m_msgno ==
iErrOleDbProvider &&
sErrTimeoutExpired) != NULL) &&
(++iTryCount <=
iMaxRetries))
{
// hit deadlock; backoff for
increasingly longer period
delete e;
Sleep(10 * iTryCount);
}
else
throw;
}
}
// while (TRUE)
{
if (iTryCount)
throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}
}

void CTPCC_DBLIB::ResetError()

```

```

{
    if (m_DbLibErr != NULL)
    {
        delete m_DbLibErr;
        m_DbLibErr = (CDBLIBERR*)NULL;
    }

    if (m_SqlErr != NULL)
    {
        delete m_SqlErr;
        m_SqlErr = (CSQLERR*)NULL;
    }

    return;
}

```

tm_com_dll\src\tm_com.h

```

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

#pragma once

#include <stdio.h>
#include "..\..\tpcc_com_ps\src\tpcc_com_ps.h"

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifdef DllDecl
#define DllDecl __declspec( dllimport )
#else
#define DllDecl
#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
    CCOMERR( int iErrorType, int iError )
    {
        m_iErrorType = iErrorType;
        m_iError = iError;
        m_hr = S_OK;
    }

    int m_hr;
    int m_iErrorType;
    int m_iError;
}

```

```

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

int ErrorNum() {return m_hr;}

char *ErrorText()
{
    if (m_hr == S_OK)
        sprintf( m_szErrorText,
"Error: Class %d, error # %d", m_iErrorType, m_iError );
    else
        sprintf( m_szErrorText,
"Error: COM HRESULT %x", m_hr );
    return m_szErrorText;
};

class DllDecl CTPCC_COM : public CTPCC_BASE
{
private:
    BOOL m_bSinglePool;

    // COM Interface pointers
    ITPCC*
    m_pNewOrder;
    ITPCC*
    m_pPayment;
    ITPCC*
    m_pStockLevel;
    ITPCC*
    m_pOrderStatus;

    struct COM_DATA
    {
        int ErrorType;
        int error;
        union
        {
            NEW_ORDER_DATA
            PAYMENT_DATA
            DELIVERY_DATA
            STOCK_LEVEL_DATA
            ORDER_STATUS_DATA
        } u;
    } *m_pTxn;

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

    inline PNEW_ORDER_DATA
    BuffAddr_NewOrder() { return &m_pTxn-
>u.NewOrder; };

    inline PPAYMENT_DATA
    BuffAddr_Payment() { return &m_pTxn-
>u.Payment; };

    inline PDELIVERY_DATA
    BuffAddr_Delivery() { return &m_pTxn-

```

```

>u.Delivery; };

    inline PSTOCK_LEVEL_DATA
    BuffAddr_StockLevel() { return &m_pTxn->u.StockLevel; };
    inline PORDER_STATUS_DATA
    BuffAddr_OrderStatus() { return &m_pTxn->u.OrderStatus; };

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

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

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

typedef CTPCC_COM* (TYPE_CTPCC_COM)(BOOL);

```

tm_com_dll\src\tm_com.cpp

```

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

// needed for CoinitializeEx
#define _WIN32_WINNT 0x0400

#include <windows.h>

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

#include "..\..\common\src\trans.h" //tpckit transaction header
#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;

    m_bSinglePool = bSinglePool;

    m_pNewOrder          = NULL;
    m_pPayment            = NULL;
    m_pStockLevel        = NULL;
    m_pOrderStatus       = NULL;

    m_pTxn = (COM_DATA*)CoTaskMemAlloc(sizeof(COM_DATA));
    if (!m_pTxn)
        throw new CCOMERR( E_FAIL );

    hr = CoInitializeEx(NULL, COINIT_MULTITHREADED);
    if (FAILED(hr))
    {
        throw new CCOMERR( hr );
    }

    // create components
    if (m_bSinglePool)
    {
        hr = CoCreateInstance(CLSID_TPCC, NULL,
            CLSCTX_SERVER, IID_ITPCC, (void **)&m_pNewOrder);
        if (FAILED(hr))
            throw new CCOMERR(hr);

        // all txns will use same component
        m_pPayment = m_pNewOrder;
        m_pStockLevel = m_pNewOrder;
        m_pOrderStatus = m_pNewOrder;
    }
    else
    {
        // use different components for each txn

        hr = CoCreateInstance(CLSID_NewOrder, NULL,
            CLSCTX_SERVER, IID_ITPCC, (void **)&m_pNewOrder);
        if (FAILED(hr))
            throw new CCOMERR(hr);

        hr = CoCreateInstance(CLSID_Payment, NULL,
            CLSCTX_SERVER, IID_ITPCC, (void **)&m_pPayment);
        if (FAILED(hr))
            throw new CCOMERR(hr);

        hr = CoCreateInstance(CLSID_StockLevel, NULL,
            CLSCTX_SERVER, IID_ITPCC, (void **)&m_pStockLevel);
        if (FAILED(hr))
            throw new CCOMERR(hr);

        hr = CoCreateInstance(CLSID_OrderStatus, NULL,
            CLSCTX_SERVER, IID_ITPCC, (void **)&m_pOrderStatus);
        if (FAILED(hr))
            throw new CCOMERR(hr);
    }

    // call setcomplete to release each component back into pool
    hr = m_pNewOrder->CallSetComplete();
    if (FAILED(hr))
        throw new CCOMERR(hr);

    if (!m_bSinglePool)
    {
        hr = m_pPayment->CallSetComplete();
        if (FAILED(hr))
            throw new CCOMERR(hr);
    }
}

```

```

        hr = m_pStockLevel->CallSetComplete();
        if (FAILED(hr))
            throw new CCOMERR(hr);

        hr = m_pOrderStatus->CallSetComplete();
        if (FAILED(hr))
            throw new CCOMERR(hr);
    }
}

CTPCC_COM::~CTPCC_COM()
{
    if (m_pTxn)
        CoTaskMemFree(m_pTxn);

    ReleaseInterface(m_pNewOrder);
    if (!m_bSinglePool)
    {
        ReleaseInterface(m_pPayment);
        ReleaseInterface(m_pStockLevel);
        ReleaseInterface(m_pOrderStatus);
    }
    CoUninitialize();
}

void CTPCC_COM::NewOrder()
{
    int iSize = sizeof(COM_DATA);

    HRESULT hr = m_pNewOrder->NewOrder(&iSize, (unsigned
char**)&m_pTxn);
    if (FAILED(hr))
        throw new CCOMERR( hr );

    if ( m_pTxn->ErrorType != ERR_SUCCESS )
        throw new CCOMERR( m_pTxn->ErrorType, m_pTxn-
>error );
}

void CTPCC_COM::Payment()
{
    int iSize = sizeof(COM_DATA);

    HRESULT hr = m_pPayment->Payment(&iSize, (unsigned
char**)&m_pTxn);
    if (FAILED(hr))
        throw new CCOMERR( hr );

    if ( m_pTxn->ErrorType != ERR_SUCCESS )
        throw new CCOMERR( m_pTxn->ErrorType, m_pTxn-
>error );
}

void CTPCC_COM::StockLevel()
{
    int iSize = sizeof(COM_DATA);

    HRESULT hr = m_pStockLevel->StockLevel(&iSize, (unsigned
char**)&m_pTxn);
    if (FAILED(hr))
        throw new CCOMERR( hr );

    if ( m_pTxn->ErrorType != ERR_SUCCESS )
        throw new CCOMERR( m_pTxn->ErrorType, m_pTxn-
>error );
}

void CTPCC_COM::OrderStatus()
{
    int iSize = sizeof(COM_DATA);
}

```

```

        HRESULT hr = m_pOrderStatus->OrderStatus(&iSize, (unsigned
char**)&m_pTxn);
        if (FAILED(hr))
            throw new CCOMERR( hr );

        if ( m_pTxn->ErrorType != ERR_SUCCESS )
            throw new CCOMERR( m_pTxn->ErrorType, m_pTxn-
>error );
    }
}

```

tpcc_com_all\src\tpcc_com_all.rgs

```

HKCR
{
    TPCC.AllTxns.1 = s 'All Txns Class'
    {
        CLSID = s '{122A31282520-11D3-BA71-
00C04FBFE08B}'
        TPCC.AllTxns = s 'TPCC Class'
        {
            CurVer = s 'TPCC.AllTxns.1'
        }
        NoRemove CLSID
        {
            ForceRemove {122A31282520-11D3-BA71-
00C04FBFE08B} = s 'TPCC Class'
            {
                ProgID = s 'TPCC.AllTxns.1'
                VersionIndependentProgID = s
                {
                    InprocServer32 = s '%MODULE%'
                    {
                        val ThreadingModel = s
                    }
                }
            }
        }
    }
}

```

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"

//
// English (U.S.) resources
//
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif // _WIN32

#ifndef APSTUDIO_INVOKED

```

```

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

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

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

#endif // APSTUDIO_INVOKED

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

VS_VERSION_INFO VERSIONINFO
FILEVERSION 1,0,0,1
PRODUCTVERSION 1,0,0,1
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x4L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN
BLOCK "StringFileInfo"
BEGIN
BLOCK "040904B0"
BEGIN
VALUE "CompanyName", "0"
VALUE "FileDescription", "tpcc_com_all Module\0"
VALUE "FileVersion", "1, 0, 0, 1\0"
VALUE "InternalName", "TPCCNEWORDER0"
VALUE "LegalCopyright", "Copyright 1997\0"
VALUE "OriginalFilename", "tpcc_com_all.DLL\0"
VALUE "ProductName", "tpcc_com_all Module\0"
VALUE "ProductVersion", "1, 0, 0, 1\0"
VALUE "OLESelfRegister", "0"
END
END
BLOCK "VarFileInfo"
BEGIN
VALUE "Translation", 0x409, 1200
END
END
#endif // !_MAC

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

```

```

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

```

```

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

```

```

STRINGTABLE DISCARDABLE
BEGIN
IDS_PROJNAME "tpcc_com_all"
END

```

```

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

```

```

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

```

```

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

```

tpcc_com_all\src\tpcc_com_all.idl

```

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

```

```

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

```

```

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

```

```

[
    uuid(122A3117-2520-11D3-BA71-00C04FBFE08B),
    version(1.0),
    helpstring("TPC-C 1.0 Type Library")
]
library TPCClib

```

```

{
    importlib("stdole32.tlb");
    importlib("stdole2.tlb");

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

    [
        uuid(975BAABF-84A7-11D2-BA47-00C04FBFE08B),
        helpstring("NewOrder Class")
    ]
    coclass NewOrder
    {
        [default] interface ITPCC;
    };

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

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

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

```

tpcc_com_all\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 5.03.0280 */
/* at Mon Jan 24 20:00:20 2000
*/
/* Compiler settings for \src\tpcc_com_all.idl:
Oicf (OptLev=2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
error checks: allocation ref bounds_check enum stub_data

```

```

VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADERING( )

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

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

#ifndef __tpcc_com_all_h__
#define __tpcc_com_all_h__

/* Forward Declarations */

#ifndef __TPCC_FWD_DEFINED__
#define __TPCC_FWD_DEFINED__

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

#endif /* __TPCC_FWD_DEFINED__ */

#ifndef __NewOrder_FWD_DEFINED__
#define __NewOrder_FWD_DEFINED__

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

#endif /* __NewOrder_FWD_DEFINED__ */

#ifndef __OrderStatus_FWD_DEFINED__
#define __OrderStatus_FWD_DEFINED__

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

#endif /* __OrderStatus_FWD_DEFINED__ */

#ifndef __Payment_FWD_DEFINED__
#define __Payment_FWD_DEFINED__

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

#endif /* __Payment_FWD_DEFINED__ */

#ifndef __StockLevel_FWD_DEFINED__
#define __StockLevel_FWD_DEFINED__

#ifdef __cplusplus

```

```

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

#endif /* __StockLevel_FWD_DEFINED__ */

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

#ifdef __cplusplus
extern "C"{
#endif

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

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

extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_s_ifspec;

#ifdef __TPCCLib_LIBRARY_DEFINED__
#define __TPCCLib_LIBRARY_DEFINED__

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

EXTERN_C const IID LIBID_TPCCLib;

EXTERN_C const CLSID CLSID_TPCC;

#ifdef __cplusplus

class DECLSPEC_UUID("122A31282520-11D3-BA71-00C04FBFE08B")
TPCC;
#endif

EXTERN_C const CLSID CLSID_NewOrder;

#ifdef __cplusplus

class DECLSPEC_UUID("975BAABF84A7-11D2-BA47-00C04FBFE08B")
NewOrder;
#endif

EXTERN_C const CLSID CLSID_OrderStatus;

#ifdef __cplusplus

class DECLSPEC_UUID("266836ABA50D-11D2-BA4E-00C04FBFE08B")
OrderStatus;
#endif

EXTERN_C const CLSID CLSID_Payment;

#ifdef __cplusplus

class DECLSPEC_UUID("CD02F7EFA4FA-11D2-BA4E-00C04FBFE08B")

```

```

Payment;
#endif

EXTERN_C const CLSID CLSID_StockLevel;

#ifdef __cplusplus

class DECLSPEC_UUID("2668369EA50D-11D2-BA4E-00C04FBFE08B")
StockLevel;
#endif
#endif /* __TPCCLib_LIBRARY_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

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

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

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jan 24 20:00:07 2000
*/
/* Compiler settings for \src\tpcc_com_ps.idl:
Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_exe_c_ext
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADERING( )

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

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

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

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

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

/* Forward Declarations */

#ifndef __ITPCC_FWD_DEFINED__
#define __ITPCC_FWD_DEFINED__
typedef interface ITPCC ITPCC;

```

tpcc_com_all\src\tpcc_com_ps.h

```

#endif /* __ITPCC_FWD_DEFINED__ */

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

#ifdef __cplusplus
extern "C"{
#endif

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

#ifdef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

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

EXTERN_C const IID IID_ITPCC;

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

    MIDL_INTERFACE("FEE6AA284B1-11d2-BA47-00C04BFE08B")
    ITPCC : public IUnknown
    {
    public:
        virtual HRESULT STDMETHODCALLTYPE NewOrder(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR
            *txn) = 0;

        virtual HRESULT STDMETHODCALLTYPE Payment(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR
            *txn) = 0;

        virtual HRESULT STDMETHODCALLTYPE Delivery(
            /* [in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn)
            = 0;

        virtual HRESULT STDMETHODCALLTYPE StockLevel(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR
            *txn) = 0;

        virtual HRESULT STDMETHODCALLTYPE OrderStatus(
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR
            *txn) = 0;

        virtual HRESULT STDMETHODCALLTYPE CallSetComplete( void ) = 0;

    };

#else /* C style interface */

    typedef struct ITPCCVtbl
    {
        BEGIN_INTERFACE

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

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

```

```

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

        HRESULT ( STDMETHODCALLTYPE __RPC_FAR *NewOrder )(
            ITPCC __RPC_FAR * This,
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR
            *txn);

        HRESULT ( STDMETHODCALLTYPE __RPC_FAR *Payment )(
            ITPCC __RPC_FAR * This,
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR
            *txn);

        HRESULT ( STDMETHODCALLTYPE __RPC_FAR *Delivery )(
            ITPCC __RPC_FAR * This,
            /* [in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn);

        HRESULT ( STDMETHODCALLTYPE __RPC_FAR *StockLevel )(
            ITPCC __RPC_FAR * This,
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR
            *txn);

        HRESULT ( STDMETHODCALLTYPE __RPC_FAR *OrderStatus )(
            ITPCC __RPC_FAR * This,
            /* [out][in] */ int __RPC_FAR *iSize,
            /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR
            *txn);

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

    END_INTERFACE
    } ITPCCVtbl;

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

#ifdef COBJMASCROS

#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,iSize,txn) \
    (This->lpVtbl->NewOrder(This,iSize,txn))

#define ITPCC_Payment(This,iSize,txn)\
    (This->lpVtbl->Payment(This,iSize,txn))

#define ITPCC_Delivery(This,iSize,txn) \
    (This->lpVtbl->Delivery(This,iSize,txn))

#define ITPCC_StockLevel(This,iSize,txn) \
    (This->lpVtbl->StockLevel(This,iSize,txn))

#define ITPCC_OrderStatus(This,iSize,txn) \
    (This->lpVtbl->OrderStatus(This,iSize,txn))

```

```

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

#endif /* COBJMASCROS */

/* C style interface */

HRESULT STDMETHODCALLTYPE ITPCC_NewOrder_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn);

void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);

HRESULT STDMETHODCALLTYPE ITPCC_Payment_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn);

void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);

HRESULT STDMETHODCALLTYPE ITPCC_Delivery_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn);

void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);

HRESULT STDMETHODCALLTYPE ITPCC_StockLevel_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn);

void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);

HRESULT STDMETHODCALLTYPE ITPCC_OrderStatus_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn);

void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,

```

```

PRPC_MESSAGE_pRpcMessage,
DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
ITPCC __RPC_FAR * This);

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

#endif /* __ITPCC_INTERFACE_DEFINED__ */

/* Additional Prototypes for ALL interfaces */
/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

```

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
#ifdef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE 202
#define _APS_NEXT_COMMAND_VALUE 32768
#define _APS_NEXT_CONTROL_VALUE 201
#define _APS_NEXT_SYMED_VALUE 106
#endif
#endif

```

tpcc_com_all\src\methods.h

```

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

```

```

* 4.20.000 - first version
*/

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

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

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

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

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

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

static void WriteMessageToEventLog(LPTSTR lpszMsg);

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

```

```

END_COM_MAP()

CTPCC_Common();
~CTPCC_Common();

// ITPCC
public:
HRESULT __stdcall NewOrder( int* iSize,
UCHAR** txn);
HRESULT __stdcall Payment( int* iSize,
UCHAR** txn);
HRESULT __stdcall Delivery( int* iSize,
UCHAR** txn) {return E_NOTIMPL;};
HRESULT __stdcall StockLevel( int* iSize, UCHAR** txn);
HRESULT __stdcall OrderStatus( int* iSize, UCHAR** txn);

HRESULT __stdcall CallSetComplete();

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

// IObjectConstruct
STDMETHODIMP Construct(IDispatch * pUnk);

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

struct COM_DATA
{
int retval;
int error;
union
{
NEW_ORDER_DATA
PAYMENT_DATA
DELIVERY_DATA
STOCK_LEVEL_DATA StockLevel;
ORDER_STATUS_DATA OrderStatus;
};
};

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

BEGIN_COM_MAP(CTPCC)
COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()
};

////////////////////////////////////
// CNewOrder
class CNewOrder :

```

```

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

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

    // ITPCC
    public:
    //      HRESULT __stdcall NewOrder(          int* iSize,
    UCHAR** txn) {return E_NOTIMPL;}
    //      HRESULT __stdcall Payment(          int* iSize,
    UCHAR** txn) {return E_NOTIMPL;}
    //      HRESULT __stdcall StockLevel(      int* iSize, UCHAR** txn)
    {return E_NOTIMPL;}
    //      HRESULT __stdcall OrderStatus(    int* iSize, UCHAR** txn)
    {return E_NOTIMPL;}
    };

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

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

    // ITPCC
    public:
    //      HRESULT __stdcall NewOrder(          int* iSize,
    UCHAR** txn) {return E_NOTIMPL;}
    //      HRESULT __stdcall Payment(          int* iSize,
    UCHAR** txn) {return E_NOTIMPL;}
    //      HRESULT __stdcall StockLevel(      int* iSize, UCHAR** txn)
    {return E_NOTIMPL;}
    //      HRESULT __stdcall OrderStatus(    int* iSize, UCHAR** txn)
    {return E_NOTIMPL;}
    };

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

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

    // ITPCC
    public:
    //      HRESULT __stdcall NewOrder(          int* iSize,
    UCHAR** txn) {return E_NOTIMPL;}
    //      HRESULT __stdcall Payment(          int* iSize,
    UCHAR** txn) {return E_NOTIMPL;}
    //      HRESULT __stdcall StockLevel(      int* iSize, UCHAR** txn)
    {return E_NOTIMPL;}
    };

```

```

    {return E_NOTIMPL;}
    HRESULT __stdcall OrderStatus(    int* iSize, UCHAR** txn)
    {return E_NOTIMPL;}
    };

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

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

    // ITPCC
    public:
    //      HRESULT __stdcall NewOrder(          int* iSize,
    UCHAR** txn) {return E_NOTIMPL;}
    //      HRESULT __stdcall Payment(          int* iSize,
    UCHAR** txn) {return E_NOTIMPL;}
    //      HRESULT __stdcall StockLevel(      int* iSize, UCHAR** txn)
    {return E_NOTIMPL;}
    //      HRESULT __stdcall OrderStatus(    int* iSize, UCHAR** txn)
    {return E_NOTIMPL;}
    };

```

tpcc_com_allsrc\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_allsrc\tpcc_com_all.cpp

```

/*      FILE:          TPCC_COM_ALL.CPP
*      Microsoft TPC-C Kit Ver.
4.20.000
*      Copyright Microsoft, 1999
*      All Rights Reserved
*
*      Version 4.10.000 audited
by Richard Gimarc, Performance Metrics, 3/17/99
*
*      PURPOSE:      Implementation for TPC-C Tuxedo class.
*      Contact:      Charles Levine (clevine@microsoft.com)
*
*      Change history:
*          4.20.000 - updated rev number to match kit
*/

#define STRICT
#define _WIN32_WINNT 0x0400

```

```

#define _ATL_APARTMENT_THREADED

#include <stdio.h>
#include <atlbase.h>
//You may derive a class from CComModule and use it if you want to override
//something, but do not change the name of _Module
extern CComModule _Module;

#include <atlcom.h>
#include <initguid.h>
#include <transact.h>
#include <atimpl.cpp>
#include <comsvcs.h>

#include <sqltypes.h>
#include <sql.h>
#include <sqlext.h>

#include "tpcc_com_ps.h"
#include "..\common\src\trans.h"
//tpckit transaction header contains definations of structures
specific to TPCC
#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 TPCC txns
#include "..\db_odbc_dll\src\tpcc_odbc.h" // ODBC
//implementation of TPCC txns

#include "resource.h"
#include "tpcc_com_all.h"
#include "tpcc_com_all_i.c"
#include "Methods.h"
#include "..\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "..\common\src\ReadRegistry.cpp"

CComModule _Module;

BEGIN_OBJECT_MAP(ObjectMap)
    OBJECT_ENTRY(CLSID_TPCC, CTPCC)
    OBJECT_ENTRY(CLSID_NewOrder, CNewOrder)
    OBJECT_ENTRY(CLSID_OrderStatus, COrderStatus)
    OBJECT_ENTRY(CLSID_Payment, CPayment)
    OBJECT_ENTRY(CLSID_StockLevel, CStockLevel)
END_OBJECT_MAP()

// configuration settings from registry
TPCCREGISTRYDATA Reg;
char
    szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

static HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB *pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC *pCTPCC_ODBC_new;

////////////////////////////////////////////////////
// DLL Entry Point
extern "C"
BOOL WINAPI DllMain(HINSTANCE hInstance, DWORD dwReason, LPVOID
/*lpReserved*/)
{
    char szDllName[128];

    try
    {

```

```

        if (dwReason == DLL_PROCESS_ATTACH)
        {
            _Module.Init(ObjectMap, hInstance);
            DisableThreadLibraryCalls(hInstance);

            DWORD dwSize =
            MAX_COMPUTERNAME_LENGTH+1;

            GetComputerName(szMyComputerName, &dwSize);
            szMyComputerName[dwSize] = 0;

            if
            ( ReadTPCCRegistrySettings( &Reg ) )
            CCOMPONENT_ERR( ERR_MISSING_REGISTRY_ENTRIES );

                throw new
                if (Reg.eDB_Protocol == DBLIB)
                {
                    strcpy( szDllName,
                    Reg.szPath );
                    strcat( szDllName,
                    "tpcc_dblib.dll");
                    hLibInstanceDb =
                    LoadLibrary( szDllName );
                    if (hLibInstanceDb ==
                    NULL)
                        throw new
                        CCOMPONENT_ERR( ERR_LOADDLL_FAILED, szDllName, GetLastError() );

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

                                                // get function pointer to
                                                wrapper for class constructor
                                                pCTPCC_ODBC_new =
                                                (TYPE_CTPCC_ODBC*)
                                                GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
                                                if (pCTPCC_ODBC_new
                                                == NULL)
                                                    throw new
                                                    CCOMPONENT_ERR( ERR_GETPROCADDR_FAILED, szDllName,
                                                    GetLastError() );
                                                        }
                                                        else
                                                        throw new
                                                        CCOMPONENT_ERR( ERR_UNKNOWN_DB_PROTOCOL );
                                                            }
                                                            else if (dwReason == DLL_PROCESS_DETACH)
                                                            _Module.Term();
                                                                }
                                                                    }

```

```

        catch (CBaseErr *e)
        {
            WriteMessageToEventLog(e>ErrorText());
            delete e;
            return FALSE;
        }
        catch (...)
        {
            WriteMessageToEventLog(TEXT("Unhandled
            exception in object DllMain"));
            return FALSE;
        }
        return TRUE; // OK
    }

    //////////////////////////////////////
    // Used to determine whether the DLL can be unloaded by OLE
    STDAPI DllCanUnloadNow(void)
    {
        return (_Module.GetLockCount()==0) ? S_OK : S_FALSE;
    }

    //////////////////////////////////////
    // Returns a class factory to create an object of the requested type
    STDAPI DllGetClassObject(REFCLSID rclsid, REFIID riid, LPVOID* ppv)
    {
        return _Module.GetClassObject(rclsid, riid, ppv);
    }

    //////////////////////////////////////
    // DllRegisterServer - Adds entries to the system registry
    STDAPI DllRegisterServer(void)
    {
        // registers object, typelib and all interfaces in typelib
        return _Module.RegisterServer(TRUE);
    }

    //////////////////////////////////////
    // DllUnregisterServer - Removes entries from the system registry
    STDAPI DllUnregisterServer(void)
    {
        _Module.UnregisterServer();
        return S_OK;
    }

    static void WriteMessageToEventLog(LPTSTR lpszMsg)
    {
        TCHAR szMsg[256];
        HANDLE hEventSource;
        LPTSTR lpszStrings[2];

        // Use event logging to log the error.
        //
        hEventSource = RegisterEventSource(NULL, TEXT("tpcc_com_all.dll"));

        _stprintf(szMsg, TEXT("Error in COM+ TPGC Component: "));
        lpszStrings[0] = szMsg;
        lpszStrings[1] = lpszMsg;

        if (hEventSource != NULL)
        {
            ReportEvent(hEventSource, // handle of event source
            EVENTLOG_ERROR_TYPE, // event type
            0, // event category
            0, // event ID
            NULL, // current user's SID

```

```

        2, // strings in lpszStrings
        0, // no bytes of raw data
        (LPCTSTR *)lpszStrings, // array of error strings
        NULL); // no raw data
    }
    (VOID) DeregisterEventSource(hEventSource);
}

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

/* FUNCTION: CCOMPONENT_ERR::ErrorText
*/
char* CCOMPONENT_ERR::ErrorText(void)
{
    static SERRORMSG errorMsgs[] =
    {
        { ERR_MISSING_REGISTRY_ENTRIES,
        "Required entries missing from registry."
        },
        { ERR_LOADDLL_FAILED,
        "Load of DLL failed. DLL="
        },
        { ERR_GETPROCADDR_FAILED,
        "Could not map proc in DLL. GetProcAddr error. DLL="
        },
        { ERR_UNKNOWN_DB_PROTOCOL,
        "Unknown database protocol specified in registry."
        },
        { 0, ""
        }
    };

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

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

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

```

```

CTPCC_Common::CTPCC_Common()
{
    m_pTxn = NULL;
    m_bCanBePooled = TRUE;
}

CTPCC_Common::~CTPCC_Common()
{
    if (m_pTxn)
        delete m_pTxn;
}

HRESULT CTPCC_Common::CallSetComplete()
{
    IObjectContext* pObjectContext = NULL;

    // get our object context
    HRESULT hr = CoGetObjectContext( IID_IObjectContext, (void
**) &pObjectContext );
    pObjectContext->SetComplete();
    ReleaseInterface(pObjectContext);
    return hr;
}

//
// called by the ctor activator
//
STDMETHODIMP CTPCC_Common::Construct(IDispatch * pUnk)
{
    // Code to access construction string, if needed later...
    // if (pUnk)
    //     return E_UNEXPECTED;
    // IObjectConstructString * pString = NULL;
    // HRESULT hr = pUnk-
>QueryInterface(IID_IObjectConstructString, (void **) &pString);
    // pString->Release();

    try
    {
        if (Reg.eDB_Protocol == ODBC)
            m_pTxn =
pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
szMyComputerName, Reg.szDbName );
        else if (Reg.eDB_Protocol == DBLIB)
            m_pTxn =
pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
szMyComputerName, Reg.szDbName );
    }
    catch (CBaseErr *e)
    {
        WriteMessageToEventLog(e->ErrorText());
        delete e;
        return E_FAIL;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception in object ::Construct"));
        return E_FAIL;
    }

    return S_OK;
}

HRESULT CTPCC_Common::NewOrder(int* iSize, UCHAR **txn)
{
    PNEW_ORDER_DATA pNewOrder;
    COM_DATA *pData;

    try

```

```

{
    pData = (COM_DATA*) *txn;
    pNewOrder = m_pTxn->BuffAddr_NewOrder();

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

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

    pData->retval = e->ErrorType();
    pData->error = e->ErrorNum();
    delete e;
    return E_FAIL;
}
catch (...)
{
    WriteMessageToEventLog(TEXT("Unhandled
exception.));
    pData->retval = ERR_TYPE_LOGIC;
    pData->error = 0;
    m_bCanBePooled = FALSE;
    return E_FAIL;
}
}

HRESULT CTPCC_Common::Payment(int* iSize, UCHAR** txn)
{
    PPAYMENT_DATA pPayment;
    COM_DATA *pData;

    try
    {
        pData = (COM_DATA*) *txn;
        pPayment = m_pTxn->BuffAddr_Payment();

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

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

        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        delete e;
        return E_FAIL;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception.));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        m_bCanBePooled = FALSE;
        return E_FAIL;
    }
}
}

HRESULT CTPCC_Common::StockLevel(int* iSize, UCHAR** txn)
{
    PSTOCK_LEVEL_DATA pStockLevel;
    COM_DATA *pData;

    try
    {
        pData = (COM_DATA*) *txn;
        pStockLevel = m_pTxn->BuffAddr_StockLevel();

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

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

        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        delete e;
        return E_FAIL;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception.));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        m_bCanBePooled = FALSE;
        return E_FAIL;
    }
}
}

HRESULT CTPCC_Common::OrderStatus(int* iSize, UCHAR** txn)
{
    PORDER_STATUS_DATA pOrderStatus;
    COM_DATA *pData;

    try
    {
        pData = (COM_DATA*) *txn;
        pOrderStatus = m_pTxn->BuffAddr_OrderStatus();

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

```

```

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

HRESULT CTPCC_Common::StockLevel(int* iSize, UCHAR** txn)
{
    PSTOCK_LEVEL_DATA pStockLevel;
    COM_DATA *pData;

    try
    {
        pData = (COM_DATA*) *txn;
        pStockLevel = m_pTxn->BuffAddr_StockLevel();

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

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

        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        delete e;
        return E_FAIL;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception.));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        m_bCanBePooled = FALSE;
        return E_FAIL;
    }
}
}

HRESULT CTPCC_Common::OrderStatus(int* iSize, UCHAR** txn)
{
    PORDER_STATUS_DATA pOrderStatus;
    COM_DATA *pData;

    try
    {
        pData = (COM_DATA*) *txn;
        pOrderStatus = m_pTxn->BuffAddr_OrderStatus();

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

```



```

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

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

        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        delete e;
        return E_FAIL;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception.));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        m_bCanBePooled = FALSE;
        return E_FAIL;
    }
}

```

tpcc_com_all\src\tpcc_com_all_i.c

```

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

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

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

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jan 24 20:00:20 2000 */
/*
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADING( )

#if !defined(_M_IA64) && !defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

#ifdef INITGUID
#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_ */

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

#endif /* __IID_DEFINED__ */

#ifdef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif /* CLSID_DEFINED */

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

#endif /* !_MIDL_USE_GUIDDEF_ */

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

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

MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

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

MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif
#endif

```

```

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

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

#else /* !_MIDL_USE_GUIDDEF_ */

#ifdef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif /* __IID_DEFINED__ */

#ifdef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif /* CLSID_DEFINED */

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

#endif /* !_MIDL_USE_GUIDDEF_ */

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

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

MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

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

MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif
#endif

```

```

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

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

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

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

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jan 24 20:00:20 2000 */
/*
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext,
robust
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADING( )

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

#ifdef __cplusplus
extern "C"{
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else /* !_MIDL_USE_GUIDDEF_ */

#ifdef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif /* __IID_DEFINED__ */

#ifdef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif /* CLSID_DEFINED */

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

#endif /* !_MIDL_USE_GUIDDEF_ */

MIDL_DEFINE_GUID(IID,

```

```
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);
```

```
MIDL_DEFINE_GUID(CLSID, CLSID_TPCC, 0x122A3128, 0x2520, 0x11D3, 0xBA, 0x71, 0x00, 0xC0, 0x4F, 0xBF, 0xE0, 0x8B);
```

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

```
MIDL_DEFINE_GUID(CLSID, CLSID_OrderStatus, 0x266836AD, 0xA50D, 0x11D2, 0xBA, 0x4E, 0x00, 0xC0, 0x4F, 0xBF, 0xE0, 0x8B);
```

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

```
MIDL_DEFINE_GUID(CLSID, CLSID_StockLevel, 0x2668369E, 0xA50D, 0x11D2, 0xBA, 0x4E, 0x00, 0xC0, 0x4F, 0xBF, 0xE0, 0x8B);
```

```
#undef MIDL_DEFINE_GUID
```

```
#ifdef __cplusplus  
}  
#endif
```

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

tpcc_com_ps\src\tpcc_com_ps.idl

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

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

```
{  
    object,  
    uuid(FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B),  
    helpstring("ITPCC Interface"),  
    pointer_default(unique)  
}  
interface ITPCC : IUnknown  
{
```

```
HRESULT STDMETHODCALLTYPE  
(  
    [in, out] int* iSize,  
    [in, out, size_is(, *iSize)] char** txn  
);  
HRESULT STDMETHODCALLTYPE  
(  
    [in, out] int* iSize,  
    [in, out, size_is(, *iSize)] char** txn  
);  
HRESULT STDMETHODCALLTYPE  
(  
    [in] int* iSize,  
    [in, size_is(, *iSize)] char** txn  
);  
HRESULT STDMETHODCALLTYPE  
(  
    [in, out] int* iSize,  
    [in, out, size_is(, *iSize)] char** txn  
);  
HRESULT STDMETHODCALLTYPE  
(  
    [in, out] int* iSize,  
    [in, out, size_is(, *iSize)] char** txn  
);  
HRESULT STDMETHODCALLTYPE  
(  
    [in, out] int* iSize,  
    [in, out, size_is(, *iSize)] char** txn  
);  
}; // interface ITPCC
```

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 5.03.0280 */  
/* at Mon Jan 24 20:00:07 2000 */
```

```
/* Compiler settings for \src\tpcc_com_ps.idl:  
Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext  
error checks: allocation ref bounds_check enum stub_data  
VC __declspec() decoration level:  
__declspec(uuid()), __declspec(selectany), __declspec(novtable)  
DECLSPEC_UUID(), MIDL_INTERFACE()  
*/  
//@@@MIDL_FILE_HEADING( )
```

```
/* verify that the <rpcndr.h> version is high enough to compile this file*/  
#ifndef __REQUIRED_RPCNDR_H_VERSION__  
#define __REQUIRED_RPCNDR_H_VERSION__ 440  
#endif  
  
#include "rpc.h"  
#include "rpcndr.h"  
  
#ifndef __RPCNDR_H_VERSION__  
#error this stub requires an updated version of <rpcndr.h>  
#endif // __RPCNDR_H_VERSION__
```

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

```
#ifndef __tpcc_com_ps_h__  
#define __tpcc_com_ps_h__
```

```
/* Forward Declarations */
```

```
#ifndef __ITPCC_FWD_DEFINED__  
#define __ITPCC_FWD_DEFINED__  
typedef interface ITPCC ITPCC;  
#endif /* __ITPCC_FWD_DEFINED__ */
```

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

```
#ifdef __cplusplus  
extern "C"{  
#endif
```

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

```
#ifndef __ITPCC_INTERFACE_DEFINED__  
#define __ITPCC_INTERFACE_DEFINED__
```

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

```
EXTERN_C const IID IID_ITPCC;
```

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

```
MIDL_INTERFACE("FEEE6AA284B1-11d2-BA47-00C04FBFE08B")  
ITPCC : public IUnknown  
{  
public:  
    virtual HRESULT STDMETHODCALLTYPE NewOrder(  
        /* [out][in] */ int __RPC_FAR *iSize,  
        /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR  
        *txn) = 0;
```

```
    virtual HRESULT STDMETHODCALLTYPE Payment(  
        /* [out][in] */ int __RPC_FAR *iSize,  
        /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR
```

```

*txn) = 0;

virtual HRESULT __stdcall Delivery(
    /* [in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn)
= 0;

virtual HRESULT __stdcall StockLevel(
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR
*txn) = 0;

virtual HRESULT __stdcall OrderStatus(
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR
*txn) = 0;

virtual HRESULT __stdcall CallSetComplete( void) = 0;

};

#else /* C style interface */

typedef struct ITPCCVtbl
{
    BEGIN_INTERFACE

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

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

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

    HRESULT ( __stdcall __RPC_FAR *NewOrder )(
        ITPCC __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR *iSize,
        /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR
*txn);

    HRESULT ( __stdcall __RPC_FAR *Payment )(
        ITPCC __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR *iSize,
        /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR
*txn);

    HRESULT ( __stdcall __RPC_FAR *Delivery )(
        ITPCC __RPC_FAR * This,
        /* [in] */ int __RPC_FAR *iSize,
        /* [size_is][size_is][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn);

    HRESULT ( __stdcall __RPC_FAR *StockLevel )(
        ITPCC __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR *iSize,
        /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR
*txn);

    HRESULT ( __stdcall __RPC_FAR *OrderStatus )(
        ITPCC __RPC_FAR * This,
        /* [out][in] */ int __RPC_FAR *iSize,
        /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR
*txn);

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

    END_INTERFACE
} ITPCCVtbl;

```

```

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

#ifndef COBJMACROS

#define ITPCC_QueryInterface(This,riid,ppvObject) \
(This)->lpVtbl -> QueryInterface(This,riid,ppvObject)

#define ITPCC_AddRef(This) \
(This)->lpVtbl -> AddRef(This)

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

#define ITPCC_NewOrder(This,iSize,txn) \
(This)->lpVtbl -> NewOrder(This,iSize,txn)

#define ITPCC_Payment(This,iSize,txn)\
(This)->lpVtbl -> Payment(This,iSize,txn)

#define ITPCC_Delivery(This,iSize,txn) \
(This)->lpVtbl -> Delivery(This,iSize,txn)

#define ITPCC_StockLevel(This,iSize,txn) \
(This)->lpVtbl -> StockLevel(This,iSize,txn)

#define ITPCC_OrderStatus(This,iSize,txn) \
(This)->lpVtbl -> OrderStatus(This,iSize,txn)

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

#endif /* COBJMACROS */

#endif /* C style interface */

HRESULT __stdcall ITPCC_NewOrder_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn);

void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_Payment_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn);

void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *_pdwStubPhase);

```

```

HRESULT __stdcall ITPCC_Delivery_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn);

void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_StockLevel_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn);

void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_OrderStatus_Proxy(
    ITPCC __RPC_FAR * This,
    /* [out][in] */ int __RPC_FAR *iSize,
    /* [size_is][size_is][out][in] */ unsigned char __RPC_FAR * __RPC_FAR *txn);

void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC __RPC_FAR * This);

void __RPC_STUB ITPCC_CallSetComplete_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *_pdwStubPhase);

#endif /* __ITPCC_INTERFACE_DEFINED__ */

/* Additional Prototypes for ALL interfaces */
/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

LIBRARY "tpcc_com_ps"

```

tpcc_com_ps\src\tpcc_com_ps.def

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_p.c

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

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

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jan 24 20:00:07 2000 */

/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()

//@@MIDL_FILE_HEADERING()

#if !defined(_M_IA64) && !defined(_M_IX86)
#define USE_STUBLESS_PROXY

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

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

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 33
#define PROC_FORMAT_STRING_SIZE 193
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 0

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

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

extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

/* Object interface: IUnknown, ver. 0.0,

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

/* Object interface: ITPCC, ver. 0.0,

GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

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

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

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

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

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

CStdStubBuffer_METHODS
};

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

#pragma data_seg(".rdata")

#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.
#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 */
#ifndef _ALPHA_
/* 8 */ NdrFcShort(0x10), /* x86, MIPS, PPC Stack size/offset = 16 */
#else
NdrFcShort(0x20), /* Alpha
Stack size/offset = 32 */
#endif
/* 10 */ NdrFcShort(0x8), /* 8 */
/* 12 */ NdrFcShort(0x10), /* 16 */
/* 14 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has return, */
0x3, /* 3 */
/* Parameter iSize */

<pre> /* 16 */ NdrFcShort(0x158), /* Flags: in, out, base type, simple ref, */ #ifdef _ALPHA_ /* 18 */ NdrFcShort(0x4), /* x86, MIPS, PPC Stack size/offset = 4 */ #else NdrFcShort(0x8), /* Alpha Stack size/offset = 8 */ #endif /* 20 */ 0x8, /* FC_LONG */ /* 0 */ /* Parameter txn */ /* 22 */ NdrFcShort(0x201b), /* Flags: must size, must free, in, out, srv alloc size=8 */ #ifdef _ALPHA_ /* 24 */ NdrFcShort(0x8), /* x86, MIPS, PPC Stack size/offset = 8 */ #else NdrFcShort(0x10), /* Alpha Stack size/offset = 16 */ #endif /* 26 */ NdrFcShort(0x6), /* Type Offset=6 */ /* Return value */ /* 28 */ NdrFcShort(0x70), /* Flags: out, return, base type, */ #ifdef _ALPHA_ /* 30 */ NdrFcShort(0xc), /* x86, MIPS, PPC Stack size/offset = 12 */ #else NdrFcShort(0x18), /* Alpha Stack size/offset = 24 */ #endif /* 32 */ 0x8, /* FC_LONG */ /* 0 */ /* Procedure Payment */ /* 34 */ 0x33, /* FC_AUTO_HANDLE */ /* Old Flags: 0x6c, object, Oi2 */ /* 36 */ NdrFcLong(0x0), /* 0 */ /* 40 */ NdrFcShort(0x4), /* 4 */ #ifdef _ALPHA_ /* 42 */ NdrFcShort(0x10), /* x86, MIPS, PPC Stack size/offset = 16 */ #else NdrFcShort(0x20), /* Alpha Stack size/offset = 32 */ #endif /* 44 */ NdrFcShort(0x8), /* 8 */ /* 46 */ NdrFcShort(0x10), /* 16 */ /* 48 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has return, */ /* 3 */ /* Parameter iSize */ /* 50 */ NdrFcShort(0x158), /* Flags: in, out, base type, simple ref, */ #ifdef _ALPHA_ /* 52 */ NdrFcShort(0x4), /* x86, MIPS, PPC Stack size/offset = 4 */ #else NdrFcShort(0x8), /* Alpha Stack size/offset = 8 */ #endif /* 54 */ 0x8, /* FC_LONG */ /* 0 */ </pre>	<pre> /* Parameter txn */ /* 56 */ NdrFcShort(0x201b), /* Flags: must size, must free, in, out, srv alloc size=8 */ #ifdef _ALPHA_ /* 58 */ NdrFcShort(0x8), /* x86, MIPS, PPC Stack size/offset = 8 */ #else NdrFcShort(0x10), /* Alpha Stack size/offset = 16 */ #endif /* 60 */ NdrFcShort(0x6), /* Type Offset=6 */ /* Return value */ /* 62 */ NdrFcShort(0x70), /* Flags: out, return, base type, */ #ifdef _ALPHA_ /* 64 */ NdrFcShort(0xc), /* x86, MIPS, PPC Stack size/offset = 12 */ #else NdrFcShort(0x18), /* Alpha Stack size/offset = 24 */ #endif /* 66 */ 0x8, /* FC_LONG */ /* 0 */ /* Procedure Delivery */ /* 68 */ 0x33, /* FC_AUTO_HANDLE */ /* Old Flags: 0x6c, object, Oi2 */ /* 70 */ NdrFcLong(0x0), /* 0 */ /* 74 */ NdrFcShort(0x5), /* 5 */ #ifdef _ALPHA_ /* 76 */ NdrFcShort(0x10), /* x86, MIPS, PPC Stack size/offset = 16 */ #else NdrFcShort(0x20), /* Alpha Stack size/offset = 32 */ #endif /* 78 */ NdrFcShort(0x8), /* 8 */ /* 80 */ NdrFcShort(0x8), /* 8 */ /* 82 */ 0x6, /* Oi2 Flags: clt must size, has return, 0x3, */ /* Parameter iSize */ /* 84 */ NdrFcShort(0x148), /* Flags: in, base type, simple ref, */ #ifdef _ALPHA_ /* 86 */ NdrFcShort(0x4), /* x86, MIPS, PPC Stack size/offset = 4 */ #else NdrFcShort(0x8), /* Alpha Stack size/offset = 8 */ #endif /* 88 */ 0x8, /* FC_LONG */ /* 0 */ /* Parameter txn */ /* 90 */ NdrFcShort(0x200b), /* Flags: must size, must free, in, srv alloc size=8 */ #ifdef _ALPHA_ /* 92 */ NdrFcShort(0x8), /* x86, MIPS, PPC Stack size/offset = 8 */ #else NdrFcShort(0x10), /* Alpha Stack size/offset = 16 */ #endif /* 94 */ NdrFcShort(0x18), /* Type Offset=24 */ </pre>	<pre> /* Return value */ /* 96 */ NdrFcShort(0x70), /* Flags: out, return, base type, */ #ifdef _ALPHA_ /* 98 */ NdrFcShort(0xc), /* x86, MIPS, PPC Stack size/offset = 12 */ #else NdrFcShort(0x18), /* Alpha Stack size/offset = 24 */ #endif /* 100 */ 0x8, /* FC_LONG */ /* 0 */ /* Procedure StockLevel */ /* 102 */ 0x33, /* FC_AUTO_HANDLE */ /* Old Flags: 0x6c, object, Oi2 */ /* 104 */ NdrFcLong(0x0), /* 0 */ /* 108 */ NdrFcShort(0x6), /* 6 */ #ifdef _ALPHA_ /* 110 */ NdrFcShort(0x10), /* x86, MIPS, PPC Stack size/offset = 16 */ #else NdrFcShort(0x20), /* Alpha Stack size/offset = 32 */ #endif /* 112 */ NdrFcShort(0x8), /* 8 */ /* 114 */ NdrFcShort(0x10), /* 16 */ /* 116 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has return, */ /* 3 */ /* Parameter iSize */ /* 118 */ NdrFcShort(0x158), /* Flags: in, out, base type, simple ref, */ #ifdef _ALPHA_ /* 120 */ NdrFcShort(0x4), /* x86, MIPS, PPC Stack size/offset = 4 */ #else NdrFcShort(0x8), /* Alpha Stack size/offset = 8 */ #endif /* 122 */ 0x8, /* FC_LONG */ /* 0 */ /* Parameter txn */ /* 124 */ NdrFcShort(0x201b), /* Flags: must size, must free, in, out, srv alloc size=8 */ #ifdef _ALPHA_ /* 126 */ NdrFcShort(0x8), /* x86, MIPS, PPC Stack size/offset= 8 */ #else NdrFcShort(0x10), /* Alpha Stack size/offset = 16 */ #endif /* 128 */ NdrFcShort(0x6), /* Type Offset=6 */ /* Return value */ /* 130 */ NdrFcShort(0x70), /* Flags: out, return, base type, */ #ifdef _ALPHA_ /* 132 */ NdrFcShort(0xc), /* x86, MIPS, PPC Stack size/offset = 12 */ #else NdrFcShort(0x18), /* Alpha Stack size/offset = 24 */ #endif /* 134 */ 0x8, /* FC_LONG */ /* 0 */ </pre>
---	--	--

```

/* Procedure OrderStatus */
/* 136 */ 0x33, /* FC_AUTO_HANDLE */
/* Old Flags:
object, Oi2 */
/* 138 */ NdrFcLong( 0x0 ), /* 0 */
/* 142 */ NdrFcShort( 0x7 ), /* 7 */
#ifdef _ALPHA
/* 144 */ NdrFcShort( 0x10 ), /* x86, MIPS, PPC Stack size/offset =
16 */
#else
NdrFcShort( 0x20 ), /* Alpha
Stack size/offset = 32 */
#endif
/* 146 */ NdrFcShort( 0x8 ), /* 8 */
/* 148 */ NdrFcShort( 0x10 ), /* 16 */
/* 150 */ 0x7, /* Oi2 Flags: srv must size, clt must
size, has return, */
0x3, /* 3 */

/* Parameter iSize */
/* 152 */ NdrFcShort( 0x158 ), /* Flags: in, out, base type, simple ref,
*/
#ifdef _ALPHA
/* 154 */ NdrFcShort( 0x4 ), /* x86, MIPS, PPC Stack size/offset = 4
*/
#else
NdrFcShort( 0x8 ), /* Alpha
Stack size/offset = 8 */
#endif
/* 156 */ 0x8, /* FC_LONG */
/* 0 */

/* Parameter txn */
/* 158 */ NdrFcShort( 0x201b ), /* Flags: must size, must free, in, out,
srv alloc size=8 */
#ifdef _ALPHA
/* 160 */ NdrFcShort( 0x8 ), /* x86, MIPS, PPC Stack size/offset = 8
*/
#else
NdrFcShort( 0x10 ), /* Alpha
Stack size/offset = 16 */
#endif
/* 162 */ NdrFcShort( 0x6 ), /* Type Offset=6 */

/* Return value */
/* 164 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA
/* 166 */ NdrFcShort( 0xc ), /* x86, MIPS, PPC Stack size/offset =
12 */
#else
NdrFcShort( 0x18 ), /* Alpha
Stack size/offset = 24 */
#endif
/* 168 */ 0x8, /* FC_LONG */
/* 0 */

/* Procedure CallSetComplete */
/* 170 */ 0x33, /* FC_AUTO_HANDLE */
/* Old Flags:
object, Oi2 */
/* 172 */ NdrFcLong( 0x0 ), /* 0 */
/* 176 */ NdrFcShort( 0x8 ), /* 8 */
#ifdef _ALPHA
/* 178 */ NdrFcShort( 0x8 ), /* x86, MIPS, PPC Stack size/offset = 8
*/
#else

```

```

Stack size/offset = 16 */
#endif
/* 180 */ NdrFcShort( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x8 ), /* 8 */
/* 184 */ 0x4, /* Oi2 Flags: has return, */
0x1, /* 1 */

/* Return value */
/* 186 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA
/* 188 */ NdrFcShort( 0x4 ), /* x86, MIPS, PPC Stack size/offset = 4
*/
#else
NdrFcShort( 0x8 ), /* Alpha
Stack size/offset = 8 */
#endif
/* 190 */ 0x8, /* FC_LONG */
/* 0 */
0x0

}
};

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
0,
{
NdrFcShort( 0x0 ), /* 0 */
0x11, 0x8, /* FC_RP [simple_pointer]
*/
/* 2 */ 0x8, /* FC_LONG */
/* 4 */ 0x8, /* FC_LONG */
/* 6 */ 0x5c, /* FC_PAD */
0x11, 0x14, /* FC_RP
[allocated_on_stack] [pointer_deref] */
/* 8 */ NdrFcShort( 0x2 ), /* Offset= 2 (10) */
/* 10 */
0x13, 0x0, /* FC_OP */
/* 12 */ NdrFcShort( 0x2 ), /* Offset= 2 (14) */
/* 14 */
0x1b, /*
FC_CARRAY */
0x0, /* 0 */
/* 16 */ NdrFcShort( 0x1 ), /* 1 */
/* 18 */ 0x28, /* Corr desc: parameter, FC_LONG */
0x54, /*
FC_DEREFERENCE */
#ifdef _ALPHA
/* 20 */ NdrFcShort( 0x4 ), /* x86, MIPS, PPC Stack size/offset = 4
*/
#else
NdrFcShort( 0x8 ), /* Alpha
Stack size/offset = 8 */
#endif
/* 22 */ 0x2, /* FC_CHAR */
/* 24 */ 0x5b, /* FC_END */
0x11, 0x14, /* FC_RP
[allocated_on_stack] [pointer_deref] */
/* 26 */ NdrFcShort( 0x2 ), /* Offset= 2 (28) */
/* 28 */
0x12, 0x0, /* FC_UP */
/* 30 */ NdrFcShort( 0xfffff0 ), /* Offset= -16 (14) */
0x0
}
};

const CInterfaceProxyVtbl * _tpcc_com_ps_ProxyVtblList[] =

```

```

{
(CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
0
};

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

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

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

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{
if(!_tpcc_com_ps_CHECK_IID(0))
{
*pIndex = 0;
return 1;
}
return 0;
}

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

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

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

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

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jan 24 20:00:07 2000
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext,
robust
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
/@@MIDL_FILE_HEADERING( )

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

```

```

/* verify that the <rpcproxy.h> version is high enough to compile this file*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 475
#endif

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

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 35
#define PROC_FORMAT_STRING_SIZE 253
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 0

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

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

extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

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

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEE6AA2,0x84B1,0x1d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

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

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

```

```

0,
0,
0,
0
};

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

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

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

static const MIDL_STUB_DESC Object_StubDesc =
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x50002, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.3.280 */
    0,
    0,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* Reserved3 */
    0, /* Reserved4 */
    0, /* Reserved5 */
};

#pragma data_seg("rdata")

#if !defined(__RPC_WIN64__)
#error Invalid build platform for this stub.
#endif

static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =

```

```

{
    0,
    {
        /* Procedure NewOrder */
        FC_AUTO_HANDLE /* 0x33, /* Old Flags:
        object, Oi2 */
        /* 2 */ NdrFcLong( 0x0 ), /* 0 */
        /* 6 */ NdrFcShort( 0x3 ), /* 3 */
        /* 8 */ NdrFcShort( 0x20 ), /* ia64, axp64 Stack size/offset = 32 */
        /* 10 */ NdrFcShort( 0x8 ), /* 8 */
        /* 12 */ NdrFcShort( 0x10 ), /* 16 */
        /* 14 */ 0x47, /* Oi2 Flags: srv must size, clt must
        size, has return, has ext, */
        /* 16 */ 0xa, /* 10 */
        /* 17 */ 0x7, /* Ext Flags:
        new corr desc, clt corr check, srv corr check, */
        /* 18 */ NdrFcShort( 0x1 ), /* 1 */
        /* 20 */ NdrFcShort( 0x1 ), /* 1 */
        /* 22 */ NdrFcShort( 0x0 ), /* 0 */
        /* 24 */ NdrFcShort( 0x0 ), /* 0 */
        /* Parameter iSize */
        /* 26 */ NdrFcShort( 0x158 ), /* Flags: in, out, base type, simple ref,
        */
        /* 28 */ NdrFcShort( 0x8 ), /* ia64, axp64 Stack size/offset = 8 */
        /* 30 */ 0x8, /* FC_LONG */
        /* 31 */ 0x0, /* 0 */
        /* Parameter txn */
        /* 32 */ NdrFcShort( 0x201b ), /* Flags: must size, must free, in, out,
        srv alloc size=8 */
        /* 34 */ NdrFcShort( 0x10 ), /* ia64, axp64 Stack size/offset = 16 */
        /* 36 */ NdrFcShort( 0x6 ), /* Type Offset=6 */
        /* Return value */
        /* 38 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
        /* 40 */ NdrFcShort( 0x18 ), /* ia64, axp64 Stack size/offset = 24 */
        /* 42 */ 0x8, /* FC_LONG */
        /* 43 */ 0x0, /* 0 */
        /* Procedure Payment */
        /* 44 */ 0x33, /* FC_AUTO_HANDLE */
        /* Old Flags:
        object, Oi2 */
        /* 46 */ NdrFcLong( 0x0 ), /* 0 */
        /* 50 */ NdrFcShort( 0x4 ), /* 4 */
        /* 52 */ NdrFcShort( 0x20 ), /* ia64, axp64 Stack size/offset = 32 */
        /* 54 */ NdrFcShort( 0x8 ), /* 8 */
        /* 56 */ NdrFcShort( 0x10 ), /* 16 */
        /* 58 */ 0x47, /* Oi2 Flags: srv must size, clt must
        size, has return, has ext, */
        /* 60 */ 0xa, /* 10 */
        /* 61 */ 0x7, /* Ext Flags:
        new corr desc, clt corr check, srv corr check, */
        /* 62 */ NdrFcShort( 0x1 ), /* 1 */
        /* 64 */ NdrFcShort( 0x1 ), /* 1 */
        /* 66 */ NdrFcShort( 0x0 ), /* 0 */
        /* 68 */ NdrFcShort( 0x0 ), /* 0 */
        /* Parameter iSize */
        /* 70 */ NdrFcShort( 0x158 ), /* Flags: in, out, base type, simple ref,

```

```

*/
/* 72 */ NdrFcShort( 0x8 ), /* ia64, axp64 Stack size/offset = 8 */
/* 74 */ 0x8, /* FC_LONG */
/* 0 */

/* Parameter txn */

/* 76 */ NdrFcShort( 0x201b ), /* Flags: must size, must free, in, out,
srv alloc size=8 */
/* 78 */ NdrFcShort( 0x10 ), /* ia64, axp64 Stack size/offset = 16 */
/* 80 */ NdrFcShort( 0x6 ), /* Type Offset=6 */

/* Return value */

/* 82 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 84 */ NdrFcShort( 0x18 ), /* ia64, axp64 Stack size/offset = 24 */
/* 86 */ 0x8, /* FC_LONG */
/* 0 */

/* Procedure Delivery */

/* 88 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags:

object, Oi2 */
/* 90 */ NdrFcLong( 0x0 ), /* 0 */
/* 94 */ NdrFcShort( 0x5 ), /* 5 */
/* 96 */ NdrFcShort( 0x20 ), /* ia64, axp64 Stack size/offset = 32 */
/* 98 */ NdrFcShort( 0x8 ), /* 8 */
/* 100 */ NdrFcShort( 0x8 ), /* 8 */
/* 102 */ 0x46, /* Oi2 Flags: clt must size, has return,
has ext, */

/* 104 */ 0xa, /* 3 */
/* 10 */
/* Ext Flags:

new corr desc, srv corr check, */
/* 106 */ NdrFcShort( 0x0 ), /* 0 */
/* 108 */ NdrFcShort( 0x1 ), /* 1 */
/* 110 */ NdrFcShort( 0x0 ), /* 0 */
/* 112 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter iSize */

/* 114 */ NdrFcShort( 0x148 ), /* Flags: in, base type, simple ref, */
/* 116 */ NdrFcShort( 0x8 ), /* ia64, axp64 Stack size/offset = 8 */
/* 118 */ 0x8, /* FC_LONG */
/* 0 */

/* Parameter txn */

/* 120 */ NdrFcShort( 0x200b ), /* Flags: must size, must free, in, srv
alloc size=8 */
/* 122 */ NdrFcShort( 0x10 ), /* ia64, axp64 Stack size/offset = 16 */
/* 124 */ NdrFcShort( 0x1a ), /* Type Offset=26 */

/* Return value */

/* 126 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 128 */ NdrFcShort( 0x18 ), /* ia64, axp64 Stack size/offset = 24 */
/* 130 */ 0x8, /* FC_LONG */
/* 0 */

/* Procedure StockLevel */

/* 132 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags:

object, Oi2 */
/* 134 */ NdrFcLong( 0x0 ), /* 0 */
/* 138 */ NdrFcShort( 0x6 ), /* 6 */
/* 140 */ NdrFcShort( 0x20 ), /* ia64, axp64 Stack size/offset = 32 */
/* 142 */ NdrFcShort( 0x8 ), /* 8 */
/* 144 */ NdrFcShort( 0x10 ), /* 16 */
/* 146 */ 0x47, /* Oi2 Flags: srv must size, clt must

```

```

size, has return, has ext, */
0x3, /* 3 */
/* 148 */ 0xa, /* 10 */
0x7, /* Ext Flags:

new corr desc, clt corr check, srv corr check, */
/* 150 */ NdrFcShort( 0x1 ), /* 1 */
/* 152 */ NdrFcShort( 0x1 ), /* 1 */
/* 154 */ NdrFcShort( 0x0 ), /* 0 */
/* 156 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter iSize */

/* 158 */ NdrFcShort( 0x158 ), /* Flags: in, out, base type, simple ref,
*/
/* 160 */ NdrFcShort( 0x8 ), /* ia64, axp64 Stack size/offset = 8 */
/* 162 */ 0x8, /* FC_LONG */
/* 0 */

/* Parameter txn */

/* 164 */ NdrFcShort( 0x201b ), /* Flags: must size, must free, in, out,
srv alloc size=8 */
/* 166 */ NdrFcShort( 0x10 ), /* ia64, axp64 Stack size/offset = 16 */
/* 168 */ NdrFcShort( 0x6 ), /* Type Offset=6 */

/* Return value */

/* 170 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 172 */ NdrFcShort( 0x18 ), /* ia64, axp64 Stack size/offset = 24 */
/* 174 */ 0x8, /* FC_LONG */
/* 0 */

/* Procedure OrderStatus */

/* 176 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags:

object, Oi2 */
/* 178 */ NdrFcLong( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x7 ), /* 7 */
/* 184 */ NdrFcShort( 0x20 ), /* ia64, axp64 Stack size/offset = 32 */
/* 186 */ NdrFcShort( 0x8 ), /* 8 */
/* 188 */ NdrFcShort( 0x10 ), /* 16 */
/* 190 */ 0x47, /* Oi2 Flags: srv must size, clt must
size, has return, has ext, */

/* 192 */ 0xa, /* 3 */
/* 10 */
/* Ext Flags:

new corr desc, clt corr check, srv corr check, */
/* 194 */ NdrFcShort( 0x1 ), /* 1 */
/* 196 */ NdrFcShort( 0x1 ), /* 1 */
/* 198 */ NdrFcShort( 0x0 ), /* 0 */
/* 200 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter iSize */

/* 202 */ NdrFcShort( 0x158 ), /* Flags: in, out, base type, simple ref,
*/
/* 204 */ NdrFcShort( 0x8 ), /* ia64, axp64 Stack size/offset = 8 */
/* 206 */ 0x8, /* FC_LONG */
/* 0 */

/* Parameter txn */

/* 208 */ NdrFcShort( 0x201b ), /* Flags: must size, must free, in, out,
srv alloc size=8 */
/* 210 */ NdrFcShort( 0x10 ), /* ia64, axp64 Stack size/offset = 16 */
/* 212 */ NdrFcShort( 0x6 ), /* Type Offset=6 */

/* Return value */

/* 214 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 216 */ NdrFcShort( 0x18 ), /* ia64, axp64 Stack size/offset = 24 */

```

```

/* 218 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure CallSetComplete */

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

object, Oi2 */
/* 222 */ NdrFcLong( 0x0 ), /* 0 */
/* 226 */ NdrFcShort( 0x8 ), /* 8 */
/* 228 */ NdrFcShort( 0x10 ), /* ia64, axp64 Stack size/offset = 16 */
/* 230 */ NdrFcShort( 0x0 ), /* 0 */
/* 232 */ NdrFcShort( 0x8 ), /* 8 */
/* 234 */ 0x44, /* Oi2 Flags: has return, has ext, */
0x1, /* 1 */
/* Ext Flags:

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

/* Return value */

/* 246 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 248 */ NdrFcShort( 0x8 ), /* ia64, axp64 Stack size/offset = 8 */
/* 250 */ 0x8, /* FC_LONG */
0x0, /* 0 */

}
};

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

/* 2 */ 0x11, 0x8, /* FC_RP [simple_pointer]

*/
/* 4 */ 0x8, /* FC_LONG */
0x5c, /* FC_PAD */

/* 6 */ 0x11, 0x14, /* FC_RP

[allocated_on_stack] [pointer_deref] */
/* 8 */ NdrFcShort( 0x2 ), /* Offset= 2 (10) */
/* 10 */

0x13, 0x0, /* FC_OP */
/* Offset= 2 (14) */

0x1b, /*

FC_CARRAY */
0x0, /* 0 */
/* 16 */ NdrFcShort( 0x1 ), /* 1 */
/* 18 */ 0x28, /* Corr desc: parameter, FC_LONG */
0x54, /*

FC_DEREFERENCE */
/* 20 */ NdrFcShort( 0x8 ), /* ia64, axp64 Stack size/offset = 8 */
/* 22 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 24 */ 0x2, /* FC_CHAR */
0x5b, /* FC_END */

0x11, 0x14, /* FC_RP

[allocated_on_stack] [pointer_deref] */
/* 28 */ NdrFcShort( 0x2 ), /* Offset= 2 (30) */
/* 30 */

0x12, 0x0, /* FC_UP */
/* Offset= -18 (14) */

/* 32 */ NdrFcShort( 0xfffffee ),
0x0

```



```

    }
};

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

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

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

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

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * plndex )
{
    if(!_tpcc_com_ps_CHECK_IID(0))
    {
        *plndex = 0;
        return 1;
    }

    return 0;
}

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

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

```

tpcc_com_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 5.03.0280 */
/* at Mon Jan 24 20:00:07 2000
*/

```

```

/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADING( )

#ifdef _M_IA64 && !defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else /* !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif /* __IID_DEFINED__

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

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)
const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif /* !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,
0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

```

```

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

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

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

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jan 24 20:00:07 2000
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext,
robust
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADING( )

#ifdef _M_IA64 || defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else /* !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif /* __IID_DEFINED__

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

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)
const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif /* !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0x8B);

```

```
#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

tpcc_com_ps\src\dlldata.c

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

DO NOT ALTER THIS FILE

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

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

*****/

#include <rpcproxy.h>

#ifdef __cplusplus
extern "C" {
#endif

EXTERN_PROXY_FILE( tpcc_com_ps )

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

DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )

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

/* end of generated dlldata file */
```

common\src\txn_base.h

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

```
*/
#pragma once

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifdef DllDecl
#define DllDecl __declspec( dllimport )
#endif

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

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

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

common\src\trans.h

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

// String length constants
#define SERVER_NAME_LEN 20
#define DATABASE_NAME_LEN 20
#define USER_NAME_LEN 20
#define PASSWORD_LEN 20
#define TABLE_NAME_LEN 20
#define I_DATA_LEN 50
#define I_NAME_LEN 24
#define BRAND_LEN 1
#define LAST_NAME_LEN 16
#define W_NAME_LEN 10
#define ADDRESS_LEN 20
#define STATE_LEN 2
#define ZIP_LEN 9
#define S_DIST_LEN 24
#define S_DATA_LEN 50
```

```
#define D_NAME_LEN 10
#define FIRST_NAME_LEN 16
#define MIDDLE_NAME_LEN 2
#define PHONE_LEN 16
#define DATETIME_LEN 30
#define CREDIT_LEN 2
#define C_DATA_LEN 250
#define H_DATA_LEN 24
#define DIST_INFO_LEN 24
#define MAX_OL_NEW_ORDER_ITEMS 15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN 25
#define OL_DIST_INFO_LEN 24

// TIMESTAMP_STRUCT is provided by the ODBC header file sqltypes.h, but is
not available
// when compiling with dlib, 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
    short
    ol_supply_w_id;
    long ol_i_id;
    short ol_quantity;

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

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

```

short    d_id;
long     c_id;
short    o_ol_cnt;

// output params
EXEC_STATUS
exec_status_code;
char     c_last[LAST_NAME_LEN+1];
char     c_credit[CREDIT_LEN+1];
double   c_discount;
double   w_tax;
double   d_tax;
long     o_id;
short    o_commit_flag;
TIMESTAMP_STRUCT  o_entry_d;
short    o_all_local;
double   total_amount;
OL_NEW_ORDER_DATAOL[MAX_OL_NEW_ORDER_ITEMS];
} NEW_ORDER_DATA, *PNEW_ORDER_DATA;

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

    // output params
    EXEC_STATUS
    exec_status_code;
    TIMESTAMP_STRUCT  h_date;
    char
    w_street_1[ADDRESS_LEN+1];
    char
    w_street_2[ADDRESS_LEN+1];
    char
    w_city[ADDRESS_LEN+1];
    char
    w_state[STATE_LEN+1];
    char
    w_zip[ZIP_LEN+1];

    char
    d_street_1[ADDRESS_LEN+1];
    char
    d_street_2[ADDRESS_LEN+1];
    char
    d_city[ADDRESS_LEN+1];
    char
    d_state[STATE_LEN+1];
    char
    d_zip[ZIP_LEN+1];

    char
    c_first[FIRST_NAME_LEN+1];
    char
    c_middle[MIDDLE_NAME_LEN + 1];
    char
    c_street_1[ADDRESS_LEN+1];
    char
    c_street_2[ADDRESS_LEN+1];
    char
    c_city[ADDRESS_LEN+1];
    char
    c_state[STATE_LEN+1];
    char
    c_zip[ZIP_LEN+1];

    char
    c_phone[PHONE_LEN+1];
    TIMESTAMP_STRUCT  c_since;
    char
    c_credit[CREDIT_LEN+1];
    double
    c_credit_lim;
    double
    c_discount;
    double
    c_balance;
    char
    c_data[200+1];
} PAYMENT_DATA, *PPAYMENT_DATA;

```

```

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

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

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

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

    // output params
    EXEC_STATUS
    exec_status_code;
    SYSTEMTIME
    long     queue_time;
    long     o_id[10];
} DELIVERY_DATA, *PDELIVERY_DATA;

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

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

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

```

common/src/readregistry.h

```

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

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

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

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

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );

```

```

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

#pragma once

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

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

//error message structure used in ErrorText routines
typedef struct _SERRORMSG
{
    int
    iError;

```

```

//error id of message
char      szMsg[256];           //message to
sent to browser
} SERRORMSG;

#define ERR_FATAL_LEVEL          1
#define ERR_WARNING_LEVEL       2
#define ERR_INFORMATION_LEVEL   3

#define ERR_TYPE_LOGIC          -1 //logic error
in program; internal error
#define ERR_SUCCESS              0 //success (a
non-error error)
#define ERR_BAD_ITEM_ID         1 //expected
abort record in txnRecord
#define ERR_TYPE_DELIVERY_POST  2 //expected delivery post failed
#define ERR_TYPE_WEBDLL         3 //tpcc web
generated error
#define ERR_TYPE_SQL            4 //sql server
generated error
#define ERR_TYPE_DBLIB          5 //dblib
generated error
#define ERR_TYPE_ODBC           6 //odbc
generated error
#define ERR_TYPE_SOCKET         7 //error on
communication socket client rte only
#define ERR_TYPE_DEADLOCK       8 //dblib and odbc only
deadlock condition
#define ERR_TYPE_COM            9 //error from COM call
#define ERR_TYPE_TUXEDO         10 //tuxedo error
#define ERR_TYPE_OS             11 //operating
system error
#define ERR_TYPE_MEMORY         12 //memory
allocation error
#define ERR_TYPE_TPCC_ODBC      13 //error from tpcc odbc txn
module
#define ERR_TYPE_TPCC_DBLIB     14 //error from tpcc dblib txn
module
#define ERR_TYPE_DELISRV        15 //delivery server error
#define ERR_TYPE_TXNLOG         16 //txn log error
#define ERR_TYPE_BCCONN        17 //Benchcraft
connection class
#define ERR_TYPE_TPCC_CONN      18 //Benchcraft connection
class
#define ERR_TYPE_ENCINA         19 //Encina
error
#define ERR_TYPE_COMPONENT      20 //error from COM
component

```

```

class CBaseErr
{
public:
    char      *m_szApp;
    char      *m_szMsg;
    char      *m_szLoc; // code location where the error
    occurred

    int      m_idMsg;

    CBaseErr(void)
    {
        m_idMsg      = 0;
        m_szMsg      = new
char[m_szMsg_size];
        m_szApp      = new
char[m_szApp_size];
        m_szLoc      = NULL;

        m_szMsg[0] = 0;
        m_szApp[0] = 0;

        GetModuleFileName(GetModuleHandle(NULL),
m_szApp, m_szApp_size);
    }

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

    CBaseErr(int idMsg)
    {
        m_idMsg      = idMsg;
        m_szApp      = new
char[m_szApp_size];
        m_szMsg      = new
char[m_szMsg_size];
        m_szLoc      = NULL;

        GetModuleFileName(GetModuleHandle(NULL),
m_szApp, m_szApp_size);
        LoadString(GetModuleHandle(NULL), idMsg,
m_szMsg, m_szMsg_size);
    }

    CBaseErr(LPCTSTR szMsg)
    {
        m_idMsg      = 0;
        m_szApp      = new
char[m_szApp_size];
        m_szMsg      = new
char[m_szMsg_size];
        m_szLoc      = NULL;

        GetModuleFileName(GetModuleHandle(NULL),
m_szApp, m_szApp_size);
        strcpy(m_szMsg, szMsg);
    }

    void SetError(char *szMsg, LPCTSTR szLocation)
    {
        if (szMsg != NULL)
            strcpy(m_szMsg, szMsg);
        else
            m_szMsg[0] = 0;

        if (szLocation != NULL)

```

```

    {
        delete [] m_szLoc;
        m_szLoc = new
char[strlen(szLocation)+1];
        strcpy(m_szLoc, szLocation);
    }
    else
    {
        delete [] m_szLoc;
        m_szLoc = NULL;
    }
}

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

    if (szStr)
        j = sprintf(szTmp, "%s\n", szStr);
    if (m_szLoc)
        j += sprintf(szTmp+j,
"Location=%s\n", m_szLoc);
    if (m_szMsg)
        j += sprintf(szTmp+j, "%s\n",
m_szMsg);

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

char *GetApp(void) { return m_szApp; }
char *GetMsg(void) { return m_szMsg; }
char *GetLocation(void) { return m_szLoc; }

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

class CSocketErr : public CBaseErr
{
public:
    enum Action
    {
        eNone,
        eSend,
        eSocket,
        eConnect
    };

    CSocketErr(Action eAction, LPCTSTR szLocation);
    CSocketErr(int iError) { m_errId = iError; };
    int      m_errId;
    Action   m_eAction;

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

class CSystemErr : public CBaseErr
{
public:
    enum Action
    {
        eNone,
        eTransactNamedPipe,
        eWaitNamedPipe,

```

```

eSetNamedPipeHandleState,
eCreateFile,
eCreateProcess,
eCallNamedPipe,
eCreateEvent,
eCreateThread,
eVirtualAlloc,
eReadFile,
eWriteFile,
eMapViewOfFile,
eCreateFileMapping,
eInitializeSecurityDescriptor,
eSetSecurityDescriptorDacl,
eCreateNamedPipe,
eConnectNamedPipe,
};

CSystemErr(Action eAction, LPCTSTR szLocatio);

void Draw(HWND hwnd, LPCTSTR szStr = NULL);

int m_errld;
Action m_eAction;

int ErrorType() { return ERR_TYPE_OS;}
int ErrorNum() { return m_errld;}
char *ErrorText() { return "";} // TODO:
need to code error text
};

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

    int ErrorType() { return ERR_TYPE_MEMORY;}
    int ErrorNum() { return 0;}
    char *ErrorText() { return "";} // TODO:
need to code error text
};

```

common\src\readregistry.cpp

```

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

/* FUNCTION: ReadTPCCRegistrySettings
* PURPOSE: This function reads the NT registry for startup
parameters. There parameters are
under the TPCC key.
* RETURNSFALSE = no errors
TRUE = error reading registry
*/

```

```

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg )
{
    HKEY hKey;
    DWORD size;
    DWORD type;
    DWORD dwTmp;
    char szTmp[256];

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

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

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

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

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

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

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

    RegCloseKey(hKey);

    return FALSE;
}

```

common\txnlog\include\txnlog.h

```

/* FILE: TXNLOG.H
Microsoft TPC-C Kit Ver.
4.10.000
NOTE: this file is RTE
specific and should not be included
in Full Disclosure Reports.
Copyright Microsoft, 1999
PURPOSE: Structure definitions for logging delivery txn completion
stats.
Contact: Charles Levine (clevine@microsoft.com)
*/

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

typedef struct _TXN_PAYMENT
{
    BYTE CustByName;
    BYTE IsRemote;
} TXN_PAYMENT;

typedef struct _TXN_ORDERSTATUS
{
    BYTE CustByName;
} TXN_ORDERSTATUS;

typedef union _TXN_DETAILS

```



```

//indicates which buffer is active: 0 or 1
int iIoBuffer;
//buffer for any pending IO operation
int iFilePointer;
//position in file.
int iNextRec;
//when reading, ordinal value of next record

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

JULIAN_TIME lastTS;
//when writing sorted output, used to verify
records are sorted
BOOL bWrite;
//writing log file

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

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

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

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

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

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

public:
CTxnLog::CTxnLog(LPCTSTR szFileName, DWORD
dwOpts);
~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 WriteCtrlRecToLog(BYTE SubType, LPCTSTR lpStr,
DWORD dwLen);

void CloseTransactionLogFile(void);

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

int Sort(void);
PTXN_RECORD_HEADER GetSortedRecord(int index);

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 CTPCC_DBLIB_ERRS
{
ERR_BAD_FILE_FORMAT = 1,
// "File format is invalid."
ERR_UNKNOWN_LOG_VERSION,
// "Log file version is unknown."
ERR_BROKEN_LOG_FILE,
// "Log file is broken."
ERR_LOG_NOT_SORTED,
// "Log file is not sorted"
ERR_INVALID_TIME_SEQ,
// "Internal Error: Record Time Sequence invalid."
};

CTXNLOG_ERR( int iErr ) { m_erno = iErr; };

int m_erno;

int ErrorType() {return ERR_TYPE_TXNLOG;};
int ErrorNum() {return m_erno;};

// TODO: need to complete...
char *ErrorText() {return "";};

};

```

common\txnlog\include\spinlock.h

```

/* FILE: SPINLOCK.H
*
* Copyright 1997 Microsoft Corp., All rights reserved.
*
* Authors: Mike Parkes, Charles Levine, Philip Durr
* Microsoft Corp.
*/

#ifdef _INC_Spinlock

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

/*****

```

```

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

class Spinlock
{
// Private data.
HANDLE m_Spinlock; Semaphore;
volatile LONG m_Count; volatile LONG
volatile LONG

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

public:
// Public functions.
Spinlock( void );

inline BOOL ClaimLock( BOOL Wait =
TRUE );

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

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

void WaitForLock( void );
void WakeAllSleepers( void );

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

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

/*****

```

```

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

inline BOOL Spinlock::ClaimLock( BOOL Wait )
{
    if ( ! ClaimSpinlock( (volatile LONG*) &
m_Spinlock ) )
    {
        if ( Wait )
            WaitForLock();
        return Wait;
    }
    return TRUE;
}

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

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

#define _INC_Spinlock

#endif

```

common\txnlog\include\rtetime.h

```

/* FILE: rtetime.h : header file
* Copyright 1997 Microsoft Corp., All rights reserved.
*
* Authors: Charles Levine, Philip Durr
*
* Microsoft Corp.
*/

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

```


Appendix B : Database Design

Build Scripts

BACKUP.SQL

```
-- File: BACKUP.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates backup of tpcc database
```

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

```
dump database tpcc to tpccback1, tpccback2, tpccback3, tpccback4 with int,
stats = 1
```

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

```
go
```

BACKUPDEV.SQL

```
-- File: BACKUPDEVB.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates tpcc database Backup Devices
```

```
use master
go
```

```
-- create backup devices
```

```
exec sp_addumpdevice 'disk','tpccback1','V:\tpccback1.dmp'
go
exec sp_addumpdevice 'disk','tpccback2','V:\tpccback2.dmp'
go
exec sp_addumpdevice 'disk','tpccback3','W:\tpccback3.dmp'
go
exec sp_addumpdevice 'disk','tpccback4','W:\tpccback4.dmp'
go
```

CREATEDB.SQL

```
-- File: CREATEDB.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates tpcc database and backup files
```

```
use master
go
```

```
-- Create temporary table for timing
```

```
if exists ( select name from sysobjects where name = 'tpcc_timer' )
drop table tpcc_timer
```

```
go

create table tpcc_timer
(
    start_date char(30),
    end_date char(30)
)

insert into tpcc_timer values (0,0)
go

-- Store starting time

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

-- create main database files

CREATE DATABASE tpcc
ON PRIMARY
(
    NAME = MSSQL70_tpcc_root,
    FILENAME = "C:\MSSQL70_tpcc_root.mdf",
    SIZE = 8MB,
    FILEGROWTH = 0),
FILEGROUP MSSQL70_misc_fg
(
    NAME = MSSQL70_misc1,
    FILENAME = "F:",
    SIZE = 16500MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL70_misc2,
    FILENAME = "G:",
    SIZE = 16500MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL70_misc3,
    FILENAME = "H:",
    SIZE = 16500MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL70_misc4,
    FILENAME = "I:",
    SIZE = 16500MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL70_misc5,
    FILENAME = "J:",
    SIZE = 16500MB,
    FILEGROWTH = 0),
FILEGROUP MSSQL70_cs_fg
(
    NAME = MSSQL70_cs1,
    FILENAME = "K:",
    SIZE = 30000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL70_cs2,
    FILENAME = "L:",
    SIZE = 30000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL70_cs3,
    FILENAME = "M:",
    SIZE = 30000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL70_cs4,
    FILENAME = "N:",
    SIZE = 30000MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL70_cs5,
    FILENAME = "O:",
    SIZE = 30000MB,
    FILEGROWTH = 0)
LOG ON
(
    NAME = MSSQL70_tpcc_log,
    FILENAME = "E:",
    SIZE = 30000MB,
    FILEGROWTH = 0)
```

```
go

-- Store ending time
update tpcc_timer
set end_date = (select convert(char(30), getdate(),9))
go

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

-- remove temporary table

if exists ( select name from sysobjects where name = 'tpcc_timer' )
drop table tpcc_timer

go
```

DBOPT1.SQL

```
-- File: DBOPT1.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- 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
go
```

```
use tpcc
go
```

```
checkpoint
go
```

DBOPT2.SQL

```
-- File: DBOPT2.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Resets database options after data load
```

```
sp_dboption tpcc,'select into/bulkcopy',FALSE
GO
```

```
sp_dboption tpcc,'trunc. log on chkpt.',FALSE
GO
```

```
USE tpcc
GO
```

```
CHECKPOINT
GO
```

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

```
RECONFIGURE WITH OVERRIDE
GO
```

```
DECLARE @msg varchar(50)
```

```
IF (SELECT (SUBSTRING((SELECT @@version),1,25))) = 'Microsoft SQL
Server 2000'
```

```
BEGIN
```

```
-- --
-- OPTIONS FOR SQL SERVER 8.0 --
```

```

-- Set option values for user-defined indexes --
--
SET @msg = ''
PRINT @msg --
SET @msg = 'Setting SQL Server 8.0'

indexoptions'

PRINT @msg
SET @msg = ''
PRINT @msg --

EXEC sp_indexoption 'customer',
'DisAllowPageLocks', TRUE
EXEC sp_indexoption 'district',
'DisAllowPageLocks', TRUE
EXEC sp_indexoption 'warehouse',
'DisAllowPageLocks', TRUE
EXEC sp_indexoption 'stock',
'DisAllowPageLocks', TRUE
EXEC sp_indexoption 'order_line',
'DisAllowRowLocks', TRUE
EXEC sp_indexoption 'orders',
'DisAllowRowLocks', TRUE
EXEC sp_indexoption 'new_order',
'DisAllowRowLocks', TRUE
EXEC sp_indexoption 'item',
'DisAllowRowLocks', TRUE
EXEC sp_indexoption 'item',
'DisAllowPageLocks', TRUE
END

ELSE
BEGIN
--
-- OPTIONS FOR SQL SERVER 7.0 --
-- Set option values for user-defined indexes --
--
SET @msg = ''
PRINT @msg --
SET @msg = 'Setting SQL Server 7.0'

indexoptions'

PRINT @msg
SET @msg = ''
PRINT @msg --

EXEC sp_indexoption 'customer',
'AllowPageLocks', FALSE
EXEC sp_indexoption 'district',
'AllowPageLocks', FALSE
EXEC sp_indexoption 'warehouse',
'AllowPageLocks', FALSE
EXEC sp_indexoption 'stock',
'AllowPageLocks', FALSE
EXEC sp_indexoption 'order_line',
'AllowPageLocks', FALSE
EXEC sp_indexoption 'orders',
'AllowPageLocks', FALSE
EXEC sp_indexoption 'new_order',
'AllowPageLocks', FALSE
EXEC sp_indexoption 'item',
'AllowPageLocks', FALSE
EXEC sp_indexoption 'item',
'AllowPageLocks', FALSE
END

GO

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

```

```

Print ' Lockflag = 2 ==> Lock at Row-level then Table-level'
Print ' Lockflag = 3 ==> Lock at Table-level'
Print ''

SELECT name,lockflags
FROM sysindexes
WHERE object_id('warehouse') = id OR
object_id('district') = id OR
object_id('customer') = id OR
object_id('stock') = id OR
object_id('orders') = id OR
object_id('order_line') = id OR
object_id('history') = id OR
object_id('new_order') = id OR
object_id('item') = id

ORDER BY lockflags asc
GO

sp_configure 'allow updates',0
GO

RECONFIGURE WITH OVERRIDE
GO

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

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

```

IDXCUSCL.SQL

```

-- File: IDXCUSCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates clustered index on customer table

```

```

use tpcc
go

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

if exists ( select name from sysindexes where name = 'customer_c1' )
drop index customer.customer_c1

create unique clustered index customer_c1 on customer(c_w_id, c_d_id, c_id)
on MSSQL70_cs_fg

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

go

```

IDXCUSNC.SQL

```

-- File: IDXCUSNC.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates non-clustered index on customer table

```

```

use tpcc
go

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

if exists ( select name from sysindexes where name = 'customer_nc1' )
drop index customer.customer_nc1

create unique nonclustered index customer_nc1 on customer(c_w_id, c_d_id,
c_last, c_first, c_id)
on MSSQL70_cs_fg

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

go

```

IDXDISCL.SQL

```

-- File: IDXDISCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates clustered index on district table

```

```

use tpcc
go

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

if exists ( select name from sysindexes where name = 'district_c1' )
drop index district.district_c1

create unique clustered index district_c1 on district(d_w_id, d_id)
with fillfactor=100 on MSSQL70_misc_fg

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

go

```

IDXITMCL.SQL

```

-- File: IDXITMCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates clustered index on item table

```

```

use tpcc
go

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

if exists ( select name from sysindexes where name = 'item_c1' )
drop index item.item_c1

```

```
create unique clustered index item_c1 on item(i_id)
on MSSQL70_misc_fg
```

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

```
go
```

IDXNODCL.SQL

```
-- File: IDXNODCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates clustered index on new_order table
```

```
use tpcc
go
```

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

```
if exists ( select name from sysindexes where name = 'new_order_c1' )
drop index new_order.new_order_c1
```

```
create unique clustered index new_order_c1 on new_order(no_w_id, no_d_id,
no_o_id)
on MSSQL70_misc_fg
```

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

```
go
```

IDXODLCL.SQL

```
-- File: IDXODLCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates clustered index on order_line table
```

```
use tpcc
go
```

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

```
if exists ( select name from sysindexes where name = 'order_line_c1' )
drop index order_line.order_line_c1
```

```
create unique clustered index order_line_c1 on order_line(ol_w_id, ol_d_id,
ol_o_id, ol_number)
on MSSQL70_misc_fg
```

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

```
go
```

IDXORDCL.SQL

```
-- File: IDXORDCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates clustered index on orders table
```

```
use tpcc
go
```

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

```
if exists ( select name from sysindexes where name = 'orders_c1' )
drop index orders.orders_c1
```

```
create unique clustered index orders_c1 on orders(o_w_id, o_d_id, o_id)
on MSSQL70_misc_fg
```

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

```
go
```

IDXORDNC.SQL

```
-- File: IDXORDNC.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates non-clustered index on orders table
```

```
use tpcc
go
```

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

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

```
create index orders_nc1 on orders(o_w_id, o_d_id, o_c_id, o_id)
on MSSQL70_misc_fg
```

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

```
go
```

IDXSTKCL.SQL

```
-- File: IDXSTKCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates clustered index on stock table
```

```
use tpcc
go
```

```
declare @startdate datetime
```

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

```
if exists ( select name from sysindexes where name = 'stock_c1' )
drop index stock.stock_c1
```

```
create unique clustered index stock_c1 on stock(s_i_id, s_w_id)
on MSSQL70_cs_fg
```

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

```
go
```

IDXWARCL.SQL

```
-- File: IDXWARCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates clustered index on warehouse table
```

```
use tpcc
go
```

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

```
if exists ( select name from sysindexes where name = 'warehouse_c1' )
drop index warehouse.warehouse_c1
```

```
create unique clustered index warehouse_c1 on warehouse(w_id)
with fillfactor=100 on MSSQL70_misc_fg
```

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

```
go
```

TABLES.SQL

```
-- File: TABLES.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates TPC-C tables
```

```
use tpcc
go
```

```
--
-- Remove all existing TPC-C tables
--
```

```
if exists ( select name from sysobjects where name = 'warehouse' )
drop table warehouse
```

```
go
if exists ( select name from sysobjects where name = 'district' )
drop table district
```

```
go
if exists ( select name from sysobjects where name = 'customer' )
drop table customer
```

```
go
if exists ( select name from sysobjects where name = 'history' )
```

```

drop table history
go
if exists ( select name from sysobjects where name = 'new_order' )
drop table new_order
go
if exists ( select name from sysobjects where name = 'orders' )
drop table orders
go
if exists ( select name from sysobjects where name = 'order_line' )
drop table order_line
go
if exists ( select name from sysobjects where name = 'item' )
drop table item
go
if exists ( select name from sysobjects where name = 'stock' )
drop table stock
go
--
-- Create new tables
--

create table warehouse
(
    w_id                smallint,
    w_name              char(10),
    w_street_1         char(20),
    w_street_2         char(20),
    w_city              char(20),
    w_state             char(2),
    w_zip              char(9),
    w_tax              numeric(4,4),
    w_ytd              numeric(12,2)
) on MSSQL70_misc_fg
go

create table district
(
    d_id                tinyint,
    d_w_id             smallint,
    d_name              char(10),
    d_street_1         char(20),
    d_street_2         char(20),
    d_city              char(20),
    d_state            char(2),
    d_zip              char(9),
    d_tax              numeric(4,4),
    d_ytd              numeric(12,2),
    d_next_o_id        int
) on MSSQL70_misc_fg
go

create table customer
(
    c_id                int,
    c_d_id             tinyint,
    c_w_id             smallint,
    c_first            char(16),
    c_middle           char(2),
    c_last             char(16),
    c_street_1         char(20),
    c_street_2         char(20),
    c_city             char(20),
    c_state            char(2),
    c_zip              char(9),
    c_phone            char(16),
    c_since            datetime,
    c_credit           char(2),
    c_credit_lim       numeric(12,2),
    c_discount         numeric(4,4)

```

```

    c_balance          numeric(12,2),
    c_ytd_payment     numeric(12,2),
    c_payment_cnt     smallint,
    c_delivery_cnt    smallint,
    c_data            char(500)
) on MSSQL70_cs_fg
go

create table history
(
    h_c_id            int,
    h_c_d_id         tinyint,
    h_c_w_id         smallint,
    h_d_id           tinyint,
    h_w_id           smallint,
    h_date           datetime,
    h_amount         numeric(6,2),
    h_data           char(24)
) on MSSQL70_misc_fg
go

create table new_order
(
    no_o_id          int,
    no_d_id          tinyint,
    no_w_id          smallint
) on MSSQL70_misc_fg
go

create table orders
(
    o_id             int,
    o_d_id           tinyint,
    o_w_id           smallint,
    o_c_id           int,
    o_entry_d        datetime,
    o_carrier_id    tinyint,
    o_ol_cnt         tinyint,
    o_all_local     tinyint
) on MSSQL70_misc_fg
go

create table order_line
(
    ol_o_id          int,
    ol_d_id          tinyint,
    ol_w_id          smallint,
    ol_number        tinyint,
    ol_i_id          int,
    ol_supply_w_id  smallint,
    ol_delivery_d    datetime,
    ol_quantity      smallint,
    ol_amount        numeric(6,2),
    ol_dist_info    char(24)
) on MSSQL70_misc_fg
go

create table item
(
    i_id            int,
    i_im_id         int,
    i_name          char(24),
    i_price         numeric(5,2),
    i_data          char(50)
) on MSSQL70_misc_fg
go

create table stock
(
    s_i_id          int,
    s_w_id          smallint,
    s_quantity      smallint,

```

```

    s_dist_01       char(24),
    s_dist_02       char(24),
    s_dist_03       char(24),
    s_dist_04       char(24),
    s_dist_05       char(24),
    s_dist_06       char(24),
    s_dist_07       char(24),
    s_dist_08       char(24),
    s_dist_09       char(24),
    s_dist_10       char(24),
    s_ytd           int,
    s_order_cnt     smallint,
    s_remote_cnt    smallint,
    s_data          char(50)
) on MSSQL70_cs_fg
go

```

Stored Procedure

DELIVERY.SQL

```

-- File: DELIVERY.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21.000
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates delivery transaction stored procedure
--
-- Interface Level: 4.10.000

use tpcc
go

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

create proc tpcc_delivery @w_id smallint,
                        @o_carrier_id smallint
as
declare @d_id tinyint,
        @o_id int,
        @c_id int,
        @total numeric(12,2),
        @oid1 int,
        @oid2 int,
        @oid3 int,
        @oid4 int,
        @oid5 int,
        @oid6 int,
        @oid7 int,
        @oid8 int,
        @oid9 int,
        @oid10 int

select @d_id = 0

begin tran d

while (@@trancount < 10)
begin
select @o_id = @d_id + 1,
       @total = 0,
       @c_id = 0
select top 1

```

```

        from      @o_id      = no_o_id
        new_order (serializable uplock)
        where     no_w_id    = @w_id and
                 no_d_id    = @d_id
        order    by no_o_id asc

        if (@@rowcount <> 0)
        begin
-- claim the order for this district

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

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

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

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

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

-- accumulate lineitem amounts for this order into customer

                update    customer
                set       c_balance = c_balance
        + @total,
        = c_delivery_cnt + 1
        = @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

NEWORDER.SQL

-- File: NEWORDER.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21.000
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates new order transaction stored procedure
--
-- Interface Level: 4.10.000

use tpcc
go

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

go

create proc tpcc_neworder

        @w_id
        @d_id
        @c_id
        @o_ol_cnt
        @o_all_local
        @i_id1 int =
        @i_id2 int =
        @i_id3 int =
        @i_id4 int =
        @i_id5 int =
        @i_id6 int =
        @i_id7 int =
        @i_id8 int =
        @i_id9 int =
        @i_id10 int =
        @i_id11 int =
        @i_id12 int =
        @i_id13 int =

        @s_w_id1 smallint = 0, @ol_qty1 smallint = 0,
        @s_w_id2 smallint = 0, @ol_qty2 smallint = 0,
        @s_w_id3 smallint = 0, @ol_qty3 smallint = 0,
        @s_w_id4 smallint = 0, @ol_qty4 smallint = 0,
        @s_w_id5 smallint = 0, @ol_qty5 smallint = 0,
        @s_w_id6 smallint = 0, @ol_qty6 smallint = 0,
        @s_w_id7 smallint = 0, @ol_qty7 smallint = 0,
        @s_w_id8 smallint = 0, @ol_qty8 smallint = 0,
        @s_w_id9 smallint = 0, @ol_qty9 smallint = 0,
        @s_w_id10 smallint = 0, @ol_qty10 smallint = 0,
        @s_w_id11 smallint = 0, @ol_qty11 smallint = 0,
        @s_w_id12 smallint = 0, @ol_qty12 smallint = 0,
        @s_w_id13 smallint = 0, @ol_qty13 smallint = 0,

```

```

        @i_id14 int =
        @i_id15 int =

        0, @s_w_id14 smallint = 0, @ol_qty14 smallint = 0,
        0, @s_w_id15 smallint = 0, @ol_qty15 smallint = 0

as
declare @w_tax numeric(4,4),
        @d_tax numeric(4,4),
        @c_last char(16),
        @c_credit char(2),
        @c_discount numeric(4,4),
        @l_price numeric(5,2),
        @l_name char(24),
        @l_data char(50),
        @o_entry_d datetime,
        @remote_flag int,
        @s_quantity smallint,
        @s_data char(50),
        @s_dist char(24),
        @li_no int,
        @o_id int,
        @commit_flag tinyint,
        @li_id int,
        @li_s_w_id smallint,
        @li_qty smallint,
        @ol_number int,
        @c_id_local int

begin

begin transaction n

-- get district tax and next available order id and update
-- plus initialize local variables

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

-- process orderlines

        while (@li_no < @o_ol_cnt)
        begin

                select @li_no = @li_no + 1

-- set i_id, s_w_id, and qty for this lineitem

                select @i_id = case @li_no
                when 1 then @i_id1
                when 2 then @i_id2
                when 3 then @i_id3
                when 4 then @i_id4
                when 5 then @i_id5
                when 6 then @i_id6
                when 7 then @i_id7
                when 8 then @i_id8
                when 9 then @i_id9
                when 10 then @i_id10
                when 11 then @i_id11
                when 12 then @i_id12
                when 13 then @i_id13
                when 14 then @i_id14
                when 15 then @i_id15
                end,

```

```

@li_s_w_id = case @li_no
  when 1 then @s_w_id1
  when 2 then @s_w_id2
  when 3 then @s_w_id3
  when 4 then @s_w_id4
  when 5 then @s_w_id5
  when 6 then @s_w_id6
  when 7 then @s_w_id7
  when 8 then @s_w_id8
  when 9 then @s_w_id9
  when 10 then
@s_w_id10
  when 11 then
@s_w_id11
  when 12 then
@s_w_id12
  when 13 then
@s_w_id13
  when 14 then
@s_w_id14
  when 15 then
@s_w_id15
end,

@li_qty = case @li_no
  when 1 then @ol_qty1
  when 2 then @ol_qty2
  when 3 then @ol_qty3
  when 4 then @ol_qty4
  when 5 then @ol_qty5
  when 6 then @ol_qty6
  when 7 then @ol_qty7
  when 8 then @ol_qty8
  when 9 then @ol_qty9
  when 10 then @ol_qty10
  when 11 then @ol_qty11
  when 12 then @ol_qty12
  when 13 then @ol_qty13
  when 14 then @ol_qty14
  when 15 then @ol_qty15
end

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

-- update stock values
update stock
set s_ytd = s_ytd +
@s_quantity = s_quantity = s_quantity
- @li_qty +

case when (s_quantity - @li_qty < 10) then 91 else 0 end,
s_order_cnt = s_order_cnt + 1,
s_remote_cnt =
s_remote_cnt + case when (@li_s_w_id = @w_id) then 0 else 1 end,
@s_data = s_data,
@s_dist = case
@d_id
when 1
then s_dist_01
when 2
then s_dist_02
when 3
then s_dist_03

```

```

then s_dist_04
when 4
then s_dist_05
when 5
then s_dist_06
when 6
then s_dist_07
when 7
then s_dist_08
when 8
then s_dist_09
when 9
then s_dist_10
when 10
end
where s_i_id = @li_id and
s_w_id =
@li_s_w_id
-- if there actually is a stock (and item) with these ids, goto work
if (@@rowcount > 0)
begin
-- insert order_line data (using data from item and stock)
insert into order_line values(@o_id,
@d_id,
@w_id,
@li_no,
@li_id,
@li_s_w_id,
"dec 31, 1899",
@li_qty,
@i_price * @li_qty,
@s_dist)
-- send line-item data to client
select @i_name,
@s_quantity,
b_g = case when
(patindex("%ORIGINAL%",@i_data) > 0) and
(patindex("%ORIGINAL%",@s_data) > 0) )
then "B" else "G" end,
@i_price,
@i_price * @li_qty
end
begin
-- no item (or stock) found - triggers rollback condition
select "",0,"",0,0
select @commit_flag = 0
end
-- get customer last name, discount, and credit rating

```

```

select @c_last = c_last,
@c_discount = c_discount,
@c_credit = c_credit,
@c_id_local = c_id
from customer (repeatableread)
where c_id = @c_id and
c_w_id = @w_id and
c_d_id = @d_id

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

-- insert corresponding row into new-order table
insert into new_order values ( @o_id,
@d_id,
@w_id)

-- select warehouse tax
select @w_tax = w_tax
from warehouse (repeatableread)
where w_id = @w_id

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

-- all that work for nuthin!!!

-- return order data to client
select @w_tax,
@d_tax,
@o_id,
@c_last,
@c_discount,
@c_credit,
@o_entry_d,
@commit_flag

end
go

ORDSTAT.SQL

-- File: ORDSTAT.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21.000
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates order status transaction stored procedure
--
-- Interface Level: 4.10.000

use tpcc
go

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

```

```

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

as

declare @c_balance               numeric(12,2),
        @c_first                char(16),
        @c_middle               char(2),
        @o_id                  int,
        @o_entry_d              datetime,
        @o_carrier_id           smallint,
        @cnt                    smallint

begin tran o

if (@c_id = 0)
    begin

-- get customer id and info using last name

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

        set         rowcount @cnt

        select      @c_id      = c_id,
                   @c_balance = c_balance,
                   @c_first   = c_first,
                   @c_last    = c_last,
                   @c_middle  = c_middle
        from        customer (repeatableread)
        where       c_last    = @c_last

and

        c_w_id      = @w_id and
        c_d_id      = @d_id

        order      by c_w_id, c_d_id, c_last, c_first

        set         rowcount 0

    end

    else

        begin

-- get customer info if by id

        select      @c_balance = c_balance,
                   @c_first   = c_first,
                   @c_middle  = c_middle,
                   @c_last    = c_last
        from        customer (repeatableread)
        where       c_id      = @c_id and
                   c_d_id    = @d_id and
                   c_w_id    = @w_id

        select      @cnt      = @@rowcount

    end

-- if no such customer

        if (@cnt = 0)
            begin

```

```

                                raiserror("Customer not found",18,1)
                                goto custnotfound
        end

-- get order info

        select      @o_id      = o_id,
                   @o_entry_d = o_entry_d,
                   @o_carrier_id = o_carrier_id

        from        orders (serializable)
        where       o_c_id     = @c_id and
                   o_d_id     = @d_id and
                   o_w_id     = @w_id

        order      by o_id asc

-- select order lines for the current order

        select      ol_supply_w_id,
                   ol_i_id,
                   ol_quantity,
                   ol_amount,
                   ol_delivery_d
        from        order_line (repeatableread)
        where       ol_o_id = @o_id and
                   ol_d_id = @d_id and
                   ol_w_id = @w_id

custnotfound:

commit tran o

-- return data to client

select      @c_id,
           @c_last,
           @c_first,
           @c_middle,
           @o_entry_d,
           @o_carrier_id,
           @c_balance,
           @o_id

go

-- File: PAYMENT.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21.000
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates payment transaction stored procedure
--
-- Interface Level: 4.10.000

use tpcc
go

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

create proc tpcc_payment @w_id    smallint,
                        @c_w_id  smallint,
                        @h_amount numeric(6,2),
                        @d_id    tinyint,
                        @c_d_id  tinyint,
                        @c_id    int,
                        @c_last  char(16) = ""

```

PAYMENT.SQL

```

as
declare @w_street_1 char(20),
        @w_street_2 char(20),
        @w_city     char(20),
        @w_state    char(2),
        @w_zip      char(9),
        @w_name     char(10),
        @d_street_1 char(20),
        @d_street_2 char(20),
        @d_city     char(20),
        @d_state    char(2),
        @d_zip      char(9),
        @d_name     char(10),
        @c_first    char(16),
        @c_middle   char(2),
        @c_street_1 char(20),
        @c_street_2 char(20),
        @c_city     char(20),
        @c_state    char(2),
        @c_zip      char(9),
        @c_phone    char(16),
        @c_since    datetime,
        @c_credit   char(2),
        @c_credit_lim numeric(12,2),
        @c_balance  numeric(12,2),
        @c_discount numeric(4,4),
        @data       char(500),
        @c_data     char(500),
        @datetime   datetime,
        @w_ytd      numeric(12,2),
        @d_ytd      numeric(12,2),
        @cnt        smallint,
        @val        smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local  smallint,
        @c_id_local  int

select @screen_data = ""

begin tran p

-- get payment date

        select      @datetime = getdate()

        if (@c_id = 0)
            begin

-- get customer id and info using last name

                select      @cnt      = count(*)
                from        customer (repeatableread)
                where       c_last    = @c_last and
                           c_w_id    = @c_w_id and
                           c_d_id    = @c_d_id

                select      @val = (@cnt + 1) / 2
                set         rowcount @val

                select      @c_id      = c_id
                from        customer (repeatableread)
                where       c_last    = @c_last and
                           c_w_id    = @c_w_id and
                           c_d_id    = @c_d_id

                order      by c_last, c_first

                set         rowcount 0

            end

-- get customer info and update balances

```



```

update customer
set   @c_balance = c_balance - @h_amount,
      c_payment_cnt = c_payment_cnt + 1,
      c_ytd_payment = c_ytd_payment +
@c_h_amount,
      @c_first = c_first,
      @c_middle = c_middle,
      @c_last = c_last,
      @c_street_1 = c_street_1,
      @c_street_2 = c_street_2,
      @c_city = c_city,
      @c_state = c_state,
      @c_zip = c_zip,
      @c_phone = c_phone,
      @c_credit = c_credit,
      @c_credit_lim = c_credit_lim,
      @c_discount = c_discount,
      @c_since = c_since,
      @data = c_data,
      @c_id_local = c_id

where  c_id = @c_id and
      c_w_id = @c_w_id and
      c_d_id = @c_d_id

-- if customer has bad credit get some more info
if (@c_credit = "BC")
begin
-- compute new info
select @c_data = convert(char(5),@c_id) +
convert(char(4),@c_d_id) +
convert(char(5),@c_w_id) +
convert(char(19),@h_amount) +
substring(@data, 1, 458)

-- update customer info
update customer
set   c_data = @c_data

where  c_id = @c_id and
      c_w_id = @c_w_id and
      c_d_id = @c_d_id

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

-- get district data and update year-to-date
update district
set   d_ytd = d_ytd + @h_amount,
      @d_street_1 = d_street_1,
      @d_street_2 = d_street_2,
      @d_city = d_city,
      @d_state = d_state,
      @d_zip = d_zip,
      @d_name = d_name,
      @d_id_local = d_id

where  d_w_id = @w_id and
      d_id = @d_id

-- get warehouse data and update year-to-date

```

```

update warehouse
set   w_ytd = w_ytd + @h_amount,
      @w_street_1 = w_street_1,
      @w_street_2 = w_street_2,
      @w_city = w_city,
      @w_state = w_state,
      @w_zip = w_zip,
      @w_name = w_name,
      @w_id_local = w_id

where  w_id = @w_id

-- create history record
insert into history values ( @c_id_local,
                             @c_d_id,
                             @c_w_id,
                             @d_id_local,
                             @w_id_local,
                             @datetime,
                             @h_amount,
                             @w_name +
" " + @d_name)
commit tran p

-- return data to client
select @c_id,
       @c_last,
       @datetime,
       @w_street_1,
       @w_street_2,
       @w_city,
       @w_state,
       @w_zip,
       @d_street_1,
       @d_street_2,
       @d_city,
       @d_state,
       @d_zip,
       @c_first,
       @c_middle,
       @c_street_1,
       @c_street_2,
       @c_city,
       @c_state,
       @c_zip,
       @c_phone,
       @c_since,
       @c_credit,
       @c_credit_lim,
       @c_discount,
       @c_balance,
       @screen_data

go

STOCKLEV.SQL

-- File: STOCKLEV.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21.000
-- Copyright Microsoft, 1999, 2000
-- Purpose: Creates stock level transaction stored procedure
--
-- Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_stocklevel" )
drop procedure tpcc_stocklevel

go

```

```

create proc tpcc_stocklevel @w_id smallint,
                           @d_id tinyint,
                           @threshold
as
smallint

declare @o_id_low int,
        @o_id_high int

select @o_id_low = (d_next_o_id - 20),
       @o_id_high = (d_next_o_id - 1)
from district
where  d_w_id = @w_id and
       d_id = @d_id

select count(distinct(s_i_id))
from stock, order_line
where  ol_w_id = @w_id and
       ol_d_id = @d_id and
       between @o_id_low and
              @o_id_high and
       s_w_id = @w_id and
       s_i_id = ol_i_id and
       s_quantity < @threshold

go

```

Loader Source Code

GETARGS.C

```

// File: GETARGS.C
// Microsoft TPC-C Kit Ver.
4.21
// Copyright Microsoft, 1996,
1997, 1998, 1999, 2000
// Purpose: Source file for command line processing

// Includes
#include "tpcc.h"

//=====
//
// Function name: GetArgsLoader
//
//=====

void GetArgsLoader(int argc, char **argv, TPCCLDR_ARGS *pargs)
{
    int i;
    char *ptr;

#ifdef DEBUG
    printf("[%d]DBG: Entering GetArgsLoader(%d)", (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->pack_size = DEFLDPAKSIZE;
pargs->starting_warehouse =
DEF_STARTING_WAREHOUSE;
pargs->build_index =
BUILD_INDEX;
pargs->index_order =
INDEX_ORDER;
pargs->index_script_path = INDEX_SCRIPT_PATH;
pargs->scale_down =
SCALE_DOWN;

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

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

    ptr = argv[i];

    switch ( ptr[1] )
    {
        case 'h': /* Fall through */
        case 'H':
            GetArgsLoaderUsage();
            break;

        case 'D':
            pargs->database = ptr+2;
            break;

        case 'P':
            pargs->password = ptr+2;
            break;

        case 'S':
            pargs->server = ptr+2;
            break;

        case 'U':
            pargs->user = ptr+2;
            break;

        case 'b':
            pargs->batch = atol(ptr+2);
            break;

        case 'W':
            pargs->num_warehouses
= atol(ptr+2);

        case 's':
            pargs-
>starting_warehouse = atol(ptr+2);

        case 't':
            {

```

```

>tables_all = FALSE;
if ( strcmp(ptr+2,"item") == 0 )
    pargs->table_item = TRUE;
else if ( strcmp(ptr+2,"warehouse") == 0 )
    pargs->table_warehouse = TRUE;
else if ( strcmp(ptr+2,"customer") == 0 )
    pargs->table_customer = TRUE;
else if ( strcmp(ptr+2,"orders") == 0 )
    pargs->table_orders = TRUE;
else
{
    printf("\nUnrecognized command");
    GetArgsLoaderUsage();
    exit(1);
}
}

case 'f':
    pargs->loader_res_file =
ptr+2;
    break;

case 'p':
    pargs->pack_size =
atol(ptr+2);
    break;

case 'i':
    pargs->build_index =
atol(ptr+2);
    break;

case 'o':
    pargs->index_order =
atol(ptr+2);
    break;

case 'c':
    pargs->scale_down =
atol(ptr+2);
    break;

case 'd':
    pargs->index_script_path
= ptr+2;
    break;

default:
    GetArgsLoaderUsage();
    exit(-1);
    break;
}

}

/* check for required args */
if ( pargs->num_warehouses == UNDEF )
{
    printf("Number of Warehouses is required\n");
    exit(-2);
}

```

```

}
return;
}

//=====
//
// Function name: GetArgsLoaderUsage
//
//=====
void GetArgsLoaderUsage()
{
#ifdef DEBUG
    printf("[%d]DBG: Entering GetArgsLoaderUsage(%d), (int)
GetCurrentThreadId());
#endif

    printf("TPCCCLR:\n\n");
    printf("Parameter Default\n");
    printf("-----\n");
    printf("-W Number of Warehouses to Load Required\n");
    printf("-S Server %s\n", SERVER);
    printf("-U Username %s\n", USER);
    printf("-P Password %s\n", PASSWORD);
    printf("-D Database %s\n", DATABASE);
    printf("-b Batch Size %ld\n", (long)
BATCH);
    printf("-p TDS packet size %ld\n", (long)
DEFLDPAKSIZE);
    printf("-f Loader Results Output Filename %s\n",
LOADER_RES_FILE);
    printf("-s Starting Warehouse %ld\n", (long)
DEF_STARTING_WAREHOUSE);
    printf("-i Build Option (data = 0, data and index = 1) %ld\n",
(long) BUILD_INDEX);
    printf("-o Cluster Index Build Order (before = 1, after = 0) %ld\n",
(long) INDEX_ORDER);
    printf("-c Build Scaled Database (normal = 0, tiny = 1) %ld\n",
(long) SCALE_DOWN);
    printf("-d Index Script Path %s\n",
INDEX_SCRIPT_PATH);
    printf("-t Table to Load all tables\n");
    printf(" [item|warehouse|customer|orders]\n");
    printf(" Notes: \n");
    printf(" - the '-t' parameter may be included multiple times to\n");
    printf(" specify multiple tables to be loaded\n");
    printf(" - 'item' loads ITEM table\n");
    printf(" - 'warehouse' loads WAREHOUSE, DISTRICT, and STOCK tables\n");
    printf(" - 'customer' loads CUSTOMER and HISTORY tables\n");
    printf(" - 'orders' load NEW-ORDER, ORDERS, ORDER-LINE tables\n");

    printf("\nNote: Command line switches are case sensitive.\n");

    exit(0);
}

RANDOM.C

// File: RANDOM.C
// Microsoft TPC-C Kit Ver.
// 4.21
// Copyright Microsoft, 1996,
// 1997, 1998, 1999, 2000

```

```

// 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());
printf("Old Seed %ld New Seed %ld\n", Seed, val);
#endif

if ( val < 0 )
val = abs(val);

Seed = val;
}

/*****
 *
 * irand - returns a 32 bit integer pseudo random number with a period of
 * 1 to 2 ^ 32 - 1.
 * parameters:
 * none.
 * returns:
 * 32 bit integer - defined as long ( see above ).
 * side effects:
 * seed get recomputed.
 *****/

long irand()
{

```

```

register long s; /* copy of seed */
register long test; /* test flag */
register long hi; /* tmp value for speed */
register long lo; /* tmp value for speed */

#ifdef DEBUG
printf("[%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

if ( upper == lower ) /* pgd 08-13-96 perf enhancement */
return lower;

upper++;

if ( upper <= lower )
rand_num = upper;
else
rand_num = lower + irand() % (upper - lower); /* pgd
08-13-96 perf enhancement */

#ifdef DEBUG
printf("[%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:
//=====
long NURand(int iConst,
long x,
long y,
long C)
{
long 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 = %d\n", (int) GetCurrentThreadId(),
rand_num);
#endif

return rand_num;
}

```

STRINGS.C

```
// File: STRINGS.C
```

```

//
4.21                                Microsoft TPC-C Kit Ver.
//
//                                Copyright Microsoft, 1996,
1997, 1998, 1999, 2000
// Purpose: Source file for database loader string functions

// Includes
#include "tpcc.h"
#include <string.h>
#include <ctype.h>

//=====
//
// Function name: MakeAddress
//
//=====
void MakeAddress(char *street_1,
                char *street_2,
                char *city,
                char *state,
                char *zip)
{
#ifdef DEBUG
    printf("[%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 <%d> out of
range (0,999)\n", num);
    exit(-1);
}

#ifdef DEBUG
    printf("[%d]DBG: LastName: num = [%d] ==> [%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]
//respectively, n-string [x .. y]) represents a string of random alphanumeric
//respectively, numeric) characters of a random length of minimum x, maximum
y,
//and mean (y+x)/2. Alphanumerics are A..Z, a..z, and 0..9. The only other
//requirement is that the character set used "must be able to represent a
minimum
//of 128 different characters". We are using 8-bit chars, so this is a non issue.
//It is completely unreasonable to stuff non-printing chars into the text fields.
//-CLevine 08/13/96

int MakeAlphaString(int x, int y, int z, char *str)
{
    int len;
    int i;
    char cc = 'a';
    static char chArray[] =
    "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
    static int chArrayMax = 61;

#ifdef DEBUG
    printf("[%d]DBG: Entering MakeAlphaString()\n", (int) GetCurrentThreadId());
#endif

    len = RandomNumber(x, y);

    for (i=0; i<len; i++)
    {
        cc = chArray[RandomNumber(0, chArrayMax)];

```

```

        str[i] = cc;
    }

    if (len < z)
        memset(str+len, ' ', z - len);

    str[len] = 0;

    return len;
}

//=====
//
// Function name: MakeOriginalAlphaString
//
//=====
int MakeOriginalAlphaString(int x,
int y,
int z,
char *str,
int percent)
{
    int len;
    int val;
    int start;

#ifdef DEBUG
    printf("[%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, "000011111");

    itoa(RandomNumber(0, 9999), tmp, 10);
    memcpy(str, tmp, strlen(tmp));

    return 9;
}

//=====
//
// Function name: InitString
//
//=====
void InitString(char *str, int len)
{
#ifdef DEBUG
    printf("[%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
// 4.21 Microsoft TPC-C Kit Ver.
// 1997, 1998, 1999, 2000 Copyright Microsoft, 1996,
// 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 e_l_time;

#ifdef DEBUG
    printf("[%d]DBG: Entering TimeNow(\n", (int) GetCurrentThreadId());
#endif

    _ftime(&e_l_time);

    time_now = ((e_l_time.time - start_sec) * 1000) + e_l_time.millitm;

    return time_now;
}

```

TPCC.H

```

// File: TPCC.H
// 4.21 Microsoft TPC-C Kit Ver.
// 1997, 1998, 1999, 2000 Copyright Microsoft, 1996,
// Purpose: Header file for TPCC database loader

```

```

// Build number of TPC Benchmark Kit
#define TPCKIT_VER "4.21"

```

```

// General headers
#include <windows.h>
#include <winbase.h>
#include <stdlib.h>
#include <stdio.h>
#include <process.h>
#include <stddef.h>
#include <stdarg.h>
#include <string.h>
#include <time.h>
#include <sys/timeb.h>
#include <sys/types.h>

```

```

// ODBC headers
#include <sql.h>
#include <sqltext.h>
#include <odbcss.h>

```

```

// General constants
#define MILLI 1000
#define FALSE 0
#define TRUE 1
#define UNDEF -1
#define MINPRINTASCII 32
#define MAXPRINTASCII 126

```

```

// Default environment constants
#define SERVER ""
#define DATABASE "tpcc"
#define USER "sa"
#define PASSWORD ""

```

```

// Default loader arguments
#define BATCH 10000
#define DEFLOADPACKSIZE 32768
#define LOADER_RES_FILE "logs\\load.out"

```

```

#define LOADER_NURAND_C          123
#define DEF_STARTING_WAREHOUSE  1
#define BUILD_INDEX              1 // build both data and indexes
#define INDEX_ORDER              1 // build indexes before load
#define SCALE_DOWN               0 // build a normal scale database
#define INDEX_SCRIPT_PATH        "scripts"

typedef struct
{
    char *server;
    char *database;
    char *user;
    char *password;
    BOOL tables_all;
    BOOL table_item; // set if loading ITEM table specifically
    BOOL table_warehouse; // set if loading WAREHOUSE,
DISTRICT, and STOCK
    BOOL table_customer; // set if loading
CUSTOMER and HISTORY
    BOOL table_orders; // set if loading NEW-ORDER, ORDERS, ORDER
LINE
    long num_warehouses;
    long batch;
    long verbose;
    long pack_size;
    char *loader_res_file;
    char *synch_servername;
    long case_sensitivity;
    long starting_warehouse;
    long build_index;
    long index_order;
    long scale_down;
    char *index_script_path;
} TPCCCLR_ARGS;

// String length constants
#define SERVER_NAME_LEN 20
#define DATABASE_NAME_LEN 20
#define USER_NAME_LEN 20
#define PASSWORD_LEN 20
#define TABLE_NAME_LEN 20
#define I_DATA_LEN 50
#define I_NAME_LEN 24
#define BRAND_LEN 1
#define LAST_NAME_LEN 16
#define W_NAME_LEN 10
#define ADDRESS_LEN 20
#define STATE_LEN 2
#define ZIP_LEN 9
#define S_DIST_LEN 24
#define S_DATA_LEN 50
#define D_NAME_LEN 10
#define FIRST_NAME_LEN 16
#define MIDDLE_NAME_LEN 2
#define PHONE_LEN 16

```

```

#define CREDIT_LEN 2
#define C_DATA_LEN 500
#define H_DATA_LEN 24
#define DIST_INFO_LEN 24
#define MAX_OL_NEW_ORDER_ITEMS 15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN 25
#define OL_DIST_INFO_LEN 24
#define C_SINCE_LEN 23
#define H_DATE_LEN 23
#define OL_DELIVERY_D_LEN 23
#define O_ENTRY_D_LEN 23

```

```

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

```

```

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

```

```

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

```

```

// Functions in strings.c
void MakeAddress();
void LastName();
int MakeAlphaString();
int MakeOriginalAlphaString();
int MakeNumberString();
int MakeZipNumberString();
void InitString();
void InitAddress();
void PadString();

```

TPCCCLR.C

```

// File: TPCCCLR.C
// Microsoft TPC-C Kit Ver. 4.21
// Copyright Microsoft, 1996, 1997, 1998, 1999, 2000
// Purpose: Source file for TPC-C database loader

```

```

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

```

```

// Defines
#define MAXITEMS 10000
#define MAXITEMS_SCALE_DOWN 100
#define CUSTOMERS_PER_DISTRICT 3000
#define CUSTOMERS_SCALE_DOWN 30
#define DISTRICT_PER_WAREHOUSE 10
#define ORDERS_PER_DISTRICT 3000
#define ORDERS_SCALE_DOWN 30
#define MAX_CUSTOMER_THREADS 2
#define MAX_ORDER_THREADS 3
#define MAX_MAIN_THREADS 4

```

```

// Functions declarations
void HandleErrorDBC(SQLHDBC hdbc1);

```

```

void CheckSQL();
void CheckDataBase();

```

```

long NURand();
void LoadItem();
void LoadWarehouse();

```

```

void Stock();
void District();

```

```

void LoadCustomer();
void CustomerBufInit();
void CustomerBufLoad();
void LoadCustomerTable();
void LoadHistoryTable();

```

```

void LoadOrders();
void OrdersBufInit();
void OrdersBufLoad();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void GetPermutation();
void CheckForCommit();
void OpenConnections();
void BuildIndex();
void FormatDate();

```

```

// Shared memory structures

```

```

typedef struct
{
    long ol;
    long ol_i_id;
    short ol_supply_w_id;
    short ol_quantity;
    double ol_amount;
    char ol_dist_info[DIST_INFO_LEN+1];
    char ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCT;

```

```

typedef struct
{
    long o_id;
    short o_d_id;
    short o_w_id;
    long o_c_id;
    short o_carrier_id;
    short o_ol_cnt;
    short o_all_local;
    ORDER_LINE_STRUCT o_ol[15];
} ORDERS_STRUCT;

```

```

typedef struct
{
    long c_id;
    short c_d_id;
    short c_w_id;
    char c_first[FIRST_NAME_LEN+1];
    char c_middle[MIDDLE_NAME_LEN+1];
    char c_last[LAST_NAME_LEN+1];
    char c_street_1[ADDRESS_LEN+1];
    char

```

```

char      c_street_2[ADDRESS_LEN+1];
char      c_city[ADDRESS_LEN+1];
char      c_state[STATE_LEN+1];
char      c_zip[ZIP_LEN+1];
char      c_phone[PHONE_LEN+1];
char      c_credit[CREdit_LEN+1];
double    c_credit_lim;
double    c_discount;
// fix to avoid ODBC float to numeric conversion problem.
// double  c_balance;

char      c_balance[6];

double    c_ytd_payment;

short     c_payment_cnt;
short     c_delivery_cnt;
char      c_data[C_DATA_LEN+1];
double    h_amount;
char      h_data[H_DATA_LEN+1];
} CUSTOMER_STRUCT;

typedef struct
{
    char    c_last[LAST_NAME_LEN+1];
    char    c_first[FIRST_NAME_LEN+1];

    long    c_id;
} CUSTOMER_SORT_STRUCT;

typedef struct
{
    long    time_start;
} LOADER_TIME_STRUCT;

// Global variables
char      szLastError[300];
HENV      henv;
HDBC      v_hdbc;
// for SQL Server version verification
HDBC      i_hdbc1;
// for ITEM table
HDBC      w_hdbc1;
// for WAREHOUSE, DISTRICT, STOCK
HDBC      c_hdbc1;
// for CUSTOMER
HDBC      c_hdbc2;
// for HISTORY
HDBC      o_hdbc1;
// for ORDERS
HDBC      o_hdbc2;
// for NEW-ORDER
HDBC      o_hdbc3;
// for ORDER-LINE
HSTMT     v_hstmt;
// for SQL Server version verification
HSTMT     i_hstmt1;
HSTMT     w_hstmt1;
HSTMT     c_hstmt1, c_hstmt2;
HSTMT     o_hstmt1, o_hstmt2, o_hstmt3;

ORDERS_STRUCT orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT customer_buf[CUSTOMERS_PER_DISTRICT];
long        orders_rows_loaded;

```

```

long      new_order_rows_loaded;
long      order_line_rows_loaded;
long      history_rows_loaded;
long      customer_rows_loaded;
long      stock_rows_loaded;
long      district_rows_loaded;
long      item_rows_loaded;
long      warehouse_rows_loaded;
long      main_time_start;
long      main_time_end;

long      max_items;
long      customers_per_district;
long      orders_per_district;
long      first_new_order;
long      last_new_order;

TPCCCLDR_ARGS *aptr, args;

//=====================================================
//
// Function name: main
//
//=====================================================
int main(int argc, char **argv)
{
    DWORD      dwThreadId[MAX_MAIN_THREADS];
    HANDLE      hThread[MAX_MAIN_THREADS];
    FILE        *fLoader;
    char        buffer[255];
    int         i;

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

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

    // process command line arguments
    aptr = &args;
    GetArgsLoader(argc, argv, aptr);

    // verify database and tables exist before attempting to load
    CheckDataBase();
    printf("Build interface is ODBC.\n");

    if (aptr->build_index == 0)
        printf("Data load only - no index creation.\n");
    else
        printf("Data load and index creation.\n");

    if (aptr->index_order == 0)
        printf("Clustered indexes will be created after bulk
load.\n");
    else
        printf("Clustered indexes will be created before bulk
load.\n");

```

```

// set database scale values
if (aptr->scale_down == 1)
{
    printf("**** Scaled Down Database ****\n");
    max_items = MAXITEMS_SCALE_DOWN;
    customers_per_district =
CUSTOMERS_SCALE_DOWN;
    orders_per_district = ORDERS_SCALE_DOWN;
    first_new_order = 0;
    last_new_order = 30;
}
else
{
    max_items = MAXITEMS;
    customers_per_district =
CUSTOMERS_PER_DISTRICT;
    orders_per_district = ORDERS_PER_DISTRICT;
    first_new_order = 2100;
    last_new_order = 3000;
}

// open connections to SQL Server
OpenConnections();

// open file for loader results
fLoader = fopen(aptr->loader_res_file, "w");

if (fLoader == NULL)
{
    printf("Error, loader result file open failed.");
    exit(-1);
}

// start loading data
printf(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,
(LPCTSTR) LPTHREAD_START_ROUTINE) LoadItem,
NULL,
0,
&dwThreadId[0]);

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

```

```

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

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

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

            hThread[2] = CreateThread(NULL,
                0,
                (LPTHREAD_START_ROUTINE) LoadCustomer,
                NULL,
                0,
                &dwThreadID[2]);
            if (hThread[2] == NULL)
            {
                printf("Error, failed in creating creating
main thread = 2.\n");
                exit(-1);
            }
        }

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

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

```

```

    }

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

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

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

    fclose(fLoader);

    SQLFreeEnv(henv);

    exit(0);

    return 0;
}

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

void LoadItem()
{
    long        long        i_id;
                long        i_im_id;
    char        char        i_name[I_NAME_LEN+1];
    double      double      i_price;
    char        char        i_data[I_DATA_LEN+1];

    char        char        name[20];
    long        long        time_start;
    RETCODE     rc;
    DBINT       rcint;
    char        char        bcphint[128];

    // 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("idxitmc");

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

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

    rc = bcp_init(i_hdbc1, name, NULL, "logs\\item.err", DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))

```

```

    {
        sprintf(bcphint, "tablock, order (i_id),
ROWS_PER_BATCH = 100000");
        rc = bcp_control(i_hdbc1, BCPHINTS, (void*)
bcphint);
        if (rc != SUCCEEDED)
            HandleErrorDBC(i_hdbc1);
    }

    rc = bcp_bind(i_hdbc1, (BYTE *) &i_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT4, 1);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);

    rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 2);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);

    rc = bcp_bind(i_hdbc1, (BYTE *) i_name, 0, I_NAME_LEN, NULL,
0, 0, 3);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);

    rc = bcp_bind(i_hdbc1, (BYTE *) &i_price, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 4);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);

    rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0, I_DATA_LEN, NULL, 0,
0, 5);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);

    time_start = (TimeNow() / MILLI);
    item_rows_loaded = 0;

    for (i_id = 1; i_id <= max_items; i_id++)
    {
        i_im_id = RandomNumber(1L, 10000L);
        MakeAlphaString(14, 24, I_NAME_LEN, i_name);
        i_price = ((float) RandomNumber(100L,
10000L))/100.0;
        MakeOriginalAlphaString(26, 50, I_DATA_LEN,
i_data, 10);
        rc = bcp_sendrow(i_hdbc1);
        if (rc != SUCCEEDED)
            HandleErrorDBC(i_hdbc1);

        item_rows_loaded++;
        CheckForCommit(i_hdbc1, i_hstmt1,
item_rows_loaded, "item", &time_start);
    }

    rcint = bcp_done(i_hdbc1);
    if (rcint < 0)
        HandleErrorDBC(i_hdbc1);

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

    SQLFreeStmt(i_hstmt1, SQL_DROP);
    SQLDisconnect(i_hdbc1);
    SQLFreeConnect(i_hdbc1);

    // if build index after load
    if ((aptr->build_index == 1) && (aptr->index_order == 1))

```



```

        BuildIndex("idxtmcl");
    }

//=====
//
// Function : LoadWarehouse
//
// Loads WAREHOUSE table and loads Stock and District as Warehouses are
// created
//
//=====
void LoadWarehouse()
{
    short    w_id;
    char     w_name[W_NAME_LEN+1];
    char     w_street_1[ADDRESS_LEN+1];
    char     w_street_2[ADDRESS_LEN+1];
    char     w_city[ADDRESS_LEN+1];
    char     w_state[STATE_LEN+1];
    char     w_zip[ZIP_LEN+1];
    double   w_tax;
    double   w_ytd;
    char     name[20];
    long     time_start;
    RETCODE  rc;
    DBINT    rcint;
    char     bcpint[128];

    // Seed with unique number
    seed(2);

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

    // if build index before load...
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        BuildIndex("idxwarc1");

    InitString(w_name, W_NAME_LEN+1);
    InitAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

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

    rc = bcp_init(w_hdbc1, name, NULL, "logs\\warehouse.err", DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcpint, "tablock, order (w_id,
ROWS_PER_BATCH = %d", aptr->num_warehouses);
        rc = bcp_control(w_hdbc1, BCPHINTS, (void*)
bcpint);
        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, W_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++)
    {
        MakeAlphaString(6, 10, W_NAME_LEN, w_name);

        MakeAddress(w_street_1, w_street_2, w_city,
w_state, w_zip);

        w_tax = ((float)
RandomNumber(0L, 2000L)) / 10000.00;

        w_ytd = 300000.00;

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

        warehouse_rows_loaded++;
        CheckForCommit(w_hdbc1, i_hstmt1,
warehouse_rows_loaded, "warehouse", &time_start);
    }

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

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

    // if build index after load...
    if ((aptr->build_index == 1) && (aptr->index_order == 0))
        BuildIndex("idxward1");

    stock_rows_loaded = 0;
}

```

```

        district_rows_loaded = 0;

    District();
    Stock();
}

//=====
//
// Function : District
//
//=====
void District()
{
    short    d_id;
    short    d_w_id;
    char     d_name[D_NAME_LEN+1];
    char     d_street_1[ADDRESS_LEN+1];
    char     d_street_2[ADDRESS_LEN+1];
    char     d_city[ADDRESS_LEN+1];
    char     d_state[STATE_LEN+1];
    char     d_zip[ZIP_LEN+1];
    double   d_tax;
    double   d_ytd;
    char     name[20];
    long     d_next_o_id;
    long     time_start;
    int      w_id;
    RETCODE  rc;
    DBINT    rcint;
    char     bcpint[128];

    // Seed with unique number
    seed(4);

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

    // build index before load
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        BuildIndex("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);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcpint, "tablock, order (d_w_id, d_id,
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 10));
        rc = bcp_control(w_hdbc1, BCPHINTS, (void*)
bcpint);
        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 = apr->starting_warehouse; w_id <= apr-
>num_warehouses; w_id++)
{
    d_w_id = w_id;

    for (d_id = 1; d_id <=
DISTRICT_PER_WAREHOUSE; d_id++)
    {
        MakeAlphaString(6,10,D_NAME_LEN,
d_name);

        MakeAddress(d_street_1, d_street_2,
d_city, d_state, d_zip);

        d_tax = ((float)
RandomNumber(0L,2000L))/10000.00;

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

```

```

HandleErrorDBC(w_hdbc1);

        district_rows_loaded++;
        CheckForCommit(w_hdbc1, w_hstmt1,
district_rows_loaded, "district", &time_start);
    }
}

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

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

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

return;
}

//=====
//
// Function : Stock
//
//=====

void Stock()
{
    long    s_i_id;
    short   s_w_id;
    short   s_quantity;
    char    s_dist_01[S_DIST_LEN+1];
    char    s_dist_02[S_DIST_LEN+1];
    char    s_dist_03[S_DIST_LEN+1];
    char    s_dist_04[S_DIST_LEN+1];
    char    s_dist_05[S_DIST_LEN+1];
    char    s_dist_06[S_DIST_LEN+1];
    char    s_dist_07[S_DIST_LEN+1];
    char    s_dist_08[S_DIST_LEN+1];
    char    s_dist_09[S_DIST_LEN+1];
    char    s_dist_10[S_DIST_LEN+1];
    long    s_ytd;
    short   s_order_cnt;
    short   s_remote_cnt;
    char    s_data[S_DATA_LEN+1];
    len;
    char    name[20];
    long    time_start;
    RETCODE rc;
    DBINT   rcint;
    char    bcphint[128];

    // Seed with unique number
    seed(3);

    // if build index before load...
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        BuildIndex("idxstck");

    sprintf(name, "%s.%s", apr->database, "stock");

rc = bcp_init(w_hdbc1, name, NULL, "log\stock.err", DB_IN);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    sprintf(bcphint, "tablock, order (s_i_id, s_w_id),

```

```

ROWS_PER_BATCH = %u", (aptr->num_warehouses * 100000));
rc = bcp_control(w_hdbc1, BCPHINTS, (void*)
bcphint);
if (rc != 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);

rc = 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, 0, 4);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

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

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

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

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

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

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

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

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

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0, S_DIST_LEN,
NULL, 0, 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);

rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0, S_DATA_LEN, NULL,
0, 0, 17);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

s_ytd = s_order_cnt = s_remote_cnt = 0;

time_start = (TimeNow() / MILLISEC);

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

for (s_i_id=1; s_i_id <= max_items; s_i_id++)
{
    for (s_w_id = (short)aptr->starting_warehouse;
s_w_id <= aptr->num_warehouses; s_w_id++)
    {
        s_quantity =
(short)RandomNumber(10L, 100L);
        len =
MakeAlphaString(24, 24, S_DIST_LEN, s_dist_01);
        len =
MakeAlphaString(24, 24, S_DIST_LEN, s_dist_02);
        len =
MakeAlphaString(24, 24, S_DIST_LEN, s_dist_03);
        len =
MakeAlphaString(24, 24, S_DIST_LEN, s_dist_04);
        len =
MakeAlphaString(24, 24, S_DIST_LEN, s_dist_05);
        len =
MakeAlphaString(24, 24, S_DIST_LEN, s_dist_06);
        len =
MakeAlphaString(24, 24, S_DIST_LEN, s_dist_07);
        len =
MakeAlphaString(24, 24, S_DIST_LEN, s_dist_08);
        len =
MakeAlphaString(24, 24, S_DIST_LEN, s_dist_09);
        len =
MakeAlphaString(24, 24, S_DIST_LEN, s_dist_10);

        len = MakeOriginalAlphaString(26, 50,
S_DATA_LEN, s_data, 10);

rc = bcp_sendrow(w_hdbc1);
if (rc != 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))
    BuildIndex("idxstkcl");

return;
}

//=====
//
// Function : LoadCustomer
//
//=====

void LoadCustomer()
{
    LOADER_TIME_STRUCT customer_time_start;
    LOADER_TIME_STRUCT history_time_start;
    short w_id;

    short d_id;
    DWORD dwThreadId[MAX_CUSTOMER_THREADS];
    HANDLE hThread[MAX_CUSTOMER_THREADS];
    char name[20];
    RETCODE rc;
    DBINT rcint;
    char bcphint[128];
    char cmd[256];
    // SQLRETURN
    rc_1;
    // SQLSMALLINT
    recnum, MsgLen;
    // SQLCHAR
    SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
    // SQLINTEGER
    NativeError;

    // Seed with unique number
    seed(5);

    printf("Loading customer and history tables...\n");

    // if build index before load...
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        BuildIndex("idxcuscl");

    // Initialize bulk copy
    sprintf(name, "%s.%s", aptr->database, "customer");

    rc = bcp_init(c_hdbc1, name, NULL, "logs\customer.err", DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcphint, "tablock, order c_w_id, c_d_id, c_id),

```

```

ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
rc = bcp_control(c_hdbc1, BCPHINTS, (void*)
bcphint);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);
}

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

rc = bcp_init(c_hdbc2, name, NULL, "logs\history.err", DB_IN);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc2);

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

customer_rows_loaded = 0;
history_rows_loaded = 0;

CustomerBufInit();

customer_time_start.time_start = (TimeNow() / MILLISEC);
history_time_start.time_start = (TimeNow() / MILLISEC);

for (w_id = (short)aptr->starting_warehouse; w_id <= aptr-
>num_warehouses; w_id++)
{
    for (d_id = 1; d_id <=
DISTRICT_PER_WAREHOUSE; d_id++)
    {
        CustomerBufLoad(d_id, w_id);

        // Start parallel loading threads here...

        // Start customer table thread

        printf("...Loading customer table for:
d_id = %d, w_id = %d\n", d_id, w_id);

        hThread[0] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE)
LoadCustomerTable,
&customer_time_start,
0,
&dwThreadId[0]);

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

        // Start History table thread

        printf("...Loading history table for: d_id
= %d, w_id = %d\n", d_id, w_id);

        hThread[1] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE)

```

```

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

if (CloseHandle(hThread[0]) == FALSE)
{
    printf("Error, failed in
closing customer thread handle with errno: %d\n", GetLastError());
}
if (CloseHandle(hThread[1]) == FALSE)
{
    printf("Error, failed in
closing history thread handle with errno: %d\n", GetLastError());
}
}

// flush the bulk connection
rcint = bcp_done(c_hdbc1);
if (rcint < 0)
    HandleErrorDBC(c_hdbc1);

rcint = bcp_done(c_hdbc2);
if (rcint < 0)
    HandleErrorDBC(c_hdbc2);

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

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

// build non-clustered index
if (aptr->build_index == 1)
    BuildIndex("idxcusnc");

// Output the NURAND used for the loader into C_FIRST for C_ID =
1,
// C_W_ID = 1, and C_D_ID = 1
sprintf(cmd, "isql -S%s -U%s -P%s -d%s -e -Q"update customer
set c_first = 'C_LOAD = %d' where c_id = 1 and c_w_id = 1 and c_d_id = 1" >
logs\nurand_load.log",
    aptr->server,
    aptr->user,
    aptr->password,
    aptr->database,
    LOADER_NURAND_C);

system(cmd);

SQLFreeStmt(c_hstmt1, SQL_DROP);
SQLDisconnect(c_hdbc1);

```

```

SQLFreeConnect(c_hdbc1);
SQLFreeStmt(c_hstmt2, SQL_DROP);
SQLDisconnect(c_hdbc2);
SQLFreeConnect(c_hdbc2);

return;
}

//=====
//
// Function : CustomerBufInit
//
//=====

void CustomerBufInit()
{
    int i;

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

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

        customer_buf[i].c_credit_lim = 0;
        customer_buf[i].c_discount = (float) 0;

        // fix to avoid ODBC float to numeric conversion

        // customer_buf[i].c_balance = 0;
        strcpy(customer_buf[i].c_balance,"");

        customer_buf[i].c_ytd_payment = 0;
        customer_buf[i].c_payment_cnt = 0;
        customer_buf[i].c_delivery_cnt = 0;

        strcpy(customer_buf[i].c_data,"");

        customer_buf[i].h_amount = 0;

        strcpy(customer_buf[i].h_data,"");
    }

//=====
//
// Function : CustomerBufLoad
//
// Fills shared buffer for HISTORY and CUSTOMER

```

```

//=====
void CustomerBufLoad(int d_id, int w_id)
{
    long i;
    CUSTOMER_SORT_STRUCT c{CUSTOMERS_PER_DISTRICT};

    for (i=0;i<customers_per_district;i++)
    {
        if (i < 1000)
            LastName(i, c[i].c_last);
        else
            LastName(NURand(255,0,999,LOADER_NURAND_C),c[i].c_last);

        MakeAlphaString(8,16,FIRST_NAME_LEN,
c[i].c_first);

        c[i].c_id = i+1;
    }

    printf("...Loading customer buffer for: d_id = %d, w_id = %d\n",
d_id, w_id);

    for (i=0;i<customers_per_district;i++)
    {
        customer_buf[i].c_d_id = d_id;
        customer_buf[i].c_w_id = w_id;
        customer_buf[i].h_amount = 10.0;

        customer_buf[i].c_ytd_payment = 10.0;

        customer_buf[i].c_payment_cnt = 1;
        customer_buf[i].c_delivery_cnt = 0;

        // Generate CUSTOMER and HISTORY data

        customer_buf[i].c_id = c[i].c_id;

        strcpy(customer_buf[i].c_first, c[i].c_first);
        strcpy(customer_buf[i].c_last, c[i].c_last);

        customer_buf[i].c_middle[0] = 'O';
        customer_buf[i].c_middle[1] = 'E';

        MakeAddress(customer_buf[i].c_street_1,
customer_buf[i].c_street_2,
customer_buf[i].c_city,
customer_buf[i].c_state,
customer_buf[i].c_zip);

        MakeNumberString(16, 16, PHONE_LEN,
customer_buf[i].c_phone);

        if (RandomNumber(1L, 100L) > 10)
            customer_buf[i].c_credit[0] = 'G';
        else
            customer_buf[i].c_credit[0] = 'B';
            customer_buf[i].c_credit[1] = 'C';

        customer_buf[i].c_credit_lim = 50000.0;
        customer_buf[i].c_discount = ((float)
RandomNumber(0L, 5000L) / 10000.0;

        // fix to avoid ODBC float to numeric conversion

        // customer_buf[i].c_balance = -10.0;

```

```

        strcpy(customer_buff[i].c_balance, "-10.0");
        MakeAlphaString(300, 500, C_DATA_LEN,
customer_buff[i].c_data);
        // Generate HISTORY data
        MakeAlphaString(12, 24, H_DATA_LEN,
customer_buff[i].h_data);
    }
}
//=====
//
// Function : LoadCustomerTable
//
//=====
void LoadCustomerTable(LOADER_TIME_STRUCT *customer_time_start)
{
    int i;
    long c_id;
    short c_d_id;
    short c_w_id;
    char c_first[FIRST_NAME_LEN+1];
    char c_middle[MIDDLE_NAME_LEN+1];
    char c_last[LAST_NAME_LEN+1];
    char c_street_1[ADDRESS_LEN+1];
    char c_street_2[ADDRESS_LEN+1];
    char c_city[ADDRESS_LEN+1];
    char c_state[STATE_LEN+1];
    char c_zip[ZIP_LEN+1];
    char c_phone[PHONE_LEN+1];
    char c_credit[CREDIT_LEN+1];
    double c_credit_lim;
    double c_discount;

    // fix to avoid ODBC float to numeric conversion problem.

    // double c_balance;
    char c_balance[6];

    double c_ytd_payment;
    short c_payment_cnt;
    short c_delivery_cnt;
    char c_data[C_DATA_LEN+1];
    char c_since[C_SINCE_LEN+1];
    RETCODE rc;

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 2);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0, FIRST_NAME_LEN, NULL, 0, 0,
4);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0, MIDDLE_NAME_LEN, NULL, 0,

```

```

0, 5);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN, NULL, 0, 0,
6);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN, NULL, 0, 0,
7);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN, NULL, 0, 0,
8);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL, 0, 0, 9);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0, 0, 10);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0, 11);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL, 0, 0, 12);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0, C_SINCE_LEN,
NULL, 0, SQLCHARACTER, 13);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL, 0, 0, 14);
    if (rc != SUCCEEDED)
        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_buff[i].c_id;
        c_d_id = customer_buff[i].c_d_id;
        c_w_id = customer_buff[i].c_w_id;

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

        FormatDate(&c_since);

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

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

        c_ytd_payment = customer_buff[i].c_ytd_payment;
        c_payment_cnt = customer_buff[i].c_payment_cnt;
        c_delivery_cnt = customer_buff[i].c_delivery_cnt;

        strcpy(c_data, customer_buff[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_DATE_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, 0, 8);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc2);

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

    FormatDate(&h_date);

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

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

```

```

//=====
//
// Function : LoadOrders
//
//=====
void LoadOrders()
{
    LOADER_TIME_STRUCT orders_time_start;
    LOADER_TIME_STRUCT new_order_time_start;
    LOADER_TIME_STRUCT order_line_time_start;
short w_id;

short d_id;
DWORD dwThreadId[MAX_ORDER_THREADS];
HANDLE hThread[MAX_ORDER_THREADS];
char name[20];
RETCODE rc;
char bcphint[128];

// seed with unique number
seed(6);

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

// if build index before load...
if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    BuildIndex("idxordcl");
    BuildIndex("idxnodcl");
    BuildIndex("idxodlcl");
}

// initialize bulk copy
sprintf(name, "%s.%s", aptr->database, "orders");
rc = bcp_init(o_hdbc1, name, NULL, "logs/orders.err", DB_IN);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc1);

if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    sprintf(bcphint, "tablock, order (o_w_id, o_d_id, o_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
    rc = bcp_control(o_hdbc1, BCPHINTS, (void*)
bcphint);

    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);

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

    rc = bcp_init(o_hdbc2, name, NULL, "logs/neword.err", DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcphint, "tablock, order (no_w_id, no_d_id,
no_o_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 9000));
        rc = bcp_control(o_hdbc2, BCPHINTS, (void*)
bcphint);

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

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

```

```

rc = bcp_init(o_hdbc3, name, NULL, "logs/ordline.err", DB_IN);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    sprintf(bcphint, "tablock, order (ol_w_id, ol_d_id,
ol_o_id, ol_number), ROWS_PER_BATCH = %u", (aptr->num_warehouses *
300000));
    rc = bcp_control(o_hdbc3, BCPHINTS, (void*)
bcphint);

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

    orders_rows_loaded = 0;
    new_order_rows_loaded = 0;
    order_line_rows_loaded = 0;

    OrdersBufInit();

    orders_time_start.time_start = (TimeNow() / MILLI);
    new_order_time_start.time_start = (TimeNow() / MILLI);
    order_line_time_start.time_start = (TimeNow() / MILLI);

    for (w_id = (short)aptr->starting_warehouse; w_id <= aptr-
>num_warehouses; w_id++)
    {
        for (d_id = 1; d_id <=
DISTRICT_PER_WAREHOUSE; d_id++)
        {
            OrdersBufLoad(d_id, w_id);

            // start parallel loading threads here...

            // start Orders table thread
            printf("...Loading Order Table for: d_id
= %d, w_id = %d\n", d_id, w_id);

            hThread[0] = CreateThread(NULL,
0,
(LPCTSTR)START_ROUTINE)

            &orders_time_start,
0,
&dwThreadId[0]);

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

            // start NewOrder table thread
            printf("...Loading New-Order Table for:
d_id = %d, w_id = %d\n", d_id, w_id);

            hThread[1] = CreateThread(NULL,
0,
(LPCTSTR)START_ROUTINE)

```

```

LoadNewOrderTable,
    &new_order_time_start,
    0,
    &dwThreadID[1]);
if (hThread[1] == NULL)
{
    printf("Error, failed in
creating creating thread = 1.\n");
    exit(-1);
}
// start Order-Line table thread
printf("...Loading Order-Line Table for:
d_id = %d, w_id = %d\n", d_id, w_id);
hThread[2] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE)
LoadOrderLineTable,
&order_line_time_start,
0,
&dwThreadID[2]);
if (hThread[2] == NULL)
{
    printf("Error, failed in
creating creating thread = 2.\n");
    exit(-1);
}
WaitForSingleObject( hThread[0],
INFINITE );
WaitForSingleObject( hThread[1],
INFINITE );
WaitForSingleObject( hThread[2],
INFINITE );
if (CloseHandle(hThread[0]) == FALSE)
{
    printf("Error, failed in
closing Orders thread handle with errno: %d\n", GetLastError());
}
if (CloseHandle(hThread[1]) == FALSE)
{
    printf("Error, failed in
closing NewOrder thread handle with errno: %d\n", GetLastError());
}
if (CloseHandle(hThread[2]) == FALSE)
{
    printf("Error, failed in
closing OrderLine thread handle with errno: %d\n", GetLastError());
}
}
printf("Finished loading orders.\n");
return;

```

```

}
//=====
//
// Function : OrdersBufinit
//
// Clears shared buffer for ORDERS, NEWORDER, and ORDERLINE
//=====
void OrdersBufinit()
{
    int i;
    int j;
    for (i=0;i<orders_per_district;i++)
    {
        orders_buf[i].o_id = 0;
        orders_buf[i].o_d_id = 0;
        orders_buf[i].o_w_id = 0;
        orders_buf[i].o_c_id = 0;
        orders_buf[i].o_carrier_id = 0;
        orders_buf[i].o_ol_cnt = 0;
        orders_buf[i].o_all_local = 0;
        for (j=0;j<=14;j++)
        {
            orders_buf[i].o_ol[j].ol = 0;
            orders_buf[i].o_ol[j].ol_i_id = 0;
            orders_buf[i].o_ol[j].ol_supply_w_id = 0;
            orders_buf[i].o_ol[j].ol_quantity = 0;
            orders_buf[i].o_ol[j].ol_amount = 0;
        }
        strcpy(orders_buf[i].o_ol[j].ol_dist_info, "");
    }
}
//=====
//
// Function : OrdersBufLoad
//
// Fills shared buffer for ORDERS, NEWORDER, and ORDERLINE
//=====
void OrdersBufLoad(int d_id, int w_id)
{
    int cust[ORDERS_PER_DISTRICT+1];
    long o_id;
    short ol;
    printf("...Loading Order Buffer for: d_id = %d, w_id = %d\n",
d_id, w_id);
    GetPermutation(cust, orders_per_district);
    for (o_id=0;o_id<orders_per_district;o_id++)
    {
        // Generate ORDER and NEW-ORDER data

```

```

orders_buf[o_id].o_d_id = d_id;
orders_buf[o_id].o_w_id = w_id;
orders_buf[o_id].o_id = o_id+1;
orders_buf[o_id].o_c_id = cust[o_id+1];
orders_buf[o_id].o_ol_cnt =
(short)RandomNumber(5L, 15L);
if (o_id < first_new_order)
{
    orders_buf[o_id].o_carrier_id =
(short)RandomNumber(1L, 10L);
    orders_buf[o_id].o_all_local = 1;
}
else
{
    orders_buf[o_id].o_carrier_id = 0;
    orders_buf[o_id].o_all_local = 1;
}
for (ol=0; ol<orders_buf[o_id].o_ol_cnt; ol++)
{
    orders_buf[o_id].o_ol[ol].ol = ol+1;
    orders_buf[o_id].o_ol[ol].ol_i_id =
RandomNumber(1L, max_items);
    orders_buf[o_id].o_ol[ol].ol_supply_w_id = w_id;
    orders_buf[o_id].o_ol[ol].ol_quantity = 5;
    OL_DIST_INFO_LEN, &orders_buf[o_id].o_ol[ol].ol_dist_info);
    // Generate ORDER-LINE data
    if (o_id < first_new_order)
    {
        orders_buf[o_id].o_ol[ol].ol_amount = 0;
        // Added to insure
        ol_delivery_d set properly during load
        FormatDate(&orders_buf[o_id].o_ol[ol].ol_delivery_d);
    }
    else
    {
        orders_buf[o_id].o_ol[ol].ol_amount =
RandomNumber(1,999999)/100.0;
        // Added to insure
        ol_delivery_d set properly during load
        // odbc datetime format
        strcpy(orders_buf[o_id].o_ol[ol].ol_delivery_d, "1899-12-31
00:00:00.000");
    }
}
}
//=====
//
// Function : LoadOrdersTable
//
//=====
void LoadOrdersTable(LOADER_TIME_STRUCT *orders_time_start)

```

```

{
    int i;
    long o_id;
        short o_d_id;
        short o_w_id;

    long o_c_id;
    short o_carrier_id;
    short o_ol_cnt;
    short o_all_local;
        char o_entry_d[O_ENTRY_D_LEN+1];
        RETCODE rc;
        DBINT rcint;

    // bind ORDER data
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

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

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

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

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

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 6);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 7);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 8);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    for (i = 0; i < orders_per_district; i++)
    {
        o_id = orders_buff[i].o_id;
        o_d_id = orders_buff[i].o_d_id;
        o_w_id = orders_buff[i].o_w_id;
        o_c_id = orders_buff[i].o_c_id;
        o_carrier_id = orders_buff[i].o_carrier_id;
        o_ol_cnt = orders_buff[i].o_ol_cnt;
        o_all_local = orders_buff[i].o_all_local;

        FormatDate(&o_entry_d);

        // send data to server
        rc = bcp_sendrow(o_hdbc1);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc1);
    }
}

```

```

orders_rows_loaded++;
CheckForCommit(o_hdbc1, o_hstmt1,
orders_rows_loaded, "orders", &orders_time_start->time_start);
}

// rcint = bcp_batch(o_hdbc1);
// if (rcint < 0)
//     HandleErrorDBC(o_hdbc1);

if ((o_w_id == apr->num_warehouses) && (o_d_id == 10))
{
    rcint = bcp_done(o_hdbc1);
    if (rcint < 0)
        HandleErrorDBC(o_hdbc1);

    SQLFreeStmt(o_hstmt1, SQL_DROP);
    SQLDisconnect(o_hdbc1);
    SQLFreeConnect(o_hdbc1);

    // if build index after load...
    if ((aptr->build_index == 1) && (aptr->index_order ==
0))
        BuildIndex("idxordc1");

    // build non-clustered index
    if (aptr->build_index == 1)
        BuildIndex("idxordnc");
}

}

//=====
//
// Function : LoadNewOrderTable
//
//=====
void LoadNewOrderTable(LOADER_TIME_STRUCT *new_order_time_start)
{
    int i;
    long o_id;
    short o_d_id;
    short o_w_id;
        RETCODE rc;
        DBINT rcint;

    // Bind NEW-ORDER data

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 2);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 3);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

    for (i = first_new_order; i < last_new_order; i++)
    {
        o_id = orders_buff[i].o_id;
        o_d_id = orders_buff[i].o_d_id;
        o_w_id = orders_buff[i].o_w_id;
    }
}

```

```

rc = bcp_sendrow(o_hdbc2);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc2);

new_order_rows_loaded++;

CheckForCommit(o_hdbc2, o_hstmt2,
new_order_rows_loaded, "new_order", &new_order_time_start->time_start);
}

// rcint = bcp_batch(o_hdbc2);
// if (rcint < 0)
//     HandleErrorDBC(o_hdbc2);

if ((o_w_id == apr->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("idxnodc1");

}

}

//=====
//
// Function : LoadOrderLineTable
//
//=====
void LoadOrderLineTable(LOADER_TIME_STRUCT *order_line_time_start)
{
    int i, j;
    long o_id;
        short o_d_id;
        short o_w_id;

    long ol;
        long ol_i_id;
    short ol_supply_w_id;
    short ol_quantity;
    double ol_amount;
    char ol_dist_info[DIST_INFO_LEN+1];
        char ol_delivery_d[OL_DELIVERY_D_LEN+1];
        RETCODE rc;
        DBINT rcint;

    // bind ORDER-LINE data
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 2);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
}

```



```

rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 3);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 4);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 5);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 6);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0,
OL_DELIVERY_D_LEN, NULL, 0, SQLCHARACTER, 7);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 8);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0, SQL_VARLEN_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, 0,
10);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

for (i = 0; i < orders_per_district; i++)
{
    o_id = orders_buff[i].o_id;
    o_d_id = orders_buff[i].o_d_id;
    o_w_id = orders_buff[i].o_w_id;

    for (j=0; j < orders_buff[i].o_ol_cnt; j++)
    {
        ol = orders_buff[i].o_ol[j].ol;
        ol_i_id =
            orders_buff[i].o_ol[j].ol_i_id;
        ol_supply_w_id =
            orders_buff[i].o_ol[j].ol_supply_w_id;
        ol_quantity =
            orders_buff[i].o_ol[j].ol_quantity;
        ol_amount =
            orders_buff[i].o_ol[j].ol_amount;

        strcpy(ol_delivery_d, orders_buff[i].o_ol[j].ol_delivery_d);

        strcpy(ol_dist_info, orders_buff[i].o_ol[j].ol_dist_info);

        rc = bcp_sendrow(o_hdbc3);
        if (rc != SUCCEEDED)
            HandleErrorDBC(o_hdbc3);

        order_line_rows_loaded++;
        CheckForCommit(o_hdbc3, o_hstmt3,

```

```

order_line_rows_loaded, "order_line", &order_line_time_start>time_start);
    }
}

// rcint = bcp_batch(o_hdbc3);
// if (rcint < 0)
//     HandleErrorDBC(o_hdbc3);

if ((o_w_id == aprt->num_warehouses) && (o_d_id == 10))
{
    rcint = bcp_done(o_hdbc3);
    if (rcint < 0)
        HandleErrorDBC(o_hdbc3);

    SQLFreeStmt(o_hstmt3, SQL_DRCP);
    SQLDisconnect(o_hdbc3);
    SQLFreeConnect(o_hdbc3);

    // if build index after load...
    if ((aprt->build_index == 1) && (aprt->index_order ==
0))
        BuildIndex("idxodc!");
}

}

//=====
//
// Function : GetPermutation
//
//=====
void GetPermutation(int perm[], int n)
{
    int i, r, t;

    for (i=1; i<=n; i++)
        perm[i] = i;

    for (i=1; i<=n; i++)
    {
        r = RandomNumber(i,n);
        t = perm[i];
        perm[i] = perm[r];
        perm[r] = t;
    }
}

//=====
//
// Function : CheckForCommit
//
//=====
void CheckForCommit(HDBC hdbc,
                    HSTMT
                    hstmt,
                    int rows_loaded,
                    char
                    *table_name,
                    long *time_start)
{

```

```

long    time_end, time_diff,
        // DBINT    rcint;

if ( !(rows_loaded % 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 in %ld sec- Total
= %d (%.2f rps)\n",
        aprt->batch,
        table_name,
        time_diff,
        rows_loaded,
        (float) aprt->batch /
(time_diff ? time_diff : 1L));

    *time_start = time_end;
}

return;
}

//=====
//
// Function : OpenConnections
//
//=====
void OpenConnections()
{
    RETCODE    rc;

    char
    char
    SQLSMALLINT
        szDriverString[300];
        szDriverStringOut[1024];
        cbDriverStringOut;

    SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE,
&henv );

    SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION,
(void*)SQL_OV_ODBC3, 0);

    SQLAllocHandle(SQL_HANDLE_DBC, henv , &i_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv , &w_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc2);
    SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc2);
    SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc3);

    SQLSetConnectAttr(i_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
    SQLSetConnectAttr(w_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
    SQLSetConnectAttr(c_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
    SQLSetConnectAttr(c_hdbc2, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
    SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
    SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );

```

```

        SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER);
        SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER);

        // Open connections to SQL Server

        // Connection 1

        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );

        rc = SQLSetConnectOption (i_hdbc1, SQL_PACKET_SIZE, aptr
>pack_size);
        if (rc != SUCCEED)
            HandleErrorDBC(i_hdbc1);

        rc = SQLDriverConnect ( i_hdbc1,                NULL,

        (SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,
        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
            HandleErrorDBC(i_hdbc1);

        // Connection 2

        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );

        rc = SQLSetConnectOption (w_hdbc1, SQL_PACKET_SIZE, aptr
>pack_size);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        rc = SQLDriverConnect ( w_hdbc1,

        NULL,
        (SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,
        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),

```

```

        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        // Connection 3

        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );

        rc = SQLSetConnectOption (c_hdbc1, SQL_PACKET_SIZE, aptr
>pack_size);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);

        rc = SQLDriverConnect ( c_hdbc1,

        NULL,
        (SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,
        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);

        // Connection 4

        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );

        rc = SQLSetConnectOption (c_hdbc2, SQL_PACKET_SIZE, aptr
>pack_size);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);

        rc = SQLDriverConnect ( c_hdbc2,

        NULL,
        (SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,
        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),

```

```

        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);

        // Connection 5

        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );

        rc = SQLSetConnectOption (o_hdbc1, SQL_PACKET_SIZE, aptr
>pack_size);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc1);

        rc = SQLDriverConnect ( o_hdbc1,

        NULL,
        (SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,
        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc1);

        // Connection 6

        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );

        rc = SQLSetConnectOption (o_hdbc2, SQL_PACKET_SIZE, aptr
>pack_size);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc2);

        rc = SQLDriverConnect ( o_hdbc2,

        NULL,
        (SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,
        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),

```

```

        &cbDriverStringOut,

        SQL_DRIVER_NOPROMPT );
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

    // Connection 7

    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );

    rc = SQLSetConnectOption ( o_hdbc3, SQL_PACKET_SIZE, aptr
>pack_size);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = SQLDriverConnect ( o_hdbc3,
NULL,
(SQLCHAR*)&szDriverString[0] ,
SQL_NTS,
(SQLCHAR*)&szDriverStringOut[0],
sizeof(szDriverStringOut),
&cbDriverStringOut,
SQL_DRIVER_NOPROMPT );
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
}

//=====
//
// Function name: BuildIndex
//
//=====
void BuildIndex(char *index_script)
{
    char cmd[256];

    printf("Starting index creation: %s\n",index_script);
    sprintf(cmd, "isql -S%s -U%s -P%s -e -i%s\\%s.sql > logs\\%s.log",
aptr->server,
aptr->user,
aptr->password,
aptr->index_script_path,
index_script,
index_script);

    system(cmd);

    printf("Finished index creation: %s\n",index_script);
}

```

```

void HandleErrorDBC (SQLHDBC hdbc1)
{
    SQLCHAR SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
SQLINTEGER NativeError;
SQLSMALLINT i, MsgLen;
SQLRETURN rc2;
char timebuf[128];
char datebuf[128];
FILE *fp1;

    i = 1;
    while (( rc2 = SQLGetDiagRec(SQL_HANDLE_DBC , hdbc1, i,
SqlState , &NativeError,
Msg, sizeof(Msg) ,
&MsgLen )) != SQL_NO_DATA )
    {
        sprintf( szLastError , "%s" , Msg );

        _strtime(timebuf);
        _strdate(datebuf);

        printf( "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);

        fp1 = fopen("logs\\tpccldr.err", "w");
        if (fp1 == NULL)
            printf("ERROR: Unable to open errorlog
file.\n");
        else
        {
            fprintf(fp1, "[%s : %s] %s\n" , datebuf,
timebuf, szLastError);
            fclose(fp1);
        }
        i++;
    }

void HandleErrorSTMT (HSTMT hstmt1)
{
    SQLCHAR SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
SQLINTEGER NativeError;
SQLSMALLINT i, MsgLen;
SQLRETURN rc2;
char timebuf[128];
char datebuf[128];
FILE *fp1;

    i = 1;
    while (( rc2 = SQLGetDiagRec(SQL_HANDLE_STMT , hstmt1, i,
SqlState , &NativeError,
Msg, sizeof(Msg) ,
&MsgLen )) != SQL_NO_DATA )
    {
        sprintf( szLastError , "%s" , Msg );

        _strtime(timebuf);
        _strdate(datebuf);

        printf( "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);

        fp1 = fopen("logs\\tpccldr.err", "w");
        if (fp1 == NULL)

```

```

            printf("ERROR: Unable to open errorlog
file.\n");
            else
            {
                fprintf(fp1, "[%s : %s] %s\n" , datebuf,
timebuf, szLastError);
                fclose(fp1);
            }
        }
        i++;
    }

void FormatDate ( char* szTimeCOutput )
{
    struct tm when;
    time_t now;

    time( &now );
    when = *localtime( &now );

    mktime( &when );

    // odbc datetime format
    strftime( szTimeCOutput , 30 , "%Y-%m-%d %H:%M:%S.000",
&when );

    return;
}

//=====
//
// Function : CheckDataBase
//
//=====
void CheckDataBase()
{
    RETCODE rc;

    char szDriverString[300];
    char szDriverStringOut[1024];
    char TablesBitMap[9] =
{"000000000"};
    int i, ExitFlag;

    SQLSMALLINT cbDriverStringOut;
    SQLCHAR TabName[10];
    SQLINTEGER TabNameInd, TabCount,
TabCountInd;

    ExitFlag = 0;

    SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE,
&henv );

    SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION,
(void*)SQL_OV_ODBC3, 0 );

    SQLAllocHandle(SQL_HANDLE_DBC, henv , &v_hdbc);

    SQLSetConnectAttr(v_hdbc, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );

```

```

// Open connection to SQL Server
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );
rc = SQLSetConnectAttr( v_hdbc, SQL_ATTR_PACKET_SIZE,
(SQLPOINTER)aptr->pack_size, SQL_IS_UIINTEGER );
if (rc != SQL_SUCCESS)
    HandleErrorDBC(v_hdbc);
rc = SQLDriverConnect ( v_hdbc,
NULL,
(SQLCHAR*)&szDriverString[0] ,
SQL_NTS,
(SQLCHAR*)&szDriverStringOut[0],
sizeof(szDriverStringOut),
&cbDriverStringOut,
SQL_DRIVER_NOPROMPT );
// if the rc is SQL_ERROR, the the TPCC database probably does
not exist
if (rc == SQL_ERROR)
{
    printf("The database TPCC does not appear to
exist!\n");
    printf("\nCheck LOGS\ directory for database
creation errors.\n");
    // cleanup database connections and handles
    SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
    SQLDisconnect(v_hdbc);
    SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);
    // since there is not a database, exit back to
    SETUP.CMD
    exit(1);
}
if ( SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt) !=
SQL_SUCCESS )
    HandleErrorDBC(v_hdbc);
if ( SQLBindCol(v_hstmt, 1, SQL_C_ULONG, &TabCount, 0,
&TabCountInd) != SQL_SUCCESS )
    HandleErrorSTMT(v_hstmt);
// count the number of user tables from sysobjects
rc = SQLExecDirect(v_hstmt, "select count(*) from sysobjects
where xtype =\U" , SQL_NTS);
if ((rc != SQL_SUCCESS) && (rc !=
SQL_SUCCESS_WITH_INFO))
    HandleErrorSTMT(v_hstmt);
if ( SQLFetch(v_hstmt) != SQL_SUCCESS )
    HandleErrorSTMT(v_hstmt);
// if the number of tables is less than 9, select all the user tables in
TPCC

```

```

if (TabCount != 9)
{
    SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
    SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc ,
&v_hstmt);
    if ( SQLBindCol(v_hstmt, 1, SQL_C_CHAR,
&TabName, sizeof(TabName), &TabNameInd) != SQL_SUCCESS )
        HandleErrorSTMT(v_hstmt);
    // select the list of user tables into a result set
    rc = SQLExecDirect(v_hstmt, "select * from
sysobjects where xtype =\U" , SQL_NTS);
    if ((rc != SQL_SUCCESS) && (rc !=
SQL_SUCCESS_WITH_INFO))
        HandleErrorSTMT(v_hstmt);
    // go through the result set and set the bitmap for
    each found table
    // set the bitmap to '1' if the table name is found
    while ((rc = SQLFetch(v_hstmt)) != SQL_NO_DATA)
    {
        switch( TabName[0] )
        {
            case 'w':
                TablesBitMap[0] = '1';
                break;
            case 'd':
                TablesBitMap[1] = '1';
                break;
            case 'c':
                TablesBitMap[2] = '1';
                break;
            case 'h':
                TablesBitMap[3] = '1';
                break;
            case 'n':
                TablesBitMap[4] = '1';
                break;
            case 'o':
                if (TabName[5] = 's')
                    TablesBitMap[5] = '1';
                if (TabName[5] = '_')
                    TablesBitMap[6] = '1';
            case 'i':
                TablesBitMap[7] = '1';
                break;
            case 's':
                TablesBitMap[8] = '1';
                break;
        }
    }
    // a '0' ExitFlag means do NOT exit the loader early, a
    '1' means exit the loader early
    ExitFlag = 0;
    // iterate through the bitmap to display which
    table(s) is actually missing
    for (i = 0; i <= 8; i++)
    {
        switch(i)
        {
            case 0:
                if (TablesBitMap[i] == '0')
                {
                    printf("The

```

```

Warehouse table is missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        case 1:
            if (TablesBitMap[i] == '0')
            {
                printf("The
District table is missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        case 2:
            if (TablesBitMap[i] == '0')
            {
                printf("The
Customer table is missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        case 3:
            if (TablesBitMap[i] == '0')
            {
                printf("The
History table is missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        case 4:
            if (TablesBitMap[i] == '0')
            {
                printf("The
New_Order table is missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        case 5:
            if (TablesBitMap[i] == '0')
            {
                printf("The
Orders table is missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        case 6:
            if (TablesBitMap[i] == '0')
            {
                printf("The
Order_Line table is missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        case 7:
            if (TablesBitMap[i] == '0')
            {
                printf("The
Item table is missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        case 8:
            if (TablesBitMap[i] == '0')
            {
                printf("The
Stock table is missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        }
    }
    // if one or more tables are missing, display message

```

```

and exit the loader
        if (ExitFlag = 1)
        {
                printf("\nExiting TPC-C Loader!\n");
                printf("\nCheck LOGS\ directory for
database\n");
                printf("or table creation errors.\n");

                // cleanup database connections and
handles
                SQLFreeHandle(SQL_HANDLE_STMT,
v_hstmt);
                SQLDisconnect(v_hdbc);
                SQLFreeHandle(SQL_HANDLE_DBC,
v_hdbc);

                exit(1);
        }

        // cleanup database connections and handles
        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
        SQLDisconnect(v_hdbc);
        SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

        return;
}

```

Appendix C : Tunable Parameters

RTE input parameter

The following parameters were used with Microsoft BenchCraft RTE..

Profile: nectpcc4cl2520w
File Path: C:\benchcrf\nectpcc4cl2520w.pro
Version: 1.0.1

Number of Engines: 28

Name: DRIVER1
Description: r1-1
Directory: c:\11.out
Machine: cl01
Parameter Set: PARAM2
Index: 0
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER13218078
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER10
Description: r3-2
Directory: c:\32.out
Machine: cl03
Parameter Set: PARAM2
Index: 45000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER104635171
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER11
Description: r4-2
Directory: c:\42.out
Machine: cl04
Parameter Set: PARAM2
Index: 50000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER114658109
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER12
Description: r5-2
Directory: c:\52.out
Machine: cl05

Parameter Set: PARAM2
Index: 55000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER124712578
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER13
Description: r6-2
Directory: c:\62.out
Machine: cl06
Parameter Set: PARAM2
Index: 60000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER134742437
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER14
Description: r7-2
Directory: c:\72.out
Machine: cl07
Parameter Set: PARAM2
Index: 65000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER144762515
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER15
Description: r1-3
Directory: c:\13.out
Machine: cl01
Parameter Set: PARAM2
Index: 70000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER154787125
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER16
Description: r2-3
Directory: c:\23.out
Machine: cl02
Parameter Set: PARAM2
Index: 75000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER164817640
Connect Rate: 100

Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER17
Description: r3-3
Directory: c:\33.out
Machine: cl03
Parameter Set: PARAM2
Index: 80000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER174850515
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER18
Description: r4-3
Directory: c:\43.out
Machine: cl04
Parameter Set: PARAM2
Index: 85000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER184875750
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER19
Description: r5-3
Directory: c:\53.out
Machine: cl05
Parameter Set: PARAM2
Index: 90000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER194902984
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER2
Description: r2-1
Directory: c:\21.out
Machine: cl02
Parameter Set: PARAM2
Index: 50000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER23286218
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER20
Description: r6-3

Directory: c:\63.out
Machine: cl06
Parameter Set: PARAM2
Index: 95000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER204928281
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER21
Description: r7-3
Directory: c:\73.out
Machine: cl07
Parameter Set: PARAM2
Index: 100000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER214962937
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER22
Description: r1-4
Directory: c:\14.out
Machine: cl01
Parameter Set: PARAM2
Index: 105000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER224988515
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER23
Description: r2-4
Directory: c:\24.out
Machine: cl02
Parameter Set: PARAM2
Index: 110000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER235015671
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER24
Description: r3-4
Directory: c:\34.out
Machine: cl03
Parameter Set: PARAM2
Index: 115000000
Seed: 32558
Configured Users: 900

Pipe Name: DRIVER245045312
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER25
Description: r4-4
Directory: c:\44.out
Machine: cl04
Parameter Set: PARAM2
Index: 120000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER255093406
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER26
Description: r5-4
Directory: c:\54.out
Machine: cl05
Parameter Set: PARAM2
Index: 125000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER265125296
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER27
Description: r6-4
Directory: c:\64.out
Machine: cl06
Parameter Set: PARAM2
Index: 130000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER275143484
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER28
Description: r7-4
Directory: c:\74.out
Machine: cl07
Parameter Set: PARAM2
Index: 135000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER285162250
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER3
Description: r3-1
Directory: c:\31.out
Machine: cl03
Parameter Set: PARAM2
Index: 100000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER34372625
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER4
Description: r4-1
Directory: c:\41.out
Machine: cl04
Parameter Set: PARAM2
Index: 150000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER44404234
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER5
Description: r5-1
Directory: c:\51.out
Machine: cl05
Parameter Set: PARAM2
Index: 200000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER54436546
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER6
Description: r6-1
Directory: c:\61.out
Machine: cl06
Parameter Set: PARAM2
Index: 250000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER64498953
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER7
Description: r7-1
Directory: c:\71.out
Machine: cl07
Parameter Set: PARAM2
Index: 300000000

Seed: 32558
Configured Users: 900
Pipe Name: DRIVER74535546
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER8
Description: r1-2
Directory: c:\12.out
Machine: cl01
Parameter Set: PARAM2
Index: 35000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER84584265
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER9
Description: r2-2
Directory: c:\22.out
Machine: cl02
Parameter Set: PARAM2
Index: 40000000
Seed: 32558
Configured Users: 900
Pipe Name: DRIVER94609718
Connect Rate: 100
Start Rate: 0
CLIENT_NURAND: 233
CPU: 0

Number of User groups: 28

Driver Engine: DRIVER1
IIS Server: acl01
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 1 - 90
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER10
IIS Server: acl02
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 811 - 900
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER11
IIS Server: acl02
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 901 - 990
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER12
IIS Server: acl02
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 991 - 1080
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER13
IIS Server: acl02
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 1081 - 1170
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER14
IIS Server: acl02
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 1171 - 1260
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER15
IIS Server: acl03
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 1261 - 1350
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER16
IIS Server: acl03

SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 1351 - 1440
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER17
IIS Server: acl03
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 1441 - 1530
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER18
IIS Server: acl03
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 1531 - 1620
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER19
IIS Server: acl03
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 1621 - 1710
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER2
IIS Server: acl01
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 91 - 180
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER20
IIS Server: acl03
SQL Server: tpc02
User: sa

Protocol: Html
w_id Range: 1711 - 1800
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER21
IIS Server: acl03
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 1801 - 1890
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER22
IIS Server: acl04
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 1891 - 1980
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER23
IIS Server: acl04
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 1981 - 2070
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER24
IIS Server: acl04
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 2071 - 2160
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER25
IIS Server: acl04
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 2161 - 2250

w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER26
IIS Server: acl04
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 2251 - 2340
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER27
IIS Server: acl04
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 2341 - 2430
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER28
IIS Server: acl04
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 2431 - 2520
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER3
IIS Server: acl01
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 181 - 270
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER4
IIS Server: acl01
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 271 - 360
w_id Max Warehouse: 2520
Scale: Normal

User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER5
IIS Server: acl01
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 361 - 450
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER6
IIS Server: acl01
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 451 - 540
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER7
IIS Server: acl01
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 541 - 630
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER8
IIS Server: acl02
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 631 - 720
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1
Scale Down: No

Driver Engine: DRIVER9
IIS Server: acl02
SQL Server: tpc02
User: sa
Protocol: Html
w_id Range: 721 - 810
w_id Max Warehouse: 2520
Scale: Normal
User Count: 900
District id: 1

Scale Down: No

Number of Parameter Sets: 2

		PARAM2 TPCC1	Txn	Think	Key	RT	RT
Menu							
Delay			Weight	Time	Time	Delay	Fence
		New Order		44.88	12.05	18.01	
0.10	5.00	0.10					
		Payment		43.03	12.05	3.01	
0.10	5.00	0.10					
		Delivery		4.03	5.05	2.01	
0.10	5.00	0.10					
		Stock Level		4.03	5.05	2.01	
0.10	20.00	0.10					
		Order Status		4.03	10.05	2.01	
0.10	5.00	0.10					
~Default							
Default Parameter Set							
Menu			Txn	Think	Key	RT	RT
Delay			Weight	Time	Time	Delay	Fence
		New Order		10.00	12.05	18.01	
0.10	5.00	0.10					
		Payment		10.00	12.05	3.01	
0.10	5.00	0.10					
		Delivery		1.00	5.05	2.01	
0.10	5.00	0.10					
		Stock Level		1.00	5.05	2.01	
0.10	20.00	0.10					
		Order Status		1.00	10.05	2.01	
0.10	5.00	0.10					

<Server Configuration>

Microsoft Windows 2000 Advanced Server Configuration Parameters

The following services were disabled or stopped in the Windows NT Control Panel/Service:

- Alerter
- Application Management
- Computer Browser
- Distributed File System
- DHCP Client
- DNS Client
- IIS Admin Service
- Intersite Messaging

- Kerberos Key Distribution Center
- License Logging Service
- Distributed Transaction Coordinator
- Microsoft Search
- Network DDE
- Network DDE DSDM
- Removable Storage
- IPSEC Policy Agent
- Remote Access Auto Connection Manager
- Remote Access Connection Manager
- Routing and Remote Access
- Remote Registry Service
- Task Scheduler
- Terminal Services Licensing
- Distributed Link Tracking Client
- World Wide Web Publishing Service
- Windows Time
- Utility Manager
- Uninterruptible Power Supply
- Distributed Link Tracking Server
- Telnet
- Terminal Services
- Telephony
- Performance Logs and Alerts
- Print Spooler
- Simple Mail Transport Protocol
- Internet Connection Sharing
- System Event Notification
- RunAs Service
- Smart Card Helper
- Smart Card
- Security Accounts Manager
- QoS RSVP
- File Replication
- Network Connections
- Net Logon
- Windows Installer
- Messenger
- TCP/IP NetBIOS Helper Service
- Server
- Fax Service
- COM+ Event System
- Indexing Service
- ClipBook

BOOT. INI

The /3gb switch was added to the boot. ini file to cause NT Enterprise Server to allow 3GB of user and 1GB of kernel virtual address space, rather than the usual 2GB of virtual address space.

The /PAE switch was added to the boot.ini file to cause Windows 2000 to support more than 4GB of physical memory.

NT Registry

The Registry keys are modified as follows;

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\Parameters]
"ProcessorAffinityMask"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SessionManager\I/O System]
"CountOperations"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SessionManager\Memory Management]
"LargeSystemCache"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PerfDisk]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PerfDisk\Performance]
"Open Timeout"=dword:00002710
"Collect Timeout"=dword:00001f40
"Disable Performance Counters"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PerfOS]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PerfOS\Performance]
"Open Timeout"=dword:00002710
"Collect Timeout"=dword:00001f40
"Disable Performance Counters"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PerfProc]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PerfProc\Performance]
"Open Timeout"=dword:00002710
"Collect Timeout"=dword:00001f40
"Disable Performance Counters"=dword:00000001
```

System Configuration Report

```
[system Information]
Item Value
OS Name Microsoft Windows 2000 Advanced Server
Version 5.0.2195 Build 2195
OS Manufacturer Microsoft Corporation
System Name TPC02
System Manufacturer NEC
System Model Express5800/140Hb
System Type X86-based PC
Processor x86 Family 6 Model 10 Stepping 0 GenuineIntel ~700 Mhz
Processor x86 Family 6 Model 10 Stepping 0 GenuineIntel ~700 Mhz
Processor x86 Family 6 Model 10 Stepping 0 GenuineIntel ~700 Mhz
Processor x86 Family 6 Model 10 Stepping 0 GenuineIntel ~700 Mhz
BIOS Version PhoenixBIOS 4.0 Release 6.0.0021
Windows Directory C:\WINNT
System Directory C:\WINNT\System32
Boot Device \Device\Harddisk0\Partition1
```

Locale United States
 User Name TPC02\Administrator
 Total Physical Memory 7,863,592 KB
 Available Physical Memory 7,679,792 KB
 Total Virtual Memory 17,682,712 KB
 Available Virtual Memory 17,443,264 KB
 Page File Space 9,819,120 KB
 Page File C:\pagefile.sys

[Software Environment]

Name	Description	File	Type	Started	Start Mode
	State	Status	Error Control	Accept	Pause
	Accept Stop				
aac	Adaptec Aac Driver	c:\winnt\system32\drivers\aac.sys	Kernel Driver	True	Running OK
abiosdsk	Abiosdsk	Not Available	Kernel Driver	False	Disabled
abp480n5	abp480n5	Not Available	Kernel Driver	False	Disabled
acpi	ACPI	Not Available	Kernel Driver	False	Disabled
acpiec	ACPIEC	Not Available	Kernel Driver	False	Disabled
adpu160m	adpu160m	c:\winnt\system32\drivers\adpu160m.sys	Kernel Driver	True	Running OK
afd	AFD Networking Support Environment	c:\winnt\system32\drivers\afd.sys	Kernel Driver	True	Running OK
aha154x	Aha154x	Not Available	Kernel Driver	False	Disabled
aic116x	aic116x	Not Available	Kernel Driver	False	Disabled
aic78u2	aic78u2	Not Available	Kernel Driver	False	Disabled
aic78xx	aic78xx	c:\winnt\system32\drivers\aic78xx.sys	Kernel Driver	True	Running OK
ami0nt	ami0nt	Not Available	Kernel Driver	False	Disabled
amsint	amsint	Not Available	Kernel Driver	False	Disabled
asc	asc	Not Available	Kernel Driver	False	Disabled
asc3350p	asc3350p	Not Available	Kernel Driver	False	Disabled
asc3550	asc3550	Not Available	Kernel Driver	False	Disabled
asynccmac	RAS Asynchronous Media Driver	c:\winnt\system32\drivers\asynccmac.sys	Kernel Driver	False	Manual Stopped OK
atapi	Standard IDE/ESDI Hard Disk Controller	c:\winnt\system32\drivers\atapi.sys	Kernel Driver	True	Running OK
atdisk	Atdisk	Not Available	Kernel Driver	False	Disabled
atirage_	atirage_	c:\winnt\system32\drivers\atiragem.sys	Kernel Driver	False	Manual Stopped OK
atmarpc	ATM ARP Client Protocol	c:\winnt\system32\drivers\atmarpc.sys	Kernel Driver	False	Manual Stopped OK
audstub	Audio Stub Driver	c:\winnt\system32\drivers\audstub.sys	Kernel Driver	True	Manual Running OK

beep	Beep	c:\winnt\system32\drivers\beep.sys	Kernel Driver	True	System Running OK
buslogic	BusLogic	Not Available	Kernel Driver	False	Disabled
cd20xrnt	cd20xrnt	Not Available	Kernel Driver	False	Disabled
cdaudio	Cdaudio	c:\winnt\system32\drivers\cdaudio.sys	Kernel Driver	False	System Stopped OK
cdfs	Cdfs	c:\winnt\system32\drivers\cdfs.sys	File System	True	Disabled Running OK
cdrom	CD-ROM Driver	c:\winnt\system32\drivers\cdrom.sys	Kernel Driver	True	System Running OK
changer	Changer	Not Available	Kernel Driver	False	System
cpqarray	Cpqarray	Not Available	Kernel Driver	False	Disabled
cpqarray2	cpqarray2	Not Available	Kernel Driver	False	Disabled
cpqfcalm	cpqfcalm	Not Available	Kernel Driver	False	Disabled
cpqfws2e	cpqfws2e	Not Available	Kernel Driver	False	Disabled
dac2w2k	dac2w2k	c:\winnt\system32\drivers\dac2w2k.sys	Kernel Driver	True	Running OK
dac960nt	dac960nt	Not Available	Kernel Driver	False	Disabled
deckzpsx	deckzpsx	Not Available	Kernel Driver	False	Disabled
dfsdriver	DfsDriver	c:\winnt\system32\drivers\dfs.sys	File System	True	Boot Running OK
disk	Disk Driver	c:\winnt\system32\drivers\disk.sys	Kernel Driver	True	Boot Running OK
diskperf	Diskperf	c:\winnt\system32\drivers\diskperf.sys	Kernel Driver	False	Disabled Stopped OK
dmboot	dmboot	c:\winnt\system32\drivers\dmboot.sys	Kernel Driver	False	Disabled Stopped OK
dmio	Logical Disk Manager Driver	c:\winnt\system32\drivers\dmio.sys	Kernel Driver	True	Boot Running OK
dmload	dmload	c:\winnt\system32\drivers\dmload.sys	Kernel Driver	True	Boot Running OK
e100b	Intel PRO Adapter Driver	c:\winnt\system32\drivers\le100bnt5.sys	Kernel Driver	True	Manual Running OK
efs	EFS	c:\winnt\system32\drivers\efs.sys	File System	True	Disabled Running OK
fastfat	Fastfat	c:\winnt\system32\drivers\fastfat.sys	File System	True	Disabled Running OK
fd16_700	Fd16_700	Not Available	Kernel Driver	False	Disabled
fdc	Floppy Disk Controller Driver	c:\winnt\system32\drivers\fdc.sys	Kernel Driver	True	Manual Running OK
fireport	fireport	Not Available	Kernel Driver	False	Disabled
flashpnt	flashpnt	Not Available	Kernel Driver	False	Disabled

flpydisk	Floppy Disk Driver	c:\winnt\system32\drivers\flpydisk.sys	Kernel Driver	True	Manual Running OK
ftdisk	Volume Manager Driver	c:\winnt\system32\drivers\ftdisk.sys	Kernel Driver	True	Boot Running OK
gamdrv	gamdrv	c:\winnt\system32\drivers\gamdrv.sys	Kernel Driver	True	Disabled Stopped OK
gpc	Generic Packet Classifier	c:\winnt\system32\drivers\msgpc.sys	Kernel Driver	True	Manual Running OK
i804prt	i8042 Keyboard and PS/2 Mouse Port Driver	c:\winnt\system32\drivers\i8042prt.sys	Kernel Driver	True	System Running OK
ini910u	ini910u	Not Available	Kernel Driver	False	Disabled
intelide	IntelIde	Not Available	Kernel Driver	False	Disabled
ipfilterdriver	IP Traffic Filter Driver	c:\winnt\system32\drivers\ipfilter.sys	Kernel Driver	False	Manual Stopped OK
ipinip	IP in IP Tunnel Driver	c:\winnt\system32\drivers\ipinip.sys	Kernel Driver	False	Manual Stopped OK
ipnat	IP Network Address Translator	c:\winnt\system32\drivers\ipnat.sys	Kernel Driver	False	Manual Stopped OK
ipsec	IPSEC driver	c:\winnt\system32\drivers\ipsec.sys	Kernel Driver	False	Manual Stopped OK
ipsraidn	ipsraidn	Not Available	Kernel Driver	False	Disabled
isapnp	PnP ISA/EISA Bus Driver	c:\winnt\system32\drivers\isapnp.sys	Kernel Driver	True	Boot Running OK
kbdclass	Keyboard Class Driver	c:\winnt\system32\drivers\kbdclass.sys	Kernel Driver	True	System Running OK
ksecdd	KSecDD	c:\winnt\system32\drivers\ksecdd.sys	Kernel Driver	True	Boot Running OK
lbrtfdc	lbrtfdc	Not Available	Kernel Driver	False	System
lp6nds35	lp6nds35	Not Available	Kernel Driver	False	Disabled
macdisk	macdisk	c:\winnt\system32\drivers\mac2w2k.sys	Kernel Driver	True	Boot Running OK
mnmdd	mnmdd	c:\winnt\system32\drivers\mnmdd.sys	Kernel Driver	True	System Running OK
modem	Modem	c:\winnt\system32\drivers\modem.sys	Kernel Driver	False	Manual Stopped OK
mouclass	Mouse Class Driver	c:\winnt\system32\drivers\mouclass.sys	Kernel Driver	True	System Running OK
mountmgr	MountMgr	c:\winnt\system32\drivers\mountmgr.sys	Kernel Driver	True	Boot Running OK
mraid35x	mraid35x	Not Available	Kernel Driver	False	Disabled
mrxsmb	MRXSMB	c:\winnt\system32\drivers\mrxsmb.sys	File System	True	System Running OK
msfs	Msf	c:\winnt\system32\drivers\msfs.sys	File System	True	System Running OK

mksrsv	Microsoft Streaming Service Proxy c:\winnt\system32\drivers\mksrsv.sys	Kernel Driver	False	Normal	False
mspclock	Microsoft Streaming Clock Proxy c:\winnt\system32\drivers\mspclock.sys	Kernel Driver	False	Normal	False
mspqm	Microsoft Streaming Quality Manager Proxy c:\winnt\system32\drivers\mspqm.sys	Kernel Driver	False	Normal	False
mup Driver	Mup c:\winnt\system32\drivers\mup.sys	File System	True	Normal	OK
ncrc710	Ncrc710 Not Available	Kernel Driver	False	Disabled	False
ndis	NDIS System Driver c:\winnt\system32\drivers\ndis.sys	Kernel Driver	True	Running	OK
ndistapi	Remote Access NDIS TAPI Driver c:\winnt\system32\drivers\ndistapi.sys	Kernel Driver	True	Normal	False
ndiswan	Remote Access NDIS WAN Driver c:\winnt\system32\drivers\ndiswan.sys	Kernel Driver	True	Normal	False
ndproxy	NDIS Proxy c:\winnt\system32\drivers\ndproxy.sys	Kernel Driver	True	Normal	OK
netbios	NetBIOS Interface c:\winnt\system32\drivers\netbios.sys	File System Driver	True	Running	OK
netbt	NetBios over Tcpip c:\winnt\system32\drivers\netbt.sys	Kernel Driver	True	Running	OK
netdetect	NetDetect c:\winnt\system32\drivers\netdetect.sys	Kernel Driver	False	Normal	OK
npfs Driver	Npfs c:\winnt\system32\drivers\npfs.sys	File System	True	Normal	OK
ntfs Driver	Ntfs c:\winnt\system32\drivers\ntfs.sys	File System	True	Normal	OK
null	Null c:\winnt\system32\drivers\null.sys	Kernel Driver	True	Normal	OK
nwlnkflt	IPX Traffic Filter Driver c:\winnt\system32\drivers\nwlnkflt.sys	Kernel Driver	False	Stopped	OK
nwlnkfwd	IPX Traffic Forwarder Driver c:\winnt\system32\drivers\nwlnkfwd.sys	Kernel Driver	False	Normal	False
parallel	Parallel class driver c:\winnt\system32\drivers\parallel.sys	Kernel Driver	True	Running	OK
parport	Parallel port driver c:\winnt\system32\drivers\parport.sys	Kernel Driver	True	Running	OK
partmgr	PartMgr c:\winnt\system32\drivers\partmgr.sys	Kernel Driver	True	Running	OK
parvdm	ParVdm c:\winnt\system32\drivers\parvdm.sys	Kernel Driver	True	Running	OK
pci	PCI Bus Driver c:\winnt\system32\drivers\pci.sys	Kernel Driver	True	Running	OK

pcidump	PCIDump Not Available	Kernel Driver	False	System	False
pcide	PCIDe c:\winnt\system32\drivers\pcide.sys	Kernel Driver	True	Running	OK
pcmcia	Pcmcia c:\winnt\system32\drivers\pcmcia.sys	Kernel Driver	False	Stopped	OK
pdcomp	PDCOMP Not Available	Kernel Driver	False	Manual	False
pdframe	PDFRAME Not Available	Kernel Driver	False	Manual	False
pdrel	PDRELI Not Available	Kernel Driver	False	Manual	False
pdframe	PDRFRAME Not Available	Kernel Driver	False	Manual	False
pptpminiport	WAN Miniport (PPTP) c:\winnt\system32\drivers\rasppp.sys	Kernel Driver	True	Running	OK
ptiink	Direct Parallel Link Driver c:\winnt\system32\drivers\ptiink.sys	Kernel Driver	True	Running	OK
ql1080	ql1080 Not Available	Kernel Driver	False	Disabled	False
ql10wnt	ql10wnt Not Available	Kernel Driver	False	Disabled	False
ql1240	ql1240 Not Available	Kernel Driver	False	Disabled	False
ql2100	ql2100 Not Available	Kernel Driver	False	Disabled	False
rasacd	Remote Access Auto Connection Driver c:\winnt\system32\drivers\rasacd.sys	Kernel Driver	True	Running	OK
rasl2tp	WAN Miniport (L2TP) c:\winnt\system32\drivers\rasl2tp.sys	Kernel Driver	True	Running	OK
raspti	Direct Parallel c:\winnt\system32\drivers\raspti.sys	Kernel Driver	True	Running	OK
rca	Microsoft Streaming Network Raw Channel Access c:\winnt\system32\drivers\rca.sys	Kernel Driver	False	Normal	False
rbdbss Driver	Rbdbss c:\winnt\system32\drivers\rbdbss.sys	File System	True	Running	OK
rdpwd	RDPWD c:\winnt\system32\drivers\rdpwd.sys	Kernel Driver	False	Stopped	OK
redbook	Digital CD Audio Playback Filter Driver c:\winnt\system32\drivers\redbook.sys	Kernel Driver	False	Normal	False
serenum	Serenum Filter Driver c:\winnt\system32\drivers\serenum.sys	Kernel Driver	True	Running	OK
serial	Serial port driver c:\winnt\system32\drivers\serial.sys	Kernel Driver	True	Running	OK
sfloppy	Sfloppy c:\winnt\system32\drivers\sfloppy.sys	Kernel Driver	False	Stopped	OK
sglfb	sglfb Not Available	Kernel Driver	False	System	False
simbad	Simbad Not Available	Kernel Driver	False	Disabled	False
sparrow	Sparrow Not Available	Kernel Driver	False	Disabled	False
spud	Special Purpose Utility Driver c:\winnt\system32\drivers\spud.sys	Kernel Driver	False	System	False

srv Driver	Srv c:\winnt\system32\drivers\svr.sys	File System	False	Manual	Stopped	OK	Normal	False
swenum	Software Bus Driver c:\winnt\system32\drivers\swenum.sys	Kernel Driver	True	Manual	Running	OK	Normal	False
symc810	symc810 Not Available	Kernel Driver	False	Manual	Stopped	OK	Normal	False
symc8xx	symc8xx Not Available	Kernel Driver	False	Manual	Stopped	OK	Normal	False
sym_hi	sym_hi Not Available	Kernel Driver	False	Manual	Stopped	OK	Normal	False
tcpip	TCP/IP Protocol Driver c:\winnt\system32\drivers\tcpip.sys	Kernel Driver	True	System	Running	OK	Normal	False
tdasync	TDASYNC c:\winnt\system32\drivers\tdasync.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	Ignore
tdipx	TDIPX c:\winnt\system32\drivers\tdipx.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	Ignore
tdnetb	TDNETB c:\winnt\system32\drivers\tdnetb.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	Ignore
tdpipe	TDPIPE c:\winnt\system32\drivers\tdpipe.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	Ignore
tdspix	TDSPX c:\winnt\system32\drivers\tdspix.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	Ignore
tdtcp	TDTCP c:\winnt\system32\drivers\tdtcp.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	Ignore
termdd	Terminal Device Driver c:\winnt\system32\drivers\termdd.sys	Kernel Driver	True	Auto	Running	OK	Normal	False
tga	tga Not Available	Kernel Driver	False	System	Stopped	OK	Normal	False
udfs Driver	Udfs c:\winnt\system32\drivers\udfs.sys	File System	False	Disabled	Stopped	OK	Normal	Normal
ultra66	ultra66 Not Available	Kernel Driver	False	System	Stopped	OK	Normal	False
update	Microcode Update Driver c:\winnt\system32\drivers\update.sys	Kernel Driver	True	Manual	Running	OK	Normal	False
vgasave	VgaSave c:\winnt\system32\drivers\vgasave.sys	Kernel Driver	True	System	Running	OK	Normal	Ignore
wanarp	Remote Access IP ARP Driver c:\winnt\system32\drivers\wanarp.sys	Kernel Driver	True	Manual	Running	OK	Normal	False
wdica	WDICA Not Available	Kernel Driver	False	Manual	Stopped	OK	Ignore	False

[Environment Variables]

Variable	Value	User Name
ComSpec	%SystemRoot%\system32\cmd.exe	<SYSTEM>
NUMBER_OF_PROCESSORS	4	<SYSTEM>
OS	Windows_NT	<SYSTEM>
Os2LibPath	%SystemRoot%\system32\os2dll\	<SYSTEM>
Path	%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\Program Files\Microsoft SQL Server\MSSQL\BINN;c:\mkscnt;C:\Program Files\Microsoft SQL Server\80\Tools\BINN	<SYSTEM>
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH	<SYSTEM>

PROCESSOR_ARCHITECTURE x86 <SYSTEM>
 PROCESSOR_IDENTIFIER x86 Family 6 Model 10 Stepping 0,
 GenuineIntel <SYSTEM>
 PROCESSOR_LEVEL 6 <SYSTEM>
 PROCESSOR_REVISION 0a00 <SYSTEM>
 TEMP %SystemRoot%\TEMP <SYSTEM>
 TMP %SystemRoot%\TEMP <SYSTEM>
 windir %SystemRoot%\ <SYSTEM>
 TEMP %USERPROFILE%\Local Settings\Temp
 TPC02\Administrator
 TMP %USERPROFILE%\Local Settings\Temp
 TPC02\Administrator

[Network Connections]
 Local Name Remote Name Type Status User Name
 No network connections information

[Running Tasks]

Name	Path	Process ID	Priority	Min Working Set	Status
	Max Working Set	Start Time	Version	Size	
system idle	process	Not Available	0	0	Not Available
system	Not Available	8	0	1413120	Unknown
smss.exe	c:\winnt\system32\smss.exe	204800	1413120	6/6/2000 7:18:23 PM	5.00.2170.1
csrss.exe	Not Available	192	13	Not Available	Not Available
winlogon.exe	c:\winnt\system32\winlogon.exe	204800	1413120	6/6/2000 7:18:32 PM	5.00.2182.1
services.exe	c:\winnt\system32\services.exe	204800	1413120	6/6/2000 7:18:34 PM	5.00.2134.1
lsass.exe	c:\winnt\system32\lsass.exe	204800	1413120	6/6/2000 7:18:34 PM	5.00.2184.1
svchost.exe	c:\winnt\system32\svchost.exe	204800	1413120	6/6/2000 7:18:37 PM	5.00.2134.1
locator.exe	c:\winnt\system32\locator.exe	204800	1413120	6/6/2000 7:18:43 PM	5.00.2135.1
svchost.exe	c:\winnt\system32\svchost.exe	204800	1413120	6/6/2000 7:18:51 PM	5.00.2134.1
explorer.exe	c:\winnt\explorer.exe	204800	1413120	6/6/2000 7:20:05 PM	5.00.2920.0000
sqlmangr.exe	c:\program files\common files\microsoft shared\service manager\sqlmangr.exe	620	8	204800	2000.080.0100.00
cmd.exe	c:\winnt\system32\cmd.exe	204800	1413120	6/6/2000 7:21:30 PM	5.00.2144.1
cmd.exe	c:\winnt\system32\cmd.exe	204800	1413120	6/6/2000 7:22:22 PM	5.00.2144.1
mmc.exe	c:\winnt\system32\mmc.exe	204800	1413120	6/6/2000 7:36:35 PM	5.00.2153.1
winmgmt.exe	c:\winnt\system32\wbem\winmgmt.exe	8	204800	1413120	6/6/2000 7:36:36 PM
		1.50.1085.0001	188.05 KB	(192,567 bytes)	12/7/1999 9:00:00 AM

[Loaded Modules]	Name	Version	Size	File Date	ManufacturerPath
	provthrd.dll	1.50.1085.0000	68.07 KB (69,708 bytes)	2/16/2000	
	ntevt.dll	1.50.1085.0000	192.06 KB (196,669 bytes)		
	psapi.dll	5.00.2134.1	28.27 KB (28,944 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
	framedyn.dll	1.50.1085.0000	164.05 KB (167,992 bytes)		
	cimwin32.dll	1.50.1085.0000	1.03 MB (1,077,306 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
	wbemess.dll	1.50.1085.0001	352.05 KB (360,503 bytes)		
	wbemcore.dll	1.50.1085.0001	632.05 KB (647,224 bytes)		
	winmgmt.exe	1.50.1085.0001	188.05 KB (192,567 bytes)		
	fastprox.dll	1.50.1085.0001	144.08 KB (147,534 bytes)		
	wbemsvcdll.dll	1.50.1085.0000	140.07 KB (143,430 bytes)		
	wbemcomn.dll	1.50.1085.0001	684.05 KB (700,472 bytes)		
	wbemprox.dll	1.50.1085.0001	40.05 KB (41,016 bytes)		
	mlang.dll	5.00.2920.0000	510.77 KB (523,024 bytes)		
	cabinet.dll	5.00.2147.1	54.77 KB (56,080 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
	msinfo32.dll	5.00.2177.1	312.27 KB (319,760 bytes)	2/16/2000 4:04:05 PM	Microsoft Corporation
	mmcndmgr.dll	5.00.2178.1	815.27 KB (834,832 bytes)		
	msvc50.dll	5.00.7051	552.50 KB (565,760 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
	mfc42u.dll	6.00.8665.0	972.05 KB (995,384 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
	cmd.exe	5.00.2144.1	230.77 KB (236,304 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
	clusapi.dll	5.00.2179.1	50.27 KB (51,472 bytes)	4/27/2000 10:04:11 AM	Microsoft Corporation
	odbcint.dll	3.520.5215.0	88.00 KB (90,112 bytes)	4/27/2000 10:04:11 AM	Microsoft Corporation
	cmdlg32.dll	5.00.2920.0000	236.77 KB (242,448 bytes)		
	odbc32.dll	3.520.5215.0	216.27 KB (221,456 bytes)	4/27/2000 10:04:09 AM	Microsoft Corporation
	w95scm.dll	2000.080.0100.00	48.06 KB (49,216 bytes)	5/9/2000 8:26:06 PM	Microsoft Corporation
	shared\service manager\w95scm.dll				

sqlmangr.exe	5/9/2000 8:26:06 PM	2000.080.0100.00	60.00 KB (61,440 bytes)	Microsoft Corporation	c:\program files\common files\microsoft shared\service manager\sqlmangr.exe
faxshell.dll	5.00.2134.1	8.27 KB (8,464 bytes)		Microsoft Corporation	c:\winnt\system32\faxshell.dll
msacm32.dll	5.00.2134.1	65.27 KB (66,832 bytes)		Microsoft Corporation	c:\winnt\system32\msacm32.dll
avifil32.dll	5.00.2134.1	76.27 KB (78,096 bytes)		Microsoft Corporation	c:\winnt\system32\avifil32.dll
msvfw32.dll	5.00.2134.1	113.77 KB (116,496 bytes)		Microsoft Corporation	c:\winnt\system32\msvfw32.dll
docprop2.dll	5.00.2178.1	297.77 KB (304,912 bytes)		Microsoft Corporation	c:\winnt\system32\docprop2.dll
wininet.dll	5.00.2920.0000			Microsoft Corporation	c:\winnt\system32\wininet.dll
urlmon.dll	5.00.2920.0000			Microsoft Corporation	c:\winnt\system32\urlmon.dll
browseui.dll	5.00.2920.0000			Microsoft Corporation	c:\winnt\system32\browseui.dll
linkinfo.dll	5.00.2134.1	15.77 KB (16,144 bytes)		Microsoft Corporation	c:\winnt\system32\linkinfo.dll
cfgmgr32.dll	5.00.2134.1	16.77 KB (17,168 bytes)		Microsoft Corporation	c:\winnt\system32\cfgmgr32.dll
msi.dll	1.10.1029.0	1.71 MB (1,794,320 bytes)		Microsoft Corporation	c:\winnt\system32\msi.dll
powrprof.dll	5.00.2920.0000			Microsoft Corporation	c:\winnt\system32\powrprof.dll
batmeter.dll	5.00.2920.0000			Microsoft Corporation	c:\winnt\system32\batmeter.dll
stobject.dll	5.00.2144.1	81.77 KB (83,728 bytes)		Microsoft Corporation	c:\winnt\system32\stobject.dll
webcheck.dll	5.00.2920.0000			Microsoft Corporation	c:\winnt\system32\webcheck.dll
netui1.dll	5.00.2134.1	210.27 KB (215,312 bytes)		Microsoft Corporation	c:\winnt\system32\netui1.dll
netui0.dll	5.00.2134.1	70.27 KB (71,952 bytes)		Microsoft Corporation	c:\winnt\system32\netui0.dll
ntlanman.dll	5.00.2157.1	35.27 KB (36,112 bytes)		Microsoft Corporation	c:\winnt\system32\ntlanman.dll
ntshrui.dll	5.00.2134.1	46.77 KB (47,888 bytes)		Microsoft Corporation	c:\winnt\system32\ntshrui.dll
mydocs.dll	5.00.2920.0000			Microsoft Corporation	c:\winnt\system32\mydocs.dll
browseui.dll	5.00.2920.0000			Microsoft Corporation	c:\winnt\system32\browseui.dll
shdocvw.dll	5.00.2920.0000			Microsoft Corporation	c:\winnt\system32\shdocvw.dll
explorer.exe	5.00.2920.0000			Microsoft Corporation	c:\winnt\explorer.exe
rasdlg.dll	5.00.2194.1	514.27 KB (526,608 bytes)		Microsoft Corporation	c:\winnt\system32\rasdlg.dll
netcfgx.dll	5.00.2175.1	533.77 KB (546,576 bytes)		Microsoft Corporation	c:\winnt\system32\netcfgx.dll
sens.dll	5.00.2163.1	36.77 KB (37,648 bytes)		Microsoft Corporation	c:\winnt\system32\sens.dll
rasmans.dll	5.00.2188.1	146.77 KB (150,288 bytes)		Microsoft Corporation	c:\winnt\system32\rasmans.dll
wmi.dll	5.00.2191.1	6.27 KB (6,416 bytes)		Microsoft Corporation	c:\winnt\system32\wmi.dll
netshell.dll	5.00.2176.1	456.77 KB (467,728 bytes)		Microsoft Corporation	c:\winnt\system32\netshell.dll
netman.dll	5.00.2175.1	88.77 KB (90,896 bytes)		Microsoft Corporation	c:\winnt\system32\netman.dll
txfaux.dll	1999.9.3422.24			Microsoft Corporation	c:\winnt\system32\txfaux.dll

es.dll	1999.9.3422.00	231.77 KB (237,328 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
locator.exe	5.00.2135.1	70.77 KB (72,464 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
rpcss.dll	5.00.2181.1	229.27 KB (234,768 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
svchost.exe	5.00.2134.1	7.77 KB (7,952 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
scecli.dll	5.00.2191.1	105.27 KB (107,792 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
atl.dll	3.00.8449	57.56 KB (58,938 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
certcli.dll	5.00.2175.1	132.27 KB (135,440 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
ntdsatq.dll	5.00.2181.1	31.27 KB (32,016 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
ntdsd.dll	5.00.2195.1	993.27 KB (1,017,104 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
kdcsvc.dll	5.00.2181.1	133.77 KB (136,976 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
sfmapi.dll	5.00.2134.1	38.77 KB (39,696 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
rasfm.dll	5.00.2168.1	21.27 KB (21,776 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
mpr.dll	5.00.2146.1	53.27 KB (54,544 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
schannel.dll	5.00.2170.1	139.77 KB (143,120 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
netlogon.dll	5.00.2182.1	347.77 KB (356,112 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
kerberos.dll	5.00.2181.1	196.77 KB (201,488 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
msprvs.dll	5.00.2154.1	41.50 KB (42,496 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
samsrv.dll	5.00.2192.1	357.77 KB (366,352 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
lsasrv.dll	5.00.2184.1	487.77 KB (499,472 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
lsass.exe	5.00.2184.1	32.77 KB (33,552 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
esent.dll	6.0.3939.6	1.07 MB (1,120,016 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
wmicore.dll	5.00.2178.1	70.77 KB (72,464 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
psbase.dll	5.00.2146.1	111.77 KB (114,448 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
cryptsvc.dll	5.00.2181.1	61.77 KB (63,248 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
cryptdll.dll	5.00.2135.1	41.27 KB (42,256 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
wkssvc.dll	5.00.2181.1	95.27 KB (97,552 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
eventlog.dll	5.00.2178.1	43.77 KB (44,816 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
ntdsapi.dll	5.00.2160.1	56.27 KB (57,616 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
scsvr.dll	5.00.2188.1	225.77 KB (231,184 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
umpnprg.dll	5.00.2182.1	86.27 KB (88,336 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
services.exe	5.00.2134.1	86.77 KB (88,848 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
msv1_0.dll	5.00.2164.1	94.77 KB (97,040 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
cscli.dll	5.00.2172.1	227.27 KB (232,720 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
sclogntfy.dll	5.00.2191.1	20.27 KB (20,752 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
cryptnet.dll	5.131.2157.1	41.77 KB (42,768 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation

rasadhlp.dll	5.00.2168.1	7.27 KB (7,440 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
wshnetb.dll	5.00.2134.1	7.77 KB (7,952 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
winnr.dll	5.00.2160.1	18.77 KB (19,216 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
clbcatq.dll	1999.9.3422.14	479.27 KB (490,768 bytes)	2/17/2000 12:57:47 AM	Microsoft Corporation
dhcpcsvc.dll	5.00.2153.1	88.77 KB (90,896 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
tapi32.dll	5.00.2182.1	123.27 KB (126,224 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
rasman.dll	5.00.2188.1	54.77 KB (56,080 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
rasapi32.dll	5.00.2188.1	189.77 KB (194,320 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
rtutils.dll	5.00.2168.1	43.77 KB (44,816 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
adslrpc.dll	5.00.2172.1	127.77 KB (130,832 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
activeds.dll	5.00.2172.1	172.77 KB (176,912 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
oleaut32.dll	2.40.4512	600.27 KB (614,672 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
mprapi.dll	5.00.2181.1	79.27 KB (81,168 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
icmp.dll	5.00.2134.1	7.27 KB (7,440 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
iphlpapi.dll	5.00.2173.2	67.77 KB (69,392 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
mr20.dll	5.00.2152.1	35.77 KB (36,624 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
wshtcpip.dll	5.00.2134.1	17.27 KB (17,680 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
msafd.dll	5.00.2153.1	54.27 KB (55,568 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
mswsock.dll	5.00.2152.1	62.27 KB (63,760 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
winspool.drv	5.00.2167.1	109.77 KB (112,400 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
winscard.dll	5.00.2134.1	77.27 KB (79,120 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
wlnotify.dll	5.00.2164.1	53.27 KB (54,544 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
lz32.dll	5.00.2134.1	9.77 KB (10,000 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
version.dll	5.00.2134.1	15.77 KB (16,144 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
cscdll.dll	5.00.2189.1	98.27 KB (100,624 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
rsabase.dll	5.00.2150.1	128.77 KB (131,856 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
mscat32.dll	5.131.2134.1	7.77 KB (7,952 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
ole32.dll	5.00.2181.1	966.27 KB (989,456 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
imagehlp.dll	5.00.2195.1	125.27 KB (128,272 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
msasn1.dll	5.00.2134.1	51.27 KB (52,496 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
crypt32.dll	5.131.2173.1	465.77 KB (476,944 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
wintrust.dll	5.131.2143.1	162.27 KB (166,160 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
setupapi.dll	5.00.2183.1	554.27 KB (567,568 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
winnm.dll	5.00.2161.1	184.77 KB (189,200 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
comctl32.dll	5.81	540.27 KB (553,232 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation

shlwapi.dll	5.00.2920.0000	282.77 KB (289,552 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
shell32.dll	5.00.2920.0000	2.24 MB (2,352,400 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
msgina.dll	5.00.2191.1	309.77 KB (317,200 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
winsta.dll	5.00.2134.1	36.27 KB (37,136 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
wsock32.dll	5.00.2152.1	21.27 KB (21,776 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
dnsapi.dll	5.00.2181.1	129.77 KB (132,880 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
wldap32.dll	5.00.2168.1	155.77 KB (159,504 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
ws2help.dll	5.00.2134.1	17.77 KB (18,192 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
ws2_32.dll	5.00.2134.1	69.77 KB (71,440 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
samiib.dll	5.00.2160.1	46.27 KB (47,376 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
netrap.dll	5.00.2134.1	11.27 KB (11,536 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
netapi32.dll	5.00.2194.1	302.77 KB (310,032 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
profmap.dll	5.00.2181.1	29.27 KB (29,968 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
secur32.dll	5.00.2154.1	46.77 KB (47,888 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
sfc.dll	5.00.2164.1	84.27 KB (86,288 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
nddeapi.dll	5.00.2137.1	15.27 KB (15,632 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
userenv.dll	5.00.2185.1	361.27 KB (369,936 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
user32.dll	5.00.2180.1	393.27 KB (402,704 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
gdisk32.dll	5.00.2180.1	228.77 KB (234,256 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
rpcrt4.dll	5.00.2193.1	434.27 KB (444,688 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
advapi32.dll	5.00.2191.1	349.27 KB (357,648 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
kernel32.dll	5.00.2191.1	715.27 KB (732,432 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
msvcrt.dll	6.10.8637.0	288.09 KB (295,000 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
winlogon.exe	5.00.2182.1	173.27 KB (177,424 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
sfcdll.dll	5.00.2195.1	973.27 KB (996,624 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
ntdll.dll	5.00.2163.1	469.77 KB (481,040 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation
smss.exe	5.00.2170.1	44.27 KB (45,328 bytes)	12/7/1999 9:00:00 AM	Microsoft Corporation

[Services]					
Display Name	Path	Name	State	Start Mode	Service Type
Alerter	c:\winnt\system32\alerter	Alerter	Stopped	Disabled	Share Process
Application Management	c:\winnt\system32\services.exe	services.exe	Stopped	Normal	LocalSystem
Computer Browser	c:\winnt\system32\services.exe	services.exe	Stopped	Disabled	Share
Indexing Service	c:\winnt\system32\cisvc.exe	cisvc	Stopped	Manual	Share

ClipBook	ClipSrv	Stopped	Manual	Own Process	Normal	LocalSystem
	c:\winnt\system32\clipsrv.exe					
Distributed Process	File System	Dfs	Stopped	Disabled	Own	LocalSystem
	c:\winnt\system32\dfsrv.exe					
DHCP Client	Dhcp	Stopped	Disabled	Share Process	Normal	LocalSystem
	c:\winnt\system32\services.exe					
Logical Disk Manager	Administrative Service			dmadmin	Stopped	
	Manual Share Process					
	c:\winnt\system32\dmadmin.exe /com			Normal		LocalSystem
Logical Disk Process	Manager	dmserver	Stopped	Manual	Share	LocalSystem
	c:\winnt\system32\services.exe			Normal		
DNS Client	Dnscache	Stopped	Disabled	Share Process	Normal	LocalSystem
	c:\winnt\system32\services.exe					
Event Log	Eventlog	Running	Auto	Share Process	Normal	LocalSystem
	c:\winnt\system32\services.exe					
COM+ Event Process	System	EventSystemRunning		Manual	Share	LocalSystem
	c:\winnt\system32\svchost.exe -k netsvcs			Normal		
Fax Service	Fax	Stopped	Manual	Own Process	Normal	LocalSystem
	c:\winnt\system32\faxsvc.exe					
IIS Admin Service	IISADMIN	Stopped		Disabled	Share	LocalSystem
	c:\winnt\system32\inetrv\inetinfo.exe			Normal		
Intersite Messaging Process	Ismserv	Stopped		Disabled	Own	LocalSystem
	c:\winnt\system32\ismserv.exe			Normal		
Kerberos Key Distribution Center		kdc	Stopped	Disabled	Share Process	LocalSystem
	Normal LocalSystem					
Server Process	lanmanserver	Stopped		Manual	Share	LocalSystem
	c:\winnt\system32\services.exe			Normal		
Workstation Process	lanmanworkstation	Running		Auto	Share	LocalSystem
	c:\winnt\system32\services.exe			Normal		
License Logging Service	LicenseService		Stopped	Disabled	Own Process	LocalSystem
	Normal LocalSystem					
TCP/IP NetBIOS Helper Service	LmHosts	Stopped		Manual	Share Process	LocalSystem
	Normal LocalSystem					
Messenger	Messenger	Stopped	Manual	Share Process	Normal	LocalSystem
	c:\winnt\system32\services.exe					
NetMeeting	Remote Desktop Sharing	mnmsrv	Stopped	Manual	Own Process	LocalSystem
	Normal LocalSystem					
Distributed Transaction Coordinator	MSDTC	Stopped	Disabled	Share Process	Normal	LocalSystem
	c:\winnt\system32\msdct.exe					
Windows Installer	MSIServer	Stopped	Manual	Share	Normal	LocalSystem
	c:\winnt\system32\msiexec.exe /v					
Microsoft Search Process	MSSEARCH	Stopped	Disabled	Share	Normal	LocalSystem
	"c:\program files\common files\system\search\bin\mssearch.exe"					
MSSQLSERVER	MSSQLSERVER	Stopped	Manual	Own Process	Normal	LocalSystem
	c:\progra~1\microso~3\mssql\bin\sqlservr.exe					
Network DDE Process	NetDDE	Stopped	Disabled	Share	Normal	LocalSystem
	c:\winnt\system32\netdde.exe					

Network DDE DSDM	NetDDEdsdm	Stopped	Disabled	Share Process	Normal	LocalSystem
	c:\winnt\system32\netdde.exe					
Net Logon	Netlogon	Stopped	Manual	Share Process	Normal	LocalSystem
	c:\winnt\system32\lsass.exe					
Network Connections Process	Netman	Running	Manual	Share	Normal	LocalSystem
	c:\winnt\system32\svchost.exe -k netsvcs					
File Replication Process	NtFrs	Stopped	Manual	Own	Ignore	LocalSystem
	c:\winnt\system32\ntfrs.exe					
NT LM Security Support Provider	NtLmSsp	Stopped	Manual	Share Process	Normal	LocalSystem
	c:\winnt\system32\lsass.exe					
Removable Storage Process	NtmsSvc	Stopped	Disabled	Share	Normal	LocalSystem
	c:\winnt\system32\svchost.exe -k netsvcs					
Plug and Play Process	PlugPlay	Running	Auto	Share	Normal	LocalSystem
	c:\winnt\system32\services.exe					
IPSEC Policy Agent	PolicyAgent	Stopped	Disabled	Share	Normal	LocalSystem
	c:\winnt\system32\lsass.exe					
Protected Storage	ProtectedStorage	Running	Auto	Share Process	Normal	LocalSystem
	c:\winnt\system32\services.exe					
Remote Access Auto Connection Manager	RasAuto	Stopped		Disabled	Share Process	LocalSystem
	c:\winnt\system32\svchost.exe -k netsvcs					
Remote Access Connection Manager	RasMan	Stopped	Disabled	Share Process	Normal	LocalSystem
	c:\winnt\system32\svchost.exe -k netsvcs					
Routing and Remote Access	RemoteAccess	Running	Auto	Disabled	Share Process	LocalSystem
	c:\winnt\system32\svchost.exe -k netsvcs					
Remote Registry Service	RemoteRegistry	Stopped	Disabled	Own Process	Normal	LocalSystem
	c:\winnt\system32\regsvc.exe					
Remote Procedure Call (RPC) Locator	RpcLocator	Running	Auto	Own Process	Normal	LocalSystem
	c:\winnt\system32\locator.exe					
Remote Procedure Call (RPC)	RpcSs	Running	Auto	Share Process	Normal	LocalSystem
	c:\winnt\system32\svchost.exe -k rpcss					
QoS RSVP	RSVP	Stopped	Manual	Own Process	Normal	LocalSystem
	c:\winnt\system32\rsvp.exe -s					
Security Accounts Manager	SamSs	Stopped	Manual	Share Process	Normal	LocalSystem
	c:\winnt\system32\lsass.exe					
Smart Card Helper	SCardDrv	Stopped	Manual	Ignore	Share	LocalSystem
	c:\winnt\system32\scardsvr.exe					
Smart Card	SCardSrv	Stopped	Manual	Share Process	Ignore	LocalSystem
	c:\winnt\system32\scardsvr.exe					
Task Scheduler	Schedule	Stopped	Disabled	Share	Normal	LocalSystem
	c:\winnt\system32\mtask.exe					
RunAs Service	seclogon	Stopped	Manual	Share	Ignore	LocalSystem
	c:\winnt\system32\services.exe					
System Event Notification	SENS	Stopped	Manual	Share	Normal	LocalSystem
	c:\winnt\system32\svchost.exe -k netsvcs					
Internet Connection Sharing	SharedAccess		Stopped	Manual	Share Process	LocalSystem
	c:\winnt\system32\svchost.exe -k netsvcs					

Simple Mail Transport Protocol (SMTP)	SMTPSVC	Stopped	Manual	Share Process	Normal	LocalSystem
	c:\winnt\system32\inetrv\inetinfo.exe					
Print Spooler	Spooler	Stopped	Disabled	Own Process	Normal	LocalSystem
	c:\winnt\system32\spoolsv.exe					
SQLSERVERAGENT	SQLSERVERAGENT	Stopped	Manual	Own Process	Normal	LocalSystem
	c:\progra~1\microso~3\mssql\bin\sqlagent.exe					
Performance Logs and Alerts	SysmonLog	Stopped	Manual	Own Process	Normal	LocalSystem
	c:\winnt\system32\smlogsvc.exe					
Telephony	TapiSrv	Stopped	Disabled	Share Process	Normal	LocalSystem
	c:\winnt\system32\svchost.exe -k tapisrv					
Terminal Services Process	TermService	Stopped	Manual	Own	Normal	LocalSystem
	c:\winnt\system32\termsrv.exe					
Terminal Services Licensing	TermServLicensing	Stopped		Disabled	Own Process	LocalSystem
	c:\winnt\system32\server.exe					
Telnet	TintSvr	Stopped	Manual	Own Process	Normal	LocalSystem
	c:\winnt\system32\tintsvr.exe					
Distributed Link Tracking Server	TrkSvr	Stopped	Manual	Share Process	Normal	LocalSystem
	c:\winnt\system32\services.exe					
Distributed Link Tracking Client	TrkWks	Stopped	Disabled	Share Process	Normal	LocalSystem
	c:\winnt\system32\services.exe					
Uninterruptible Power Supply	UPS	Stopped	Manual	Own Process	Normal	LocalSystem
	c:\winnt\system32\ups.exe					
Utility Manager Process	UtilMan	Stopped	Manual	Own	Normal	LocalSystem
	c:\winnt\system32\utilman.exe					
Windows Time Process	W32Time	Stopped	Manual	Share	Normal	LocalSystem
	c:\winnt\system32\services.exe					
World Wide Web Publishing Service	W3SVC	Stopped	Manual	Share Process	Normal	LocalSystem
	c:\winnt\system32\inetrv\inetinfo.exe					
Windows Management Instrumentation	WinMgmt	Running	Manual	Own Process	Ignore	LocalSystem
	c:\winnt\system32\wbem\winmgmt.exe					
Windows Management Instrumentation Driver Extensions	Wmi	Running	Manual	Share Process	Normal	LocalSystem
	c:\winnt\system32\services.exe					
[Program Groups]						
Group Name	User Name					
Accessories	Default User:Accessories	Default User				
Accessories\Accessibility	Default User:Accessories\Accessibility	Default User				
Accessories\Entertainment	Default User:Accessories\Entertainment	Default User				
Accessories\System Tools	Default User:Accessories\System Tools	Default User				
Accessories\All Users:Accessories	All Users					
Accessories\Accessibility	All Users:Accessories\Accessibility	All Users				
Accessories\Communications	All Users:Accessories\Communications	All Users				
Accessories\Communications\Fax	All Users:Accessories\Communications\Fax	All Users				
Accessories\Entertainment	All Users:Accessories\Entertainment	All Users				
Accessories\Games	All Users:Accessories\Games	All Users				
Accessories\Microsoft Script Debugger	All Users:Accessories\Microsoft Script Debugger	All Users				
Accessories\System Tools	All Users:Accessories\System Tools	All Users				

```

Administrative Tools All Users:Administrative Tools All Users
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 TPC02\Administrator:Accessories TPC02\Administrator
Accessories\Accessibility TPC02\Administrator:Accessories\Accessibility
TPC02\Administrator
Accessories\Communications
TPC02\Administrator:Accessories\Communications
TPC02\Administrator
Accessories\Communications\HyperTerminal
TPC02\Administrator:Accessories\Communications\HyperTerminal
TPC02\Administrator
Accessories\Entertainment
TPC02\Administrator:Accessories\Entertainment
TPC02\Administrator
Accessories\System Tools TPC02\Administrator:Accessories\System Tools
TPC02\Administrator
Administrative Tools TPC02\Administrator:Administrative Tools
TPC02\Administrator
Startup TPC02\Administrator:Startup TPC02\Administrator

[Startup Programs]
Program Command User Name Location
Service Manager
c:\progra~1\common~1\micro~1\servic~1\sqlmangr.exe /n
All Users Common Startup
mdac_runonce c:\winnt\system32\runonce.exe All Users
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

```

<Client Configuration>

COM+ application Configuration

TPCC.ALLTxns properties

Activation:

```

Enable object pooling
  Minimum pool size 42
  Maximum pool size 42
  Creation timeout[ms] 60000
Enable object construction
Enable Just In Time Activation

```

Concurrency:

Concurrency Required

InetInfo Registry

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00001a90
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00,0,00,53,00,\
4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00,00
"PoolThreadLimit"=dword:00000190

```

```

"ThreadTimeout"=dword:00015180
"DisableMemoryCache"=dword:00000001
"BandwidthLevel"=dword:ffffff
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffff

```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000842
"Last Help"=dword:00000843
"First Counter"=dword:00000802
"First Help"=dword:00000803
"WbemAdapStatus"=dword:00000000
"FileTime"=hex:f8,eb,d8,13,25,cb,bf,01
"FileSize"=dword:00002510

```

TPC-C application registry

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPCC]
"Path"="c:\inetpub\wwwroot\I"
"NumberOfDeliveryThreads"=dword:00000004
"MaxConnections"=dword:00001b58
"MaxPendingDeliveries"=dword:00000328
"DB_Protocol"="DBLIB"
"TxnMonitor"="COM"
"DbServer"="tpc02"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"=""
"COM_SinglePool"="YES"

```

WWW Service Registry

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,00,3a,00,5c,00,57,00,49,00,4e,00,4e,00,54,00,5c,00,53,00,\
79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00,6e,00,65,00,67,00,4,00,73,\
00,72,00,76,00,5c,00,69,00,6e,00,65,00,74,00,69,00,6e,00,66,00,6f,00,2e,00,\
65,00,78,00,65,00,00,00
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):49,00,49,00,53,00,41,00,44,00,4d,00,49,0,4e,00,00,00,\
00,00
"DependOnGroup"=hex(7):00,00
"ObjectName"="LocalSystem"
"Description"="Provides Web connectivity and administration through the Internet Information Services snap-in."

```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP]
"NOTE"="This is for backward compatibility only."

```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP\Parameters]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters]
"MajorVersion"=dword:00000005
"MinorVersion"=dword:00000000
"InstallPath"="C:\WINNT\System32\inet\iis\iisrmap.dll"
"CertMapList"="C:\WINNT\System32\inet\iisrmap.dll"
"AccessDeniedMessage"="Error: Access is Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\WINNT\System32\LogFiles"
"AcceptExOutstanding"=dword:00001a2c

```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots]
"/"="c:\inetpub\wwwroot,205"
"/Scripts"="c:\inetpub\scripts,204"
"/IISAdmin"="C:\WINNT\System32\inet\iisadmin,201"
"/IISSamples"="c:\inetpub\iissamples,201"
"/MSADC"="c:\program files\common files\system\msadc,205"
"/IISHelp"="c:\winnt\help\iishelp,201"
"/_vti_bin"="C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\40\isapi,205"
"/Printers"="C:\WINNT\web\printers,201"

```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Performance]
"Library"="w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"Last Counter"=dword:000008f2
"Last Help"=dword:000008f3
"First Counter"=dword:00000850
"First Help"=dword:00000851
"WbemAdapStatus"=dword:00000000
"FileTime"=hex:4e,67,be,16,25,cb,bf,01
"FileSize"=dword:00003d10

```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,02,\
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,

```



```

00,00,\
00,00,02,00,70,00,04,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,0
0,00,00,\
05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,0,0
0,05,\
20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01,02,00,01,01,0
0,00,00,\
00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,0
0,00,00,\
00,05,20,00,00,00,23,02,00,00,72,00,73,00,01,01,00,00,00,00,05,1
2,00,00,\
00,01,01,00,00,00,00,00,05,12,00,00,00

```

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Enum]
"0"="Root\LEGACY_W3SVC\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

```

Microsoft Windows 2000 Server Configuration Parameters

[system Information]

Item	Value
OS Name	Microsoft Windows 2000 Server
Version	5.0.2128 Build 2128
OS Manufacturer	Microsoft Corporation
System Name	ACL01
System Manufacturer	NEC
System Model	Express5800/120Ld
System Type	X86-based PC
Processor	x86 Family 6 Model 8 Stepping 3 GenuineIntel ~866 Mhz
Processor	x86 Family 6 Model 8 Stepping 3 GenuineIntel ~866 Mhz
BIOS Version	PhoenixBIOS 4.0 Release 6.0.2538
Windows Directory	C:\WINNT
Locale	United States
Total Physical Memory	523,756 KB
Available Physical Memory	422,988 KB
Total Virtual Memory	1,802,076 KB
Available Virtual Memory	1,634,084 KB
Page File Space	1,278,320 KB

[Software Environment]

[Drivers]

Name	Description	Type	State	Status
abiosdsk	Abiosdsk	Kernel Driver	Stopped	OK
abp480n5	abp480n5	Kernel Driver	Stopped	OK

acpi	Microsoft ACPI Driver	Kernel Driver	Running	OK
acpiec	Microsoft Embedded Controller Driver	Kernel Driver	Running	OK
adptsf	Adaptec DuraLAN PCI Ethernet/Fast Ethernet driver for Windows NT	Kernel Driver	Running	OK
adpu160m	adpu160m	Kernel Driver	Running	OK
afd	AFD Networking Support Environment	Kernel Driver	Running	OK
aha154x	Aha154x	Kernel Driver	Stopped	OK
aic116x	aic116x	Kernel Driver	Stopped	OK
aic78u2	aic78u2	Kernel Driver	Stopped	OK
aic78xx	aic78xx	Kernel Driver	Stopped	OK
ami0nt	ami0nt	Kernel Driver	Stopped	OK
amsint	amsint	Kernel Driver	Stopped	OK
arp1394	1394 ARP Client Protocol	Kernel Driver	Stopped	OK
asc	asc	Kernel Driver	Stopped	OK
asc3350p	asc3350p	Kernel Driver	Stopped	OK
asc3550	asc3550	Kernel Driver	Stopped	OK
asynccmac	RAS Asynchronous Media Driver	Kernel Driver	Stopped	OK
atapi	Standard IDE/ESDI Hard Disk Controller	Kernel Driver	Running	OK
atdisk	Atdisk	Kernel Driver	Stopped	OK
atirage	atirage	Kernel Driver	Running	OK
atmarpc	ATM ARP Client Protocol	Kernel Driver	Stopped	OK
audstub	Audio Stub Driver	Kernel Driver	Running	OK
beep	Beep	Kernel Driver	Running	OK
buslogic	BusLogic	Kernel Driver	Stopped	OK
cd20xrnt	cd20xrnt	Kernel Driver	Stopped	OK
cdaudio	Cdaudio	Kernel Driver	Stopped	OK
cdfs	Cdfs	File System Driver	Running	OK
cdrom	CD-ROM Driver	Kernel Driver	Running	OK
changer	Changer	Kernel Driver	Stopped	OK
cpqarray	Cpqarray	Kernel Driver	Stopped	OK
cpqfcalm	cpqfcalm	Kernel Driver	Stopped	OK
cpqfws2e	cpqfws2e	Kernel Driver	Stopped	OK
dac960nt	dac960nt	Kernel Driver	Stopped	OK
deckzpsx	deckzpsx	Kernel Driver	Stopped	OK
dfsdriver	DfsDriver	File System Driver	Running	OK
disk	Disk Driver	Kernel Driver	Running	OK
diskperf	Diskperf	Kernel Driver	Running	OK
dmbboot	dmbboot	Kernel Driver	Stopped	OK
dmio	Logical Disk Manager Driver	Kernel Driver	Running	OK
dmload	dmload	Kernel Driver	Running	OK
e100b	Intel PRO Adapter Driver	Kernel Driver	Running	OK
efs	EFS	File System Driver	Running	OK
fastfat	Fastfat	File System Driver	Running	OK
fd16_700	Fd16_700	Kernel Driver	Stopped	OK
fdc	Floppy Disk Controller Driver	Kernel Driver	Running	OK
fireport	fireport	Kernel Driver	Stopped	OK
flashpnt	flashpnt	Kernel Driver	Stopped	OK
flpydisk	Floppy Disk Driver	Kernel Driver	Running	OK
ftdisk	Volume Manager Driver	Kernel Driver	Running	OK

gpc	Generic Packet Classifier	Kernel Driver	Running	OK
i8042prt	i8042 Keyboard and PS/2 Mouse Port Driver	Kernel Driver	Running	OK
ini910u	ini910u	Kernel Driver	Stopped	OK
intelide	IntelIde	Kernel Driver	Stopped	OK
ipfilterdriver	IP Traffic Filter Driver	Kernel Driver	Stopped	OK
ipinip	IP in IP Tunnel Driver	Kernel Driver	Stopped	OK
ipnat	IP Network Address Translator	Kernel Driver	Stopped	OK
ipsec	IPSEC driver	Kernel Driver	Running	OK
ipsraidn	ipsraidn	Kernel Driver	Stopped	OK
isapnp	PnP ISA/EISA Bus Driver	Kernel Driver	Running	OK
kbdclass	Keyboard Class Driver	Kernel Driver	Running	OK
ksecdd	KSecDD	Kernel Driver	Running	OK
lbrtfdc	lbrtfdc	Kernel Driver	Stopped	OK
lp6nds35	lp6nds35	Kernel Driver	Stopped	OK
mnmdd	mnmdd	Kernel Driver	Running	OK
modem	Modem	Kernel Driver	Stopped	OK
mouclass	Mouse Class Driver	Kernel Driver	Running	OK
mountmgr	MountMgr	Kernel Driver	Running	OK
mraid35x	mraid35x	Kernel Driver	Stopped	OK
mrxsmmb	MRXSMB	File System Driver	Running	OK
msfs	Msfs	File System Driver	Running	OK
mskssrv	Microsoft Streaming Service Proxy	Kernel Driver	Stopped	OK
mspclock	Microsoft Streaming Clock Proxy	Kernel Driver	Stopped	OK
mspqm	Microsoft Streaming Quality Manager Proxy	Kernel Driver	Stopped	OK
mup	Mup	File System Driver	Running	OK
ncrc710	Ncrc710	Kernel Driver	Stopped	OK
ndis	NDIS System Driver	Kernel Driver	Running	OK
ndistapi	Remote Access NDIS TAPI Driver	Kernel Driver	Running	OK
ndiswan	Remote Access NDIS WAN Driver	Kernel Driver	Running	OK
ndproxy	NDIS Proxy	Kernel Driver	Running	OK
netbios	NetBIOS Interface	File System Driver	Running	OK
netbt	NetBios over Tcpip	Kernel Driver	Running	OK
netdetect	NetDetect	Kernel Driver	Stopped	OK
npfs	Npfs	File System Driver	Running	OK
ntfs	Ntfs	File System Driver	Running	OK
null	Null	Kernel Driver	Running	OK
nwlkflt	IPX Traffic Filter Driver	Kernel Driver	Stopped	OK
nwlkfwd	IPX Traffic Forwarder Driver	Kernel Driver	Stopped	OK
parallel	Parallel class driver	Kernel Driver	Running	OK
parport	Parallel port driver	Kernel Driver	Running	OK

```

partmgr PartMgr Kernel Driver Running OK
parvdm ParVdm Kernel Driver Running OK
pci PCI Bus Driver Kernel Driver Running OK
pcidump PCIDump Kernel Driver Stopped OK
pciide PCIIDE Kernel Driver Running OK
pccmca Pccmca Kernel Driver Stopped OK
pdcomp PDCOMP Kernel Driver Stopped OK
pdframe PDFRAME Kernel Driver Stopped OK
pdreli PDRELI Kernel Driver Stopped OK
pdrframe PDRFRAME Kernel Driver Stopped OK
pptpminiport WAN Miniport (PPTP) Kernel Driver Running OK
ptilink Direct Parallel Link Driver Kernel Driver Running OK
ql1080 ql1080 Kernel Driver Stopped OK
ql10wnt Ql10wnt Kernel Driver Stopped OK
ql1240 ql1240 Kernel Driver Stopped OK
ql2100 ql2100 Kernel Driver Stopped OK
rasacd Remote Access Auto Connection Driver Kernel Running OK
rasl2tp WAN Miniport (L2TP) Kernel Driver Running OK
raspti Direct Parallel Kernel Driver Running OK
rca Microsoft Streaming Network Raw Channel Access Kernel Driver Stopped OK
rdcss Rdbss File System Driver Running OK
rdpwd RDPWD Kernel Driver Stopped OK
redbook Digital CD Audio Playback Filter Driver Kernel Stopped OK
serenum Serenum Filter Driver Kernel Driver Running OK
serial Serial port driver Kernel Driver Running OK
sfloppy Sfloppy Kernel Driver Stopped OK
sglfb Sglfb Kernel Driver Stopped OK
simbad Simbad Kernel Driver Stopped OK
sparrow Sparrow Kernel Driver Stopped OK
spud Special Purpose Utility Driver Kernel Driver Running OK
srv Srv File System Driver Running OK
swenum Software Bus Driver Kernel Driver Running OK
symc810 symc810 Kernel Driver Stopped OK
symc8xx symc8xx Kernel Driver Stopped OK
sym_hi sym_hi Kernel Driver Stopped OK
tcpip TCP/IP Protocol Driver Kernel Driver Running OK
tdasync TDASYNC Kernel Driver Stopped OK
tdipx TDIPX Kernel Driver Stopped OK
tdnetb TDNETB Kernel Driver Stopped OK
tdpipe TDPIPE Kernel Driver Stopped OK
tdspx TDSPX Kernel Driver Stopped OK
tdtcp TDTCP Kernel Driver Stopped OK
termdd Terminal Device Driver Kernel Driver Stopped OK
tga tga Kernel Driver Stopped OK
udfs Udfs File System Driver Stopped OK
ultra66 ultra66 Kernel Driver Stopped OK
update Microcode Update Driver Kernel Driver

```

```

vgasave VgaSave Kernel Driver Running OK
wanarp Remote Access IP ARP Driver Kernel Driver Running OK
wdica WDICA Kernel Driver Stopped OK

[Environment Variables]

Variable Value User Name
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
Os2LibPath %SystemRoot%\system32\os2\dll; Path
%SystemRoot%\system32;%SystemRoot%\SystemRoot
%\System32\Wbem;C:\Program Files\Microsoft SQL
Server\80\Tools\BINN <SYSTEM>
windir %SystemRoot% <SYSTEM>
OS Windows_NT <SYSTEM>
PROCESSOR_ARCHITECTURE x86 <SYSTEM>
PROCESSOR_LEVEL6 <SYSTEM>
PROCESSOR_IDENTIFIER x86 Family 6 Model 8 Stepping 3,
GenuineIntel <SYSTEM>
PROCESSOR_REVISION 0803 <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
ACL01\Administrator
TMP %USERPROFILE%\Local Settings\Temp
ACL01\Administrator

[Network Connections]

Local Name Remote Name Type Status
No network connections information

[Running Tasks]

Name Path Version Size File Date
system idle process Not Available Unknown Unknown
system Not Available Unknown Unknown Unknown
smss.exe c:\winnt\system32\smss.exe 5.00.2090.1
42.77 KB (43,792 bytes) Not Available
csrss.exe Not Available Unknown Unknown Unknown
winlogon.exe c:\winnt\system32\winlogon.exe
5.00.2116.1 171.27 KB (175,376 bytes)
Not Available
services.exe c:\winnt\system32\services.exe
5.00.2106.1 87.27 KB (89,360 bytes)
Not Available
lsass.exe c:\winnt\system32\lsass.exe 5.00.2121.1
32.77 KB (33,552 bytes) Not Available
svchost.exe c:\winnt\system32\svchost.exe 5.00.2090.1
7.77 KB (7,952 bytes) Not Available
msdtc.exe c:\winnt\system32\msdtc.exe 1999.8.3413.3
6.77 KB (6,928 bytes) Not Available
svchost.exe c:\winnt\system32\svchost.exe 5.00.2090.1

```

```

7.77 KB (7,952 bytes) Not Available
llssrv.exe c:\winnt\system32\llssrv.exe 5.00.2090.1
113.77 KB (116,496 bytes) Not Available
regsvcs.exe c:\winnt\system32\regsvcs.exe 5.00.2091.1
63.77 KB (65,296 bytes) Not Available
mstask.exe c:\winnt\system32\mstask.exe 4.71.2113.1
114.77 KB (117,520 bytes) Not Available
inetinfo.exe c:\winnt\system32\inet\inetinfo.exe 5.00.0984
14.27 KB (14,608 bytes) Not Available
explorer.exe c:\winnt\explorer.exe 5.00.2919.3800
232.77 KB (238,352 bytes) Not Available
dfssvc.exe c:\winnt\system32\dfssvc.exe 5.00.2124.1
95.77 KB (98,064 bytes) Not Available
dllhost.exe Not Available Unknown Unknown
svchost.exe c:\winnt\system32\svchost.exe 5.00.2090.1
7.77 KB (7,952 bytes) Not Available
cmd.exe c:\winnt\system32\cmd.exe 5.00.2100.1
230.77 KB (236,304 bytes) Not Available
notepad.exe c:\winnt\system32\notepad.exe
5.00.2103.1 49.77 KB (50,960 bytes)
Not Available
mmc.exe c:\winnt\system32\mmc.exe 5.00.2115.1
589.27 KB (603,408 bytes) Not Available
winmgmt.exe c:\winnt\system32\wbem\winmgmt.exe
1.50.1025.0015 164.05 KB (167,991 bytes)
Not Available

[Loaded Modules]

Name Version Size File Date Manufacturer
provthrd.dll 1.50.1025.0001 68.08 KB (69,713 bytes)
5/31/2000 5:29:26 PM Microsoft Corporation
c:\winnt\system32\wbem\provthrd.dll
ntevt.dll 1.50.1025.0002 176.06 KB (180,290 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\ntevt.dll
perfos.dll 5.0 21.27 KB (21,776 bytes) 9/10/1999
9:00:00 AM Microsoft Corporation c:\winnt\system32\perfos.dll
psapi.dll 5.00.2090.1 28.27 KB (28,944 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\psapi.dll
framedyn.dll 1.50.1025.0002 164.05 KB (167,988
bytes) 9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\framedyn.dll
cimwin32.dll 1.50.1025.0016 1.02 MB (1,065,018
bytes) 9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\cimwin32.dll
wbemess.dll 1.50.1025.0009 324.05 KB (331,827
bytes) 9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\wbemess.dll
wbemcore.dll 1.50.1025.0012 592.05 KB (606,260
bytes) 9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\wbemcore.dll
winmgmt.exe 1.50.1025.0015 164.05 KB (167,991
bytes) 9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\winmgmt.exe
fastprox.dll 1.50.1025.0009 144.08 KB (147,536 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\fastprox.dll
wbemsvcs.dll 1.50.1025.0009 136.07 KB (139,339

```

bytes) 9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\wbemsvcs.dll

wbemcomn.dll 1.50.1025.0009 688.05 KB (704,564 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\wbemcomn.dll

wbemprox.dll 1.50.1025.0009 40.05 KB (41,012 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\wbemprox.dll

mlang.dll 5.00.2919.3800 509.77 KB (522,000 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\mlang.dll

cabinet.dll 5.00.2090.1 54.77 KB (56,080 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\cabinet.dll

msinfo32.dll 5.00.2121.1 306.27 KB (313,616 bytes)
5/31/2000 5:29:34 PM Microsoft Corporation c:\program files\common files\microsoft shared\msinfo\msinfo32.dll

mmcndmgr.dll 5.00.2108.1 815.27 KB (834,832 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\mmcndmgr.dll

mmc.exe 5.00.2115.1 589.27 KB (603,408 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\mmc.exe

notepad.exe 5.00.2103.1 49.77 KB (50,960 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\notepad.exe

cmd.exe 5.00.2100.1 230.77 KB (236,304 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\cmd.exe

tapisrv.dll 5.00.2114.1 170.77 KB (174,864 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\tapisrv.dll

dfssvc.exe 5.00.2124.1 95.77 KB (98,064 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\dfssvc.exe

netplwiz.dll 5.00.2116.1 167.27 KB (171,280 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\netplwiz.dll

netmsg.dll 5.00.2090.1 152.50 KB (156,160 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\netmsg.dll

netui2.dll 5.00.2107.1 280.27 KB (286,992 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\netui2.dll

mprui.dll 5.00.2090.1 54.77 KB (56,080 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\mprui.dll

hhsetup.dll 4.74.8576 66.27 KB (67,856 bytes) 9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\hhsetup.dll

mmcshext.dll 5.00.2108.1 24.27 KB (24,848 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\mmcshext.dll

shdoclc.dll 5.00.2919.3800 324.50 KB (332,288 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\shdoclc.dll

netui1.dll 5.00.2107.1 209.77 KB (214,800 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\netui1.dll

netui0.dll 5.00.2107.1 70.27 KB (71,952 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\netui0.dll

ntlanman.dll 5.00.2109.1 35.27 KB (36,112 bytes)

bytes) 9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\ntlanman.dll

urlmon.dll 5.00.2919.3800 426.77 KB (437,008 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\urlmon.dll

faxshell.dll 5.00.2101.1 8.27 KB (8,464 bytes) 9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\faxshell.dll

msacm32.dll 5.00.2113.1 65.27 KB (66,832 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\msacm32.dll

avifil32.dll 5.00.2113.1 76.27 KB (78,096 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\avifil32.dll

msvfw32.dll 5.00.2113.1 113.77 KB (116,496 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\msvfw32.dll

docprop2.dll 5.00.2115.1 297.77 KB (304,912 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\docprop2.dll

browseui.dll 5.00.2919.3800 34.50 KB (35,328 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\browseui.dll

wininet.dll 5.00.2919.3800 456.77 KB (467,728 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\wininet.dll

linkinfo.dll 5.00.2091.1 15.77 KB (16,144 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\linkinfo.dll

powrprof.dll 5.00.2919.3800 13.27 KB (13,584 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\powrprof.dll

batmeter.dll 5.00.2919.3800 20.27 KB (20,752 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\batmeter.dll

stobject.dll 5.00.2120.1 81.27 KB (83,216 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\stobject.dll

webcheck.dll 5.00.2919.3800 251.77 KB (257,808 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\webcheck.dll

ntshrui.dll 5.00.2090.1 46.77 KB (47,888 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\ntshrui.dll

mydocs.dll 5.00.2919.3800 55.77 KB (57,104 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\mydocs.dll

browseui.dll 5.00.2919.3800 791.77 KB (810,768 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\browseui.dll

shdocvw.dll 5.00.2919.3800 1.05 MB (1,103,632 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\shdocvw.dll

explorer.exe 5.00.2919.3800 232.77 KB (238,352 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\explorer.exe

dbghelp.dll 5.00.2128.1 77.27 KB (79,120 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\dbghelp.dll

tpcc_com_ps.dll Not Available 24.00 KB (24,576 bytes)
5/31/2000 6:57:36 PM Not Available
c:\inetpub\wwwroot\tpcc_com_ps.dll

tpcc_com_all.dll 1, 0, 0, 1 80.00 KB (81,920 bytes)

5/31/2000 6:57:36 PM
c:\inetpub\wwwroot\tpcc_com_all.dll

dbmssocn.dll 1999.05.27.28.27 KB (28,944 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\dbmssocn.dll

ntwdblib.dll 2000.080.0100.00 268.06 KB (274,489 bytes)
5/31/2000 6:38:53 PM Microsoft Corporation
c:\winnt\system32\ntwdblib.dll

tpcc_dblib.dll Not Available 28.00 KB (28,672 bytes)
5/31/2000 6:57:35 PM Not Available
c:\inetpub\wwwroot\tpcc_dblib.dll

mfc42.dll 6.00.8576.0 972.05 KB (995,383 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\mfc42.dll

wam.dll 5.00.0984 69.77 KB (71,440 bytes) 6/1/2000 2:24:37 AM Microsoft Corporation
c:\winnt\system32\inetsrv\wam.dll

odbcint.dll 3.520.5215.0 88.00 KB (90,112 bytes)
5/31/2000 6:38:21 PM Microsoft Corporation
c:\winnt\system32\odbcint.dll

comdlg32.dll 5.00.2919.3800 235.27 KB (240,912 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\comdlg32.dll

odbc32.dll 3.520.5215.0 216.27 KB (221,456 bytes)
5/31/2000 6:38:20 PM Microsoft Corporation
c:\winnt\system32\odbc32.dll

comsvcs.dll 1999.8.3413.5 1.17 MB (1,224,976 bytes)
6/1/2000 2:23:45 AM Microsoft Corporation
c:\winnt\system32\comsvcs.dll

iislog.dll 5.00.0984 75.77 KB (77,584 bytes) 6/1/2000 2:24:30 AM Microsoft Corporation
c:\winnt\system32\inetsrv\iislog.dll

wshnetbs.dll 5.00.2090.1 7.77 KB (7,952 bytes)
9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\wshnetbs.dll

inetsloc.dll 5.00.0984 20.27 KB (20,752 bytes) 6/1/2000 2:24:32 AM Microsoft Corporation
c:\winnt\system32\inetsloc.dll

isatq.dll 5.00.0984 60.27 KB (61,712 bytes) 6/1/2000 2:24:32 AM Microsoft Corporation
c:\winnt\system32\inetsrv\isatq.dll

security.dll 5.00.2112.1 5.77 KB (5,904 bytes) 9/10/1999 9:00:00 AM Microsoft Corporation
c:\winnt\system32\security.dll

svcxext.dll 5.00.0984 39.27 KB (40,208 bytes) 6/1/2000 2:24:31 AM Microsoft Corporation
c:\winnt\system32\inetsrv\svcxext.dll

admexs.dll 5.00.0984 27.77 KB (28,432 bytes) 6/1/2000 2:24:29 AM Microsoft Corporation
c:\winnt\system32\inetsrv\admexs.dll

wamreg.dll 5.00.0984 44.77 KB (45,840 bytes) 6/1/2000 2:24:37 AM Microsoft Corporation
c:\winnt\system32\inetsrv\wamreg.dll

metadata.dll 5.00.0984 68.27 KB (69,904 bytes)
6/1/2000 2:24:31 AM Microsoft Corporation
c:\winnt\system32\inetsrv\metadata.dll

iismap.dll 5.00.0984 56.27 KB (57,616 bytes) 6/1/2000 2:24:32 AM Microsoft Corporation
c:\winnt\system32\iismap.dll

nsepm.dll 5.00.0984 43.27 KB (44,304 bytes) 6/1/2000 2:24:31 AM Microsoft Corporation
c:\winnt\system32\inetsrv\nsepm.dll

admwprox.dll 5.00.0984 31.27 KB (32,016 bytes)
6/1/2000 2:24:31 AM Microsoft Corporation
c:\winnt\system32\admwprox.dll

coadmin.dll 5.00.0984 39.27 KB (40,208 bytes) 6/1/2000 2:24:32 AM Microsoft Corporation
c:\winnt\system32\inetsrv\coadmin.dll

iisadmin.dll 5.00.0984 14.77 KB (15,120 bytes) 6/1/2000 2:24:30 AM Microsoft Corporation

c:\winnt\system32\inetsrv\iisadmin.dll
 rpcref.dll 5.00.0984 4.27 KB (4,368 bytes) 6/1/2000 2:24:31 AM
 Microsoft Corporation c:\winnt\system32\inetsrv\rpcref.dll
 iisrtl.dll 5.00.0984 119.27 KB (122,128 bytes) 6/1/2000
 2:24:31 AM Microsoft Corporation c:\winnt\system32\iisrtl.dll
 inetinfo.exe 5.00.0984 14.27 KB (14,608 bytes) 6/1/2000
 2:24:30 AM Microsoft Corporation
 c:\winnt\system32\inetsrv\inetinfo.exe
 msidle.dll 5.00.2919.3800 6.27 KB (6,416 bytes) 9/10/1999
 9:00:00 AM Microsoft Corporation c:\winnt\system32\msidle.dll
 mstask.exe 4.71.2113.1 114.77 KB (117,520 bytes)
 5/31/2000 5:29:26 PM Microsoft Corporation
 c:\winnt\system32\mstask.exe
 regsvc.exe 5.00.2091.1 63.77 KB (65,296 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\regsvc.exe
 llsrc.dll 5.00.2107.1 45.77 KB (46,864 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\llsrc.dll
 llssrv.exe 5.00.2090.1 113.77 KB (116,496 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\llssrv.exe
 rasdlg.dll 5.00.2120.1 512.77 KB (525,072 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\rasdlg.dll
 netcfgx.dll 5.00.2120.1 532.27 KB (545,040 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\netcfgx.dll
 rasmans.dll 5.00.2119.1 150.27 KB (153,872 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\rasmans.dll
 wmi.dll 5.00.2112.1 6.27 KB (6,416 bytes) 9/10/1999
 9:00:00 AM Microsoft Corporation c:\winnt\system32\wmi.dll
 netshell.dll 5.00.2120.1 453.77 KB (464,656 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\netshell.dll
 netman.dll 5.00.2120.1 88.77 KB (90,896 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\netman.dll
 ntmsdba.dll 5.00.2108.1 167.27 KB (171,280 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\ntmsdba.dll
 sens.dll 5.00.2090.1 35.77 KB (36,624 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\sens.dll
 ntmssvc.dll 5.00.2108.1 390.27 KB (399,632 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\ntmssvc.dll
 es.dll 1999.8.3413.3 220.77 KB (226,064 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\es.dll
 mtxoci.dll 1999.8.3413.3 101.77 KB (104,208 bytes)
 6/1/2000 2:23:52 AM Microsoft Corporation
 c:\winnt\system32\mtxoci.dll
 resutils.dll 5.00.2123.1 39.77 KB (40,720 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\resutils.dll
 clusapi.dll 5.00.2123.1 49.27 KB (50,448 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\clusapi.dll
 msvcp50.dll 5.00.7051 552.50 KB (565,760 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation

c:\winnt\system32\msvcp50.dll
 xolehlp.dll 1999.8.3413.3 18.27 KB (18,704 bytes)
 6/1/2000 2:23:51 AM Microsoft Corporation
 c:\winnt\system32\xolehlp.dll
 msdtclog.dll 1999.8.3413.3 85.27 KB (87,312
 bytes) 6/1/2000 2:23:51 AM Microsoft Corporation
 c:\winnt\system32\msdtclog.dll
 mtxclu.dll 1999.8.3413.3 50.77 KB (51,984 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\mtxclu.dll
 msdtcprx.dll 1999.8.3413.7 625.77 KB (640,784
 bytes) 6/1/2000 2:23:52 AM Microsoft Corporation
 c:\winnt\system32\msdtcprx.dll
 txfaux.dll 1999.8.3413.7 370.77 KB (379,664 bytes)
 6/1/2000 2:23:51 AM Microsoft Corporation
 c:\winnt\system32\txfaux.dll
 msdtctm.dll 1999.8.3413.7 1.07 MB (1,120,528 bytes)
 6/1/2000 2:23:52 AM Microsoft Corporation
 c:\winnt\system32\msdtctm.dll
 msdtc.exe 1999.8.3413.3 6.77 KB (6,928 bytes) 6/1/2000
 2:23:51 AM Microsoft Corporation c:\winnt\system32\msdtc.exe
 rasadhlp.dll 5.00.2109.1 7.27 KB (7,440 bytes) 9/10/1999
 9:00:00 AM Microsoft Corporation c:\winnt\system32\rasadhlp.dll
 winnr.dll 5.00.2110.1 18.77 KB (19,216 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\winnr.dll
 rpcss.dll 5.00.2119.1 225.27 KB (230,672 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\rpcss.dll
 svchost.exe 5.00.2090.1 7.77 KB (7,952 bytes) 9/10/1999
 9:00:00 AM Microsoft Corporation c:\winnt\system32\svchost.exe
 iissuba.dll 5.00.0984 9.77 KB (10,000 bytes) 9/10/1999 9:00:00 AM
 Microsoft Corporation c:\winnt\system32\iissuba.dll
 dssbase.dll 5.00.2120.1 140.27 KB (143,632 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\dssbase.dll
 wshtcpip.dll 5.00.2090.1 17.27 KB (17,680 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\wshtcpip.dll
 msafd.dll 5.00.2095.1 52.27 KB (53,520 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\msafd.dll
 oakley.dll 5.00.2115.1 420.27 KB (430,352 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\oakley.dll
 mfc42u.dll 6.00.8576.0 972.05 KB (995,384 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\mfc42u.dll
 polagent.dll 5.00.2110.1 102.27 KB (104,720 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\polagent.dll
 scecli.dll 5.00.2112.1 101.77 KB (104,208 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\scecli.dll
 esent.dll 6.0.3938.7 848.77 KB (869,136 bytes) 9/10/1999
 9:00:00 AM Microsoft Corporation c:\winnt\system32\esent.dll
 mswsock.dll 5.00.2120.1 62.77 KB (64,272
 bytes) 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\mswsock.dll
 ntdsatq.dll 5.00.2122.1 30.77 KB (31,504 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\ntdsatq.dll

ntdsa.dll 5.00.2127.1 984.77 KB (1,008,400 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\ntdsa.dll
 kdcsvc.dll 5.00.2121.1 138.77 KB (142,096 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\kdcsvc.dll
 sfmapi.dll 5.00.2090.1 38.77 KB (39,696 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\sfmapi.dll
 rassfm.dll 5.00.2109.1 21.27 KB (21,776 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\rassfm.dll
 mpr.dll 5.00.2111.1 53.27 KB (54,544 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\mpr.dll
 schannel.dll 5.00.2118.1 136.77 KB (140,048 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\schannel.dll
 netlogon.dll 5.00.2119.1 344.77 KB (353,040 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\netlogon.dll
 msv1_0.dll 5.00.2113.1 93.77 KB (96,016 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\msv1_0.dll
 kerberos.dll 5.00.2121.1 190.27 KB (194,832 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\kerberos.dll
 msprvs.dll 5.00.2112.1 41.50 KB (42,496 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\msprvs.dll
 samsrv.dll 5.00.2124.1 352.27 KB (360,720 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\samsrv.dll
 cryptdll.dll 5.00.2112.1 40.27 KB (41,232 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\cryptdll.dll
 lsasrv.dll 5.00.2121.1 483.77 KB (495,376 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\lsasrv.dll
 lsass.exe 5.00.2121.1 32.77 KB (33,552 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\lsass.exe
 dhcpcsvc.dll 5.00.2107.1 88.77 KB (90,896
 bytes) 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\dhcpcsvc.dll
 tapi32.dll 5.00.2090.1 122.27 KB (125,200 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\tapi32.dll
 rasman.dll 5.00.2114.1 59.77 KB (61,200 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\rasman.dll
 rasapi32.dll 5.00.2116.1 187.27 KB (191,760 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\rasapi32.dll
 rtutils.dll 5.00.2109.1 43.27 KB (44,304 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\rtutils.dll
 activeds.dll 5.00.2118.1 171.77 KB (175,888 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\activeds.dll
 mprapi.dll 5.00.2112.1 90.77 KB (92,944 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation

c:\winnt\system32\mprapi.dll
 iphlpapi.dll 5.00.2095.2 67.27 KB (68,880 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\iphlpapi.dll
 nr20.dll 5.00.2120.1 35.27 KB (36,112 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\nr20.dll
 msi.dll 1.10.0816.3 1.64 MB (1,715,984 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\msi.dll
 adslrpc.dll 5.00.2120.1 125.77 KB (128,784 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\adslrpc.dll
 appmgmts.dll 5.00.2109.1 115.77 KB (118,544 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\appmgmts.dll
 ntlsap.dll 5.00.2090.1 6.77 KB (6,928 bytes) 9/10/1999
 9:00:00 AM Microsoft Corporation c:\winnt\system32\ntlsapi.dll
 wmicore.dll 5.00.2119.1 70.27 KB (71,952 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\wmicore.dll
 alrsvc.dll 5.00.2090.1 17.77 KB (18,192 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\alrsvc.dll
 trkwks.dll 5.00.2110.1 87.77 KB (89,872 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\trkwks.dll
 seclogon.dll 5.00.2122.1 15.27 KB (15,632 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\seclogon.dll
 psbase.dll 5.00.2090.1 110.77 KB (113,424 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\psbase.dll
 cryptsvc.dll 5.00.2090.1 66.77 KB (68,368 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\cryptsvc.dll
 wkssvc.dll 5.00.2120.1 91.27 KB (93,456 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\wkssvc.dll
 srsvcs.dll 5.00.2117.1 79.27 KB (81,168 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\srsvcs.dll
 cfgmgr32.dll 5.00.2098.1 16.77 KB (17,168 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\cfgmgr32.dll
 dmserver.dll 2121.1.286.1 11.77 KB (12,048 bytes)
 9/10/1999 9:00:00 AM VERITAS Software Corp.
 c:\winnt\system32\dmserver.dll
 winsta.dll 5.00.2100.1 36.27 KB (37,136 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\winsta.dll
 icmp.dll 5.00.2090.1 7.27 KB (7,440 bytes) 9/10/1999
 9:00:00 AM Microsoft Corporation c:\winnt\system32\icmp.dll
 lmhsvc.dll 5.00.2102.1 9.27 KB (9,488 bytes) 9/10/1999
 9:00:00 AM Microsoft Corporation c:\winnt\system32\lmhsvc.dll
 eventlog.dll 5.00.2090.1 43.77 KB (44,816 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\eventlog.dll
 ntdsapi.dll 5.00.2120.1 55.27 KB (56,592 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\ntdsapi.dll
 scesrv.dll 5.00.2112.1 220.27 KB (225,552 bytes)

9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\scesrv.dll
 umpnmgm.dll 5.00.2109.1 116.77 KB (119,568 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\umpnmgm.dll
 services.exe 5.00.2106.1 87.27 KB (89,360 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\services.exe
 atl.dll 3.00.8449 57.56 KB (58,938 bytes) 9/10/1999
 9:00:00 AM Microsoft Corporation c:\winnt\system32\atl.dll
 certcli.dll 5.00.2120.1 131.27 KB (134,416 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\certcli.dll
 clbcatq.dll 1999.8.3413.3 494.27 KB (506,128 bytes)
 6/1/2000 2:23:44 AM Microsoft Corporation
 c:\winnt\system32\clbcatq.dll
 oleaut32.dll 2.40.4505 596.27 KB (610,576 bytes) 9/10/1999
 9:00:00 AM Microsoft Corporation c:\winnt\system32\oleaut32.dll
 cscui.dll 5.00.2116.1 225.77 KB (231,184 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\cscui.dll
 winspool.drv 5.00.2110.1 109.77 KB (112,400 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\winspool.drv
 winscard.dll 5.00.2108.1 77.27 KB (79,120 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\winscard.dll
 wlnotify.dll 5.00.2090.1 52.77 KB (54,032 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\wlnotify.dll
 cscdll.dll 5.00.2122.1 97.77 KB (100,112 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\cscdll.dll
 winmm.dll 5.00.2114.1 184.27 KB (188,688 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\winmm.dll
 lz32.dll 5.00.2090.1 9.77 KB (10,000 bytes) 9/10/1999
 9:00:00 AM Microsoft Corporation c:\winnt\system32\lz32.dll
 version.dll 5.00.2090.1 15.77 KB (16,144 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\version.dll
 rsabase.dll 5.00.2120.1 127.77 KB (130,832 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\rsabase.dll
 setupapi.dll 5.00.2126.1 551.27 KB (564,496 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\setupapi.dll
 msct32.dll 5.131.2090.1 7.77 KB (7,952 bytes) 9/10/1999
 9:00:00 AM Microsoft Corporation c:\winnt\system32\msct32.dll
 ole32.dll 5.00.2120.1 961.27 KB (984,336 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\ole32.dll
 imagehlp.dll 5.00.2128.1 40.77 KB (41,744 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\imagehlp.dll
 msasn1.dll 5.00.2090.1 50.27 KB (51,472 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\msasn1.dll
 crypt32.dll 5.131.2118.1 454.77 KB (465,680 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\crypt32.dll
 wintrust.dll 5.131.2090.1 161.27 KB (165,136 bytes)

9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\wintrust.dll
 comctl32.dll 5.81 539.77 KB (552,720 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\comctl32.dll
 shlwapi.dll 5.00.2919.3800 281.77 KB (288,528 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\shlwapi.dll
 shell32.dll 5.00.2919.3800 2.24 MB (2,344,208 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\shell32.dll
 msgina.dll 5.00.2115.1 308.27 KB (315,664 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\msgina.dll
 wsock32.dll 5.00.2120.1 21.27 KB (21,776 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\wsock32.dll
 dnsapi.dll 5.00.2120.1 133.77 KB (136,976 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\dnsapi.dll
 wldap32.dll 5.00.2117.1 153.77 KB (157,456 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\wldap32.dll
 ws2help.dll 5.00.2095.1 17.77 KB (18,192 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\ws2help.dll
 ws2_32.dll 5.00.2104.1 67.77 KB (69,392 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\ws2_32.dll
 samlib.dll 5.00.2124.1 46.27 KB (47,376 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\samlib.dll
 netrap.dll 5.00.2090.1 11.27 KB (11,536 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\netrap.dll
 netapi32.dll 5.00.2120.1 295.77 KB (302,864 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\netapi32.dll
 profmap.dll 5.00.2112.1 27.27 KB (27,920 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\profmap.dll
 secur32.dll 5.00.2119.1 44.77 KB (45,840 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\secur32.dll
 sfc.dll 5.00.2124.1 83.27 KB (85,264 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\sfc.dll
 nddeapi.dll 5.00.2090.1 15.27 KB (15,632 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\nddeapi.dll
 userenv.dll 5.00.2127.1 343.27 KB (351,504 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\userenv.dll
 user32.dll 5.00.2120.1 392.27 KB (401,680 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\user32.dll
 gdi32.dll 5.00.2115.1 228.27 KB (233,744 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\gdi32.dll
 rpctx.dll 5.00.2128.1 440.27 KB (450,832 bytes)
 9/10/1999 9:00:00 AM Microsoft Corporation
 c:\winnt\system32\rpctx.dll

advapi32.dll 5.00.2120.1 337.77 KB (345,872 bytes) 9/10/1999 9:00:00 AM Microsoft Corporation c:\winnt\system32\advapi32.dll
kernel32.dll 5.00.2122.1 711.77 KB (728,848 bytes) 9/10/1999 9:00:00 AM Microsoft Corporation c:\winnt\system32\kernel32.dll
msvcrt.dll 6.10.8581.0 284.05 KB (290,869 bytes) 9/10/1999 9:00:00 AM Microsoft Corporation c:\winnt\system32\msvcrt.dll
winlogon.exe 5.00.2116.1 171.27 KB (175,376 bytes) 9/10/1999 9:00:00 AM Microsoft Corporation c:\winnt\system32\winlogon.exe
sfcdll.dll 5.00.2128.1 366.77 KB (375,568 bytes) 9/10/1999 9:00:00 AM Microsoft Corporation c:\winnt\system32\sfcdll.dll
ntdll.dll 5.00.2121.1 469.27 KB (480,528 bytes) 9/10/1999 9:00:00 AM Microsoft Corporation c:\winnt\system32\ntdll.dll
smss.exe 5.00.2090.1 42.77 KB (43,792 bytes) 9/10/1999 9:00:00 AM Microsoft Corporation c:\winnt\system32\smss.exe

[Services]

Display Name	State	Start Mode	Service Type
Alerter	Running	Auto Start	Share Process
Application Management		Running	Demand Start Share Process
Computer Browser	Stopped	Demand Start	Share Process
Indexing Service	Stopped	Demand Start	Share Process
ClipBook	Stopped	Demand Start	Own Process
Distributed File System	Running	Auto Start	Own Process
DHCP Client	Stopped	Demand Start	Share Process
Logical Disk Manager	Administrative Service	Stopped	Demand Start Share Process
Logical Disk Manager	Running	Auto Start	Share Process
DNS Client	Stopped	Demand Start	Share Process
Event Log	Running	Auto Start	Share Process
COM+ Event System	Running	Demand Start	Share Process
Fax Service	Stopped	Demand Start	Own Process
IIS Admin Service	Running	Auto Start	Share Process
IMDB Server	Stopped	Disabled	Own Process
Intersite Messaging	Stopped	Disabled	Own Process
Kerberos Key Distribution Center	Stopped	Disabled	Share Process
Server	Running	Auto Start	Share Process
Workstation	Running	Auto Start	Share Process
License Logging Service	Running	Auto Start	Own Process
TCP/IP NetBIOS Helper	Running	Auto Start	Share Process
Messenger	Stopped	Demand Start	Share Process
NetMeeting Remote Desktop Sharing		Stopped	Demand Start Own Process
Distributed Transaction Coordinator	Running	Auto Start	Own Process
FTP Publishing Service	Stopped	Disabled	Share Process

Windows Installer	Stopped	Demand Start	Share Process
Network DDE	Stopped	Demand Start	Share Process
Network DDE DSDM	Stopped	Demand Start	Share Process
Net Logon	Stopped	Demand Start	Share Process
Network Connections	Running	Demand Start	Share Process
File Replication	Stopped	Demand Start	Own Process
NT LM Security Support Provider	Stopped	Demand Start	Share Process
Removable Storage	Running	Auto Start	Share Process
Plug and Play	Running	Auto Start	Share Process
IPSEC Policy Agent	Running	Auto Start	Share Process
Protected Storage	Running	Auto Start	Share Process
Remote Access Auto Connection Manager	Stopped	Demand Start	Share Process
Remote Access Connection Manager	Stopped	Demand Start	Share Process
Routing and Remote Access	Stopped	Disabled	Share Process
Remote Registry Service	Running	Auto Start	Own Process
Remote Procedure Call (RPC) Locator	Stopped	Demand Start	Own Process
Remote Procedure Call (RPC)	Running	Auto Start	Share Process
QoS RSVP	Stopped	Demand Start	Own Process
Security Accounts Manager	Running	Auto Start	Share Process
Smart Card Helper	Stopped	Demand Start	Share Process
Smart Card	Stopped	Demand Start	Share Process
Task Scheduler	Running	Auto Start	Share Process
RunAs Service	Running	Auto Start	Share Process
System Event Notification	Running	Auto Start	Share Process
Internet Connection Sharing	Stopped	Demand Start	Share Process
Simple Mail Transport Protocol (SMTP)	Stopped	Disabled	Share Process
Print Spooler	Stopped	Demand Start	Own Process
Performance Logs and Alerts	Stopped	Demand Start	Own Process
Telephony	Running	Demand Start	Share Process
Terminal Services	Stopped	Disabled	Own Process
Telnet	Stopped	Demand Start	Own Process
Distributed Link Tracking Server	Stopped	Demand Start	Share Process
Distributed Link Tracking Client	Running	Auto Start	Share Process
Uninterruptible Power Supply	Stopped	Demand Start	Own Process
Utility Manager	Stopped	Demand Start	Own Process
Windows Time	Stopped	Demand Start	Share Process
World Wide Web Publishing Service	Stopped	Auto Start	Share Process

Windows Management Instrumentation	Running	Demand Start	Own Process
Windows Management Instrumentation Driver Extensions	Running	Demand Start	Share Process

[Program Groups]

Group Name	Name	User Name	Default
Accessories	Default User:Accessories		Default User
Accessories\Accessibility		Default User:Accessories\Accessibility	Default User
Accessories\Entertainment		User:Accessories\Entertainment	Default User
Accessories\System Tools		Default User:Accessories\System Tools	Default User
Startup	Default User:Startup		Default User
Accessories	All Users:Accessories		All Users
Accessories\Accessibility		All Users:Accessories\Accessibility	All Users
Accessories\Communications		All Users:Accessories\Communications	All Users
Accessories\Entertainment		All Users:Accessories\Entertainment	All Users
Accessories\Games		All Users:Accessories\Games	All Users
Accessories\Microsoft Script Debugger		All Users:Accessories\Microsoft Script Debugger	All Users
Accessories\System Tools		All Users:Accessories\System Tools	All Users
Administrative Tools		All Users:Administrative Tools	All Users
Microsoft SQL Server		All Users:Microsoft SQL Server	All Users
Startup	All Users:Startup		All Users
Accessories	ACL01\Administrator:Accessories		ACL01\Administrator
Accessories\Accessibility		ACL01\Administrator:Accessories\Accessibility	ACL01\Administrator
Accessories\Entertainment		ACL01\Administrator:Accessories\Entertainment	ACL01\Administrator
Accessories\System Tools		ACL01\Administrator:Accessories\System Tools	ACL01\Administrator
Administrative Tools		ACL01\Administrator:Administrative Tools	ACL01\Administrator
Startup	ACL01\Administrator:Startup		ACL01\Administrator

[Startup Programs]

Program	Command	User Name	Default
mdac_runonce	c:\winnt\system32\runonce.exe		All Users

<Microsoft SQL Server 2000 setting>

Startup Parameters

```
sqlservr -c -x -t3502
```

- c Start SQL Server independently of the Microsoft Windows NT Service Control Manager.
- x Disable the keeping of CPU time and cache hit ratio statistics.
- t3502 Prints a message to the log at the beginning and end of each checkpoint.
- g100 Reserve 100 MB for non-buffer pool allocations

Microsoft SQL Server Stack Size

The default stack size of Microsoft SQL Server 2000 was changed using the EDITBIN utility. The EDITBIN utility ships with Microsoft Visual C++ . The command used was to change the stack size is:

```
editbin /stack: 131072 sqlservr.exe.
```

Microsoft SQL Server 2000 Configuration Parameters

```
1> 2> 3> 4> 5> 6> 7> 8> 9> 10> 11>
-- File: VERSION.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Returns SQL Server version string
```

```
print " "
select convert(char(30), getdate(),9)
print " "
```

```
-----
Jun 5 2000 11:00:23:327AM
```

```
(1 row affected)
```

```
1> 2> 3>
select @@version
```

```
-----
-----
Microsoft SQL Server 2000 - 8.00.100 (Intel X86)
Apr 18 2000 01:19:00
Cop
yright (c) 1988-2000 Microsoft Corporation
Enterprise Edition on Windo
ws NT 5.0 (Build 2195: )
```

```
(1 row affected)
1> 2>
1> 2> 3> 4> 5> 6> 7> 8> 9> 10>
-- File: CONFIG.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.21
-- Copyright Microsoft, 1999, 2000
-- Purpose: Collects SQL Server configuration parameters
```

```
print " "
select convert(char(30), getdate(),9)
print " "
```

```
-----
Jun 5 2000 11:00:24:420AM
```

```
(1 row affected)
```

```
1> 2> 3> DBCC execution completed. If DBCC printed error messages, contact your system administrator.
Configuration option 'show advanced options' changed from 1 to 1. Run the RECONFIGURE statement to install.
```

```
sp_configure "show advanced",1
1> 2> reconfigure with override
1> 2> sp_configure
```

name	minimum	maximum	config_value	run_value
affinity mask	0	2147483647	15	15
allow updates	0	1	0	0
awe enabled	0	1	1	1
c2 audit mode	0	1	0	0
cost threshold for parallelism	0	32767	5	5
cursor threshold	-1	2147483647	-1	-1
default full-text language	0	2147483647	1033	1033
default language	0	9999	0	0
fill factor (%)	0	100	0	0
index create memory (KB)	704	2147483647	704	704
lightweight pooling	0	1	1	1

```

locks 5000 2147483647 0 0
max degree of parallelism 0 32 1 1
max server memory (MB) 4 2147483647 2147483647
2147483647
max text repl size (B) 0 2147483647 65536 65536
max worker threads 10 32767 190 190
media retention 0 365 0 0
min memory per query (KB) 512 2147483647 512 512
min server memory (MB) 0 2147483647 0 0
nested triggers 0 1 1 1
network packet size (B) 512 65536 4096 4096
open objects 0 2147483647 0 0
priority boost 0 1 1 1
query governor cost limit 0 2147483647 0 0
query wait (s) -1 2147483647 -1 -1
recovery interval (min) 0 32767 32767 32767
remote access 0 1 1 1
remote login timeout (s) 0 2147483647 20 20
remote proc trans 0 1 0 0
remote query timeout (s) 0 2147483647 0 0
scan for startup procs 0 1 0 0
set working set size 0 1 0 0
show advanced options 0 1 1 1
two digit year cutoff 1753 9999 2049 2049
user connections 0 32767 0 0
user options 0 16383 0 0

```

1>

<Disk Array configuration>

```
##### Controller 0 #####
Begin
```

```
BeginGroup
```

```
PhysicalDevice0 = Channel=0, Target=0, Size=86592mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
```

```
PhysicalDevice1 = Channel=2, Target=0, Size=86592mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
```

```
LogicalDevice0 = StripeSize=64kb, Raid=1, WriteThrough=1, Size=86592mb, BIOSGeometry=8GB,
(PhysicalDevice0, StartAddress=0mb, Size=86592mb),
(PhysicalDevice1, StartAddress=0mb, Size=86592mb);
```

```
EndGroup
```

```
BeginControllerParameter
```

```
ControllerName = eXtremeRAID 2000;
```

```
ControllerType = 28;
```

```
FirmwareVersion = 5.60;
```

```
CacheLineSize = 8KB;
BackgroundTaskRate = 50;
InitiatorID = 7;
DiskStartupMode = AutoSpin;
DevicesPerSpin = 2;
InitialDelay = 6S;
SequentialDelay = 0S;
EnableDriveSizing = 1;
EnableClustering = 0;
EnableBGInit = 1;
EnableReadAhead = 1;
EnableBiosLoadDelay = 0;
EnableForcedUnitAccess = 1;
DisableBios = 1;
EnableCDROMBoot = 0;
EnableStorageWorks = 0;
EnableSAFTE = 1;
EnableSES = 1;
EnableARM = 0;
EnableOFM = 0;
OEMCode = 0;
StartupOption = 0;
```

```
EndControllerParameter
```

```
End
```

```
##### Controller 1 #####
Begin
```

```
BeginGroup
```

```
PhysicalDevice0 = Channel=0, Target=0, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
```

```
PhysicalDevice1 = Channel=0, Target=1, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
```

```
PhysicalDevice2 = Channel=0, Target=2, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
```

```
PhysicalDevice3 = Channel=0, Target=3, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
```

```
PhysicalDevice4 = Channel=0, Target=4, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
```


PhysicalDevice41 = Channel=2, Target=15, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice42 = Channel=3, Target=0, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice43 = Channel=3, Target=1, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice44 = Channel=3, Target=2, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice45 = Channel=3, Target=3, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice46 = Channel=3, Target=4, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice47 = Channel=3, Target=5, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice48 = Channel=3, Target=8, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice49 = Channel=3, Target=9, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice50 = Channel=3, Target=10, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice51 = Channel=3, Target=11, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice52 = Channel=3, Target=12, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice53 = Channel=3, Target=13, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice54 = Channel=3, Target=14, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice55 = Channel=3, Target=15, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 IntermediateDevice0 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
 (PhysicalDevice0, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice1, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice2, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice3, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice4, StartAddress=0mb, Size=8392mb),

(PhysicalDevice5, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice6, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice7, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice8, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice9, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice10, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice11, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice12, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice13, StartAddress=0mb, Size=8392mb);
 IntermediateDevice1 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
 (PhysicalDevice14, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice15, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice16, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice17, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice18, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice19, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice20, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice21, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice22, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice23, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice24, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice25, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice26, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice27, StartAddress=0mb, Size=8392mb);
 IntermediateDevice2 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
 (PhysicalDevice28, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice29, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice30, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice31, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice32, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice33, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice34, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice35, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice36, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice37, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice38, StartAddress=0mb, Size=8392mb),

```

(PhysicalDevice39, StartAddress=0mb, Size=8392mb),
(PhysicalDevice40, StartAddress=0mb, Size=8392mb),
(PhysicalDevice41, StartAddress=0mb, Size=8392mb);
IntermediateDevice3 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
(PhysicalDevice42, StartAddress=0mb, Size=8392mb),
(PhysicalDevice43, StartAddress=0mb, Size=8392mb),
(PhysicalDevice44, StartAddress=0mb, Size=8392mb),
(PhysicalDevice45, StartAddress=0mb, Size=8392mb),
(PhysicalDevice46, StartAddress=0mb, Size=8392mb),
(PhysicalDevice47, StartAddress=0mb, Size=8392mb),
(PhysicalDevice48, StartAddress=0mb, Size=8392mb),
(PhysicalDevice49, StartAddress=0mb, Size=8392mb),
(PhysicalDevice50, StartAddress=0mb, Size=8392mb),
(PhysicalDevice51, StartAddress=0mb, Size=8392mb),
(PhysicalDevice52, StartAddress=0mb, Size=8392mb),
(PhysicalDevice53, StartAddress=0mb, Size=8392mb),
(PhysicalDevice54, StartAddress=0mb, Size=8392mb),
(PhysicalDevice55, StartAddress=0mb, Size=8392mb);
LogicalDevice0 = StripeSize=64kb, Raid=12, WriteThrough=1, Size=469952mb, BIOSGeometry=2GB,
(IntermediateDevice0, StartAddress=0mb, Size=117488mb),
(IntermediateDevice1, StartAddress=0mb, Size=117488mb),
(IntermediateDevice2, StartAddress=0mb, Size=117488mb),
(IntermediateDevice3, StartAddress=0mb, Size=117488mb);

```

EndGroup

BeginControllerParameter

```

ControllerName = eXtremeRAID 2000;
ControllerType = 28;
FirmwareVersion = 5.60;
CacheLineSize = 8KB;
BackgroundTaskRate = 50;
InitiatorID = 7;
DiskStartupMode = AutoSpin;
DevicesPerSpin = 2;
InitialDelay = 6S;
SequentialDelay = 240S;
EnableDriveSizing = 1;

```

```

EnableClustering = 0;
EnableBGInit = 1;
EnableReadAhead = 1;
EnableBiosLoadDelay = 0;
EnableForcedUnitAccess = 1;
DisableBios = 1;
EnableCDROMBoot = 0;
EnableStorageWorks = 0;
EnableSAFTE = 1;
EnableSES = 1;
EnableARM = 1;
EnableOFM = 1;
OEMCode = 0;
StartupOption = 0;

```

EndControllerParameter

End

Controller 2

Begin

BeginGroup

```

PhysicalDevice0 = Channel=0, Target=0, Size=8392mb, State=Online,
    TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice1 = Channel=0, Target=1, Size=8392mb, State=Online,
    TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice2 = Channel=0, Target=2, Size=8392mb, State=Online,
    TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice3 = Channel=0, Target=3, Size=8392mb, State=Online,
    TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice4 = Channel=0, Target=4, Size=8392mb, State=Online,
    TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice5 = Channel=0, Target=5, Size=8392mb, State=Online,
    TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice6 = Channel=0, Target=8, Size=8392mb, State=Online,
    TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice7 = Channel=0, Target=9, Size=8392mb, State=Online,
    TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice8 = Channel=0, Target=10, Size=8392mb, State=Online,
    TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;

```


PhysicalDevice45 = Channel=3, Target=3, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice46 = Channel=3, Target=4, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice47 = Channel=3, Target=5, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice48 = Channel=3, Target=8, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice49 = Channel=3, Target=9, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice50 = Channel=3, Target=10, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice51 = Channel=3, Target=11, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice52 = Channel=3, Target=12, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice53 = Channel=3, Target=13, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice54 = Channel=3, Target=14, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice55 = Channel=3, Target=15, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
IntermediateDevice0 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
(PhysicalDevice0, StartAddress=0mb, Size=8392mb),
(PhysicalDevice1, StartAddress=0mb, Size=8392mb),
(PhysicalDevice2, StartAddress=0mb, Size=8392mb),
(PhysicalDevice3, StartAddress=0mb, Size=8392mb),
(PhysicalDevice4, StartAddress=0mb, Size=8392mb),
(PhysicalDevice5, StartAddress=0mb, Size=8392mb),
(PhysicalDevice6, StartAddress=0mb, Size=8392mb),
(PhysicalDevice7, StartAddress=0mb, Size=8392mb),
(PhysicalDevice8, StartAddress=0mb, Size=8392mb),
(PhysicalDevice9, StartAddress=0mb, Size=8392mb),
(PhysicalDevice10, StartAddress=0mb, Size=8392mb),
(PhysicalDevice11, StartAddress=0mb, Size=8392mb),
(PhysicalDevice12, StartAddress=0mb, Size=8392mb),

(PhysicalDevice13, StartAddress=0mb, Size=8392mb);
IntermediateDevice1 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
(PhysicalDevice14, StartAddress=0mb, Size=8392mb),
(PhysicalDevice15, StartAddress=0mb, Size=8392mb),
(PhysicalDevice16, StartAddress=0mb, Size=8392mb),
(PhysicalDevice17, StartAddress=0mb, Size=8392mb),
(PhysicalDevice18, StartAddress=0mb, Size=8392mb),
(PhysicalDevice19, StartAddress=0mb, Size=8392mb),
(PhysicalDevice20, StartAddress=0mb, Size=8392mb),
(PhysicalDevice21, StartAddress=0mb, Size=8392mb),
(PhysicalDevice22, StartAddress=0mb, Size=8392mb),
(PhysicalDevice23, StartAddress=0mb, Size=8392mb),
(PhysicalDevice24, StartAddress=0mb, Size=8392mb),
(PhysicalDevice25, StartAddress=0mb, Size=8392mb),
(PhysicalDevice26, StartAddress=0mb, Size=8392mb),
(PhysicalDevice27, StartAddress=0mb, Size=8392mb);
IntermediateDevice2 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
(PhysicalDevice28, StartAddress=0mb, Size=8392mb),
(PhysicalDevice29, StartAddress=0mb, Size=8392mb),
(PhysicalDevice30, StartAddress=0mb, Size=8392mb),
(PhysicalDevice31, StartAddress=0mb, Size=8392mb),
(PhysicalDevice32, StartAddress=0mb, Size=8392mb),
(PhysicalDevice33, StartAddress=0mb, Size=8392mb),
(PhysicalDevice34, StartAddress=0mb, Size=8392mb),
(PhysicalDevice35, StartAddress=0mb, Size=8392mb),
(PhysicalDevice36, StartAddress=0mb, Size=8392mb),
(PhysicalDevice37, StartAddress=0mb, Size=8392mb),
(PhysicalDevice38, StartAddress=0mb, Size=8392mb),
(PhysicalDevice39, StartAddress=0mb, Size=8392mb),
(PhysicalDevice40, StartAddress=0mb, Size=8392mb),
(PhysicalDevice41, StartAddress=0mb, Size=8392mb);
IntermediateDevice3 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
(PhysicalDevice42, StartAddress=0mb, Size=8392mb),
(PhysicalDevice43, StartAddress=0mb, Size=8392mb),
(PhysicalDevice44, StartAddress=0mb, Size=8392mb),
(PhysicalDevice45, StartAddress=0mb, Size=8392mb),

```
(PhysicalDevice46, StartAddress=0mb, Size=8392mb),
(PhysicalDevice47, StartAddress=0mb, Size=8392mb),
(PhysicalDevice48, StartAddress=0mb, Size=8392mb),
(PhysicalDevice49, StartAddress=0mb, Size=8392mb),
(PhysicalDevice50, StartAddress=0mb, Size=8392mb),
(PhysicalDevice51, StartAddress=0mb, Size=8392mb),
(PhysicalDevice52, StartAddress=0mb, Size=8392mb),
(PhysicalDevice53, StartAddress=0mb, Size=8392mb),
(PhysicalDevice54, StartAddress=0mb, Size=8392mb),
(PhysicalDevice55, StartAddress=0mb, Size=8392mb);
```

```
LogicalDevice0 = StripeSize=64kb, Raid=12, WriteThrough=1, Size=469952mb, BIOSGeometry=8GB,
```

```
(IntermediateDevice0, StartAddress=0mb, Size=117488mb),
(IntermediateDevice1, StartAddress=0mb, Size=117488mb),
(IntermediateDevice2, StartAddress=0mb, Size=117488mb),
(IntermediateDevice3, StartAddress=0mb, Size=117488mb);
```

```
EndGroup
```

```
BeginControllerParameter
```

```
ControllerName = eXtremeRAID 2000;
ControllerType = 28;
FirmwareVersion = 5.60;
CacheLineSize = 8KB;
BackgroundTaskRate = 50;
InitiatorID = 7;
DiskStartupMode = AutoSpin;
DevicesPerSpin = 2;
InitialDelay = 6S;
SequentialDelay = 0S;
EnableDriveSizing = 1;
EnableClustering = 0;
EnableBGInit = 1;
EnableReadAhead = 0;
EnableBiosLoadDelay = 0;
EnableForcedUnitAccess = 0;
DisableBios = 1;
EnableCDROMBoot = 0;
EnableStorageWorks = 0;
```

```
EnableSAFTE = 1;
EnableSES = 1;
EnableARM = 1;
EnableOFM = 1;
OEMCode = 0;
StartupOption = 0;
```

```
EndControllerParameter
```

```
End
```

```
##### Controller 3 #####
```

```
Begin
```

```
BeginGroup
```

```
PhysicalDevice0 = Channel=0, Target=0, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice1 = Channel=0, Target=1, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice2 = Channel=0, Target=2, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice3 = Channel=0, Target=3, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice4 = Channel=0, Target=4, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice5 = Channel=0, Target=5, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice6 = Channel=0, Target=8, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice7 = Channel=0, Target=9, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice8 = Channel=0, Target=10, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice9 = Channel=0, Target=11, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice10 = Channel=0, Target=12, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice11 = Channel=0, Target=13, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice12 = Channel=0, Target=14, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
```


PhysicalDevice49 = Channel=3, Target=9, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice50 = Channel=3, Target=10, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice51 = Channel=3, Target=11, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice52 = Channel=3, Target=12, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice53 = Channel=3, Target=13, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice54 = Channel=3, Target=14, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 PhysicalDevice55 = Channel=3, Target=15, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
 IntermediateDevice0 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
 (PhysicalDevice0, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice1, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice2, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice3, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice4, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice5, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice6, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice7, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice8, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice9, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice10, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice11, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice12, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice13, StartAddress=0mb, Size=8392mb);
 IntermediateDevice1 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
 (PhysicalDevice14, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice15, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice16, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice17, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice18, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice19, StartAddress=0mb, Size=8392mb),

(PhysicalDevice20, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice21, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice22, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice23, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice24, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice25, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice26, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice27, StartAddress=0mb, Size=8392mb);
 IntermediateDevice2 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
 (PhysicalDevice28, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice29, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice30, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice31, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice32, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice33, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice34, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice35, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice36, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice37, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice38, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice39, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice40, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice41, StartAddress=0mb, Size=8392mb);
 IntermediateDevice3 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
 (PhysicalDevice42, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice43, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice44, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice45, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice46, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice47, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice48, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice49, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice50, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice51, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice52, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice53, StartAddress=0mb, Size=8392mb),


```

        (PhysicalDevice54, StartAddress=0mb, Size=8392mb),
        (PhysicalDevice55, StartAddress=0mb, Size=8392mb);
LogicalDevice0 = StripeSize=64kb, Raid=12, WriteThrough=1, Size=469952mb, BIOSGeometry=2GB,
        (IntermediateDevice0, StartAddress=0mb, Size=117488mb),
        (IntermediateDevice1, StartAddress=0mb, Size=117488mb),
        (IntermediateDevice2, StartAddress=0mb, Size=117488mb),
        (IntermediateDevice3, StartAddress=0mb, Size=117488mb);
EndGroup
BeginControllerParameter
    ControllerName = eXtremeRAID 2000;
    ControllerType = 28;
    FirmwareVersion = 5.60;
    CacheLineSize = 8KB;
    BackgroundTaskRate = 50;
    InitiatorID = 7;
    DiskStartupMode = AutoSpin;
    DevicesPerSpin = 2;
    InitialDelay = 6S;
    SequentialDelay = 0S;
    EnableDriveSizing = 1;
    EnableClustering = 0;
    EnableBGInit = 1;
    EnableReadAhead = 0;
    EnableBiosLoadDelay = 0;
    EnableForcedUnitAccess = 0;
    DisableBios = 1;
    EnableCDROMBoot = 0;
    EnableStorageWorks = 0;
    EnableSAFTE = 1;
    EnableSES = 1;
    EnableARM = 1;
    EnableOFM = 1;
    OEMCode = 0;
    StartupOption = 0;
EndControllerParameter
End

```

```

##### Controller 4 #####
Begin
BeginGroup
    PhysicalDevice0 = Channel=0, Target=0, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice1 = Channel=0, Target=1, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice2 = Channel=0, Target=2, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice3 = Channel=0, Target=3, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice4 = Channel=0, Target=4, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice5 = Channel=0, Target=5, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice6 = Channel=0, Target=8, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice7 = Channel=0, Target=9, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice8 = Channel=0, Target=10, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice9 = Channel=0, Target=11, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice10 = Channel=0, Target=12, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice11 = Channel=0, Target=13, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice12 = Channel=0, Target=14, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice13 = Channel=0, Target=15, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice14 = Channel=1, Target=0, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice15 = Channel=1, Target=1, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
    PhysicalDevice16 = Channel=1, Target=2, Size=8392mb, State=Online,
        TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;

```


PhysicalDevice53 = Channel=3, Target=13, Size=8392mb, State=Online
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;

PhysicalDevice54 = Channel=3, Target=14, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;

PhysicalDevice55 = Channel=3, Target=15, Size=8392mb, State=Online,
 TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;

IntermediateDevice0 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
 (PhysicalDevice0, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice1, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice2, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice3, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice4, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice5, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice6, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice7, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice8, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice9, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice10, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice11, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice12, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice13, StartAddress=0mb, Size=8392mb);

IntermediateDevice1 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
 (PhysicalDevice14, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice15, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice16, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice17, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice18, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice19, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice20, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice21, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice22, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice23, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice24, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice25, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice26, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice27, StartAddress=0mb, Size=8392mb);

IntermediateDevice2 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
 (PhysicalDevice28, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice29, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice30, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice31, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice32, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice33, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice34, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice35, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice36, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice37, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice38, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice39, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice40, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice41, StartAddress=0mb, Size=8392mb);

IntermediateDevice3 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
 (PhysicalDevice42, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice43, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice44, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice45, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice46, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice47, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice48, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice49, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice50, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice51, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice52, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice53, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice54, StartAddress=0mb, Size=8392mb),
 (PhysicalDevice55, StartAddress=0mb, Size=8392mb);

LogicalDevice0 = StripeSize=64kb, Raid=12, WriteThrough=1, Size=469952mb, BIOSGeometry=2GB,
 (IntermediateDevice0, StartAddress=0mb, Size=117488mb),
 (IntermediateDevice1, StartAddress=0mb, Size=117488mb),
 (IntermediateDevice2, StartAddress=0mb, Size=117488mb),
 (IntermediateDevice3, StartAddress=0mb, Size=117488mb);

EndGroup

BeginControllerParameter

ControllerName = eXtremeRAID 2000;
ControllerType = 28;
FirmwareVersion = 5.60;
CacheLineSize = 8KB;
BackgroundTaskRate = 50;
InitiatorID = 7;
DiskStartupMode = AutoSpin;
DevicesPerSpin = 2;
InitialDelay = 6S;
SequentialDelay = 0S;
EnableDriveSizing = 1;
EnableClustering = 0;
EnableBGInit = 1;
EnableReadAhead = 0;
EnableBiosLoadDelay = 0;
EnableForcedUnitAccess = 0;
DisableBios = 1;
EnableCDROMBoot = 0;
EnableStorageWorks = 0;
EnableSAFTE = 1;
EnableSES = 1;
EnableARM = 1;
EnableOFM = 1;
OEMCode = 0;
StartupOption = 0;

EndControllerParameter

End
Controller 5 #####
Begin

BeginGroup

PhysicalDevice0 = Channel=0, Target=0, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice1 = Channel=0, Target=1, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice2 = Channel=0, Target=2, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;

PhysicalDevice3 = Channel=0, Target=3, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice4 = Channel=0, Target=4, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice5 = Channel=0, Target=5, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice6 = Channel=0, Target=8, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice7 = Channel=0, Target=9, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice8 = Channel=0, Target=10, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice9 = Channel=0, Target=11, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice10 = Channel=0, Target=12, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice11 = Channel=0, Target=13, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice12 = Channel=0, Target=14, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice13 = Channel=0, Target=15, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice14 = Channel=1, Target=0, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice15 = Channel=1, Target=1, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice16 = Channel=1, Target=2, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice17 = Channel=1, Target=3, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice18 = Channel=1, Target=4, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice19 = Channel=1, Target=5, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice20 = Channel=1, Target=8, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;

PhysicalDevice21 = Channel=1, Target=9, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice22 = Channel=1, Target=10, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice23 = Channel=1, Target=11, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice24 = Channel=1, Target=12, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice25 = Channel=1, Target=13, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice26 = Channel=1, Target=14, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice27 = Channel=1, Target=15, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice28 = Channel=2, Target=0, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice29 = Channel=2, Target=1, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice30 = Channel=2, Target=2, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice31 = Channel=2, Target=3, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice32 = Channel=2, Target=4, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice33 = Channel=2, Target=5, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice34 = Channel=2, Target=8, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice35 = Channel=2, Target=9, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice36 = Channel=2, Target=10, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice37 = Channel=2, Target=11, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice38 = Channel=2, Target=12, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;

PhysicalDevice39 = Channel=2, Target=13, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice40 = Channel=2, Target=14, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice41 = Channel=2, Target=15, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice42 = Channel=3, Target=0, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice43 = Channel=3, Target=1, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice44 = Channel=3, Target=2, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice45 = Channel=3, Target=3, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice46 = Channel=3, Target=4, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice47 = Channel=3, Target=5, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice48 = Channel=3, Target=8, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice49 = Channel=3, Target=9, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice50 = Channel=3, Target=10, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice51 = Channel=3, Target=11, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice52 = Channel=3, Target=12, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice53 = Channel=3, Target=13, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice54 = Channel=3, Target=14, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
PhysicalDevice55 = Channel=3, Target=15, Size=8392mb, State=Online,
TransferSpeed=40MHz, TransferWidth=16Bit, MaxTag=32;
IntermediateDevice0 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
(PhysicalDevice0, StartAddress=0mb, Size=8392mb),

```

(PhysicalDevice1, StartAddress=0mb, Size=8392mb),
(PhysicalDevice2, StartAddress=0mb, Size=8392mb),
(PhysicalDevice3, StartAddress=0mb, Size=8392mb),
(PhysicalDevice4, StartAddress=0mb, Size=8392mb),
(PhysicalDevice5, StartAddress=0mb, Size=8392mb),
(PhysicalDevice6, StartAddress=0mb, Size=8392mb),
(PhysicalDevice7, StartAddress=0mb, Size=8392mb),
(PhysicalDevice8, StartAddress=0mb, Size=8392mb),
(PhysicalDevice9, StartAddress=0mb, Size=8392mb),
(PhysicalDevice10, StartAddress=0mb, Size=8392mb),
(PhysicalDevice11, StartAddress=0mb, Size=8392mb),
(PhysicalDevice12, StartAddress=0mb, Size=8392mb),
(PhysicalDevice13, StartAddress=0mb, Size=8392mb);
IntermediateDevice1 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
(PhysicalDevice14, StartAddress=0mb, Size=8392mb),
(PhysicalDevice15, StartAddress=0mb, Size=8392mb),
(PhysicalDevice16, StartAddress=0mb, Size=8392mb),
(PhysicalDevice17, StartAddress=0mb, Size=8392mb),
(PhysicalDevice18, StartAddress=0mb, Size=8392mb),
(PhysicalDevice19, StartAddress=0mb, Size=8392mb),
(PhysicalDevice20, StartAddress=0mb, Size=8392mb),
(PhysicalDevice21, StartAddress=0mb, Size=8392mb),
(PhysicalDevice22, StartAddress=0mb, Size=8392mb),
(PhysicalDevice23, StartAddress=0mb, Size=8392mb),
(PhysicalDevice24, StartAddress=0mb, Size=8392mb),
(PhysicalDevice25, StartAddress=0mb, Size=8392mb),
(PhysicalDevice26, StartAddress=0mb, Size=8392mb),
(PhysicalDevice27, StartAddress=0mb, Size=8392mb);
IntermediateDevice2 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
(PhysicalDevice28, StartAddress=0mb, Size=8392mb),
(PhysicalDevice29, StartAddress=0mb, Size=8392mb),
(PhysicalDevice30, StartAddress=0mb, Size=8392mb),
(PhysicalDevice31, StartAddress=0mb, Size=8392mb),
(PhysicalDevice32, StartAddress=0mb, Size=8392mb),
(PhysicalDevice33, StartAddress=0mb, Size=8392mb),
(PhysicalDevice34, StartAddress=0mb, Size=8392mb),

```

```

(PhysicalDevice35, StartAddress=0mb, Size=8392mb),
(PhysicalDevice36, StartAddress=0mb, Size=8392mb),
(PhysicalDevice37, StartAddress=0mb, Size=8392mb),
(PhysicalDevice38, StartAddress=0mb, Size=8392mb),
(PhysicalDevice39, StartAddress=0mb, Size=8392mb),
(PhysicalDevice40, StartAddress=0mb, Size=8392mb),
(PhysicalDevice41, StartAddress=0mb, Size=8392mb);
IntermediateDevice3 = StripeSize=64kb, Raid=0, WriteThrough=1, Size=117488mb,
(PhysicalDevice42, StartAddress=0mb, Size=8392mb),
(PhysicalDevice43, StartAddress=0mb, Size=8392mb),
(PhysicalDevice44, StartAddress=0mb, Size=8392mb),
(PhysicalDevice45, StartAddress=0mb, Size=8392mb),
(PhysicalDevice46, StartAddress=0mb, Size=8392mb),
(PhysicalDevice47, StartAddress=0mb, Size=8392mb),
(PhysicalDevice48, StartAddress=0mb, Size=8392mb),
(PhysicalDevice49, StartAddress=0mb, Size=8392mb),
(PhysicalDevice50, StartAddress=0mb, Size=8392mb),
(PhysicalDevice51, StartAddress=0mb, Size=8392mb),
(PhysicalDevice52, StartAddress=0mb, Size=8392mb),
(PhysicalDevice53, StartAddress=0mb, Size=8392mb),
(PhysicalDevice54, StartAddress=0mb, Size=8392mb),
(PhysicalDevice55, StartAddress=0mb, Size=8392mb);
LogicalDevice0 = StripeSize=64kb, Raid=12, WriteThrough=1, Size=469952mb, BIOSGeometry=2GB,
(IntermediateDevice0, StartAddress=0mb, Size=117488mb),
(IntermediateDevice1, StartAddress=0mb, Size=117488mb),
(IntermediateDevice2, StartAddress=0mb, Size=117488mb),
(IntermediateDevice3, StartAddress=0mb, Size=117488mb);
EndGroup
BeginControllerParameter
ControllerName = eXtremeRAID 2000;
ControllerType = 28;
FirmwareVersion = 5.60;
CacheLineSize = 8KB;
BackgroundTaskRate = 50;
InitiatorID = 7;
DiskStartupMode = AutoSpin;

```

```

DevicesPerSpin = 2;

InitialDelay = 6S;

SequentialDelay = 0S;

EnableDriveSizing = 1;

EnableClustering = 0;

EnableBGInit = 1;

EnableReadAhead = 0;

EnableBiosLoadDelay = 0;

EnableForcedUnitAccess = 0;

DisableBios = 1;

EnableCDROMBoot = 0;

EnableStorageWorks = 0;

EnableSAFTE = 1;

EnableSES = 1;

EnableARM = 1;

EnableOFM = 1;

OEMCode = 0;

StartupOption = 0;

```

EndControllerParameter

End

-- disk(SCSI to SCSI controller) configuration

SCSI to SCSI controller #1

```

=====
view and edit Logical drives
=====
LG RAID Size(MB) Status O #LN #SB #FL NAME
P0 RAID0 86830 GOOD S 5 - 0

```

```

=====
View scsi drives
=====
Slot Chl ID Size(MB) Speed LG_DRV Status
1 0 17366 80MB 0 ON-LINE
1 1 17366 80MB 0 ON-LINE
1 2 17366 80MB 0 ON-LINE
1 3 17366 80MB 0 ON-LINE
1 4 17366 80MB 0 ON-LINE

```

SCSI to SCSI controller #2

```

=====
view and edit Logical drives
=====
LG RAID Size(MB) Status O #LN #SB #FL NAME
P0 RAID0 86830 GOOD S 5 - 0

```

```

=====
View scsi drives
=====
Slot Chl ID Size(MB) Speed LG_DRV Status
1 0 17366 80MB 0 ON-LINE
1 1 17366 80MB 0 ON-LINE
1 2 17366 80MB 0 ON-LINE
1 3 17366 80MB 0 ON-LINE
1 4 17366 80MB 0 ON-LINE

```

Appendix D : Space Calculation

180 Day Space

Note : Numbers are in KBytes unless otherwise specified

Warehouses	2520	tpmC	31764.37	tpmC/W	12.60	
Table	Rows	Data	Index	5% Space	8H Space	Total Space
Warehouse	2,520	272	48	16	336	
District	25,200	2,808	48	143	2,999	
Item	100,000	9,528	64	221	9,813	
New-order	22,680,000	358,584	840		50,400	409,824
History	75,600,000	4,200,008	80		710,794	4,910,882
Orders	75,600,000	2,317,248	1,053,760		570,487	3,941,495
Customer	75,600,000	54,981,824	3,278,576	1,339,989		59,600,389
Order-line	756,002,500	47,250,160	100,032		8,013,225	55,363,417
Stock	252,000,000	80,640,008	150,736	1,858,187		82,648,931
Totals		189,760,440	4,584,184	3,198,556	9,344,906	206,888,086

DB File Group	Count	Size	Needed	Overhead	Not Needed
MSSQL70_misc_fg	5	84,480,000	65,285,153	652,852	18,541,995
MSSQL70_cs_fg	5	153,600,000	143,671,814	1,436,718	8,491,468
Totals		238,080,000	208,956,967	2,089,570	27,033,463

Dynamic space 52,315,696 Sum of Data for Order, Order-Line and History (excluding free extents)

Static space 147,317,054 Data + Index + 5% Space + Overhead - Dynamic space

Free space 11,413,787 Total Seg. Size - Dynamic Space - Static Space - Not Needed

Daily growth 10,550,953 (Dynamic space/W * 62.5)* tpmC

Daily spread -4,412,642 Free space - 1.5 * Daily growth (zero if negative)

180 day (KB) 2,046,488,616 Static space + 180 (daily growth + daily spread)

180 day (GB) 1951.68 Excludes OS, Paging and RDBMS Logs

Log size (MB) 30000 Total size of log file

% Log used 64.01 % of log file used during entire run

Total N-O Txn 3840684 Total count of N-O transactions during entire run

Log per N-O txn 5.12 Number of Kbytes per New-Order transaction

8 Hour Log (GB) 74.45 need double for mirroring

os, file sys, swap 8.00

Database, Sys	Disks	Qty	Total	Needed	Extra
	8,195	280	2294.60	1,959.68	343.38
	8,463	1	8.46		
Mirrored Log	16,958	10	169.58	148.90	20.68

Appendix E : Price Quotation

Microsoft

June 15, 2000

Mr. Eiichi Kennai
 NEC Corp.
 via FAX

Dear Mr. Kennai:

Here is the information you requested regarding U.S. pricing for several Microsoft products, to be used in conjunction with TPC-C benchmark testing.

Part Number	Description	Unit Price	Quantity	Price
N/A	SQL Server 2000 Enterprise Edition Per processor licensing	\$19,999	4	\$79,996
C10-00010	Windows 2000 Advanced Server 25 Client Licenses	\$3,919	1	\$3,919
C11-00016	Windows 2000 Server 5 Client Licenses	\$999	1	\$999
048-00317	Visual C++ Professional 6.0 Win32	\$549	1	\$549
	5-year maintenance for above software	\$2,095	5	\$10,475

Some products may not be currently orderable but will be available through Microsoft's normal distribution channels by August 1, 2000.

This quote is valid for the next 90 days.

If I can be of any further assistance, please contact me at (425) 703-3455 or barry@microsoft.com.

Yours truly,



Barry Goffe
 Product Manager
 SQL Server Marketing

Microsoft Corporation is an equal opportunity employer.

MYLEX
 MOVING DATA - SMARTER



June 9th, 2000

NEC Corporation
 Server Hardware Engineering Group

Attn: Mr. Robin Gomi

RE: Quotation for eXtremeRAID2000

Dear Mr. Gomi;

Mylex is please to submit the following quotation to you for eXtremeRAID2000.

P/N	Description	Unit Price
E2000-4-64NB	4 chnl Ultra 160 SCSI RAID controller 64MB without BBU	US\$2,050-

Note:

1. Above price is valid for 60 days
2. Based on F.O.B. ex-factory Fremont, CA
3. Warranty period 5 years
4. Lead time: 30 days

Sincerely

Ben Taniguchi
 Vice President
 Asia/Pacific Sales

Mylex Corporation
 34551 Ardenwood Blvd.
 Fremont, CA 94555-3607
 Phone: 510-796-8100
www.mylex.com

Applied Digital Systems, Inc.
 30 Lfbridge Lane East
 Fairport, New York 14450



phone: 716.377.7000
 fax: 716.377.5544
 web: www.adsys.com

30 Lfbridge Lane East
 Fairport, NY 14450
 Phone: 716.377.7000
 Fax: 716.377.5544
 E-Mail: heather@adsys.com

Quotation: 0601
 Date: June 1, 2000
 Terms: N30
 FOB: Fairport, NY
 Quote Valid For 30 Days

TO:

Robin Gomi
 NEC
 978-635-6267 fax

QTY	DESCRIPTION	UNIT PRICE
04	IFT-3102UG:SCSI U2 4 channel controller.	\$2092.00each
04	IFT-9070B: Integrated Batt. Module.	\$ 170.00each

Thank you for the opportunity to work with you. If you have any questions please feel free to call.

Sincerely,

Heather Visconti

Seneca Data Distributors, Inc

QUOTATION

PAGE# 1

401 ROUND POND ROAD
 NORTH SYRACUSE, NY 13212 (315)433-1160

TERMS: COD CASH

-----BILL TO-----

-----SHIP TO-----

NEC TECHNOLOGIES, INCORPORATION
 1300 MASSACHUSETTS AVENUE
 BOXBOUROUGH, MA 01719
 (508)264-8974

NEC TECHNOLOGIES, INCORPORATION
 1300 MASSACHUSETTS AVENUE
 BOXBOUROUGH, MA 01719
 (508)264-8974

OUR ORDER#: 838638 JSD ACCOUNT#: 008873
 ORDER DATE: 06/13/00 13:55:10 YOUR P/D#: quote
 F. O. B. : DRIGIN PLACED BY: ROBIN
 SHIP VIA: UPSA GROUND ALBANY NY CONTRACT#:
 SHIP DATE: 06/13/00 JOB#/NAME:

QUANTITY	UNIT	ITEM NUMBER	UNIT PRICE	EXT PRICE
	EA	TP1016C COMPEX, 16PORT, 10BASET, RJ45&BNC/AUI, RACKMOUNT, CASCADABLE ETHERNET, MICROHUB	80.0000	80.00

1 LINE

 SUBTOTAL 80.00
 TAX 4.00
 TOTAL 84.00