NEC

TPC Benchmark™ C
Full Disclosure Report

NEC Express5800/1320Xc (32 SMP)

Using Microsoft® Windows® Server 2003, Datacenter Edition
for 64-bit Itanium-based Systems
and
Microsoft® SQL Server™ 2000, Enterprise Edition (64-bit)

First Edition
Submitted for Review
April 23, 2003

NEC, the Sponsor of this benchmark test, believes that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. 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 does not warrant or represent that a user can or will achieve similar performance expressed in transactions per minute (tpmC) or normalized price/performance ($/tpmC). No warranty of system performance or price/performance is expressed or implied in this report.

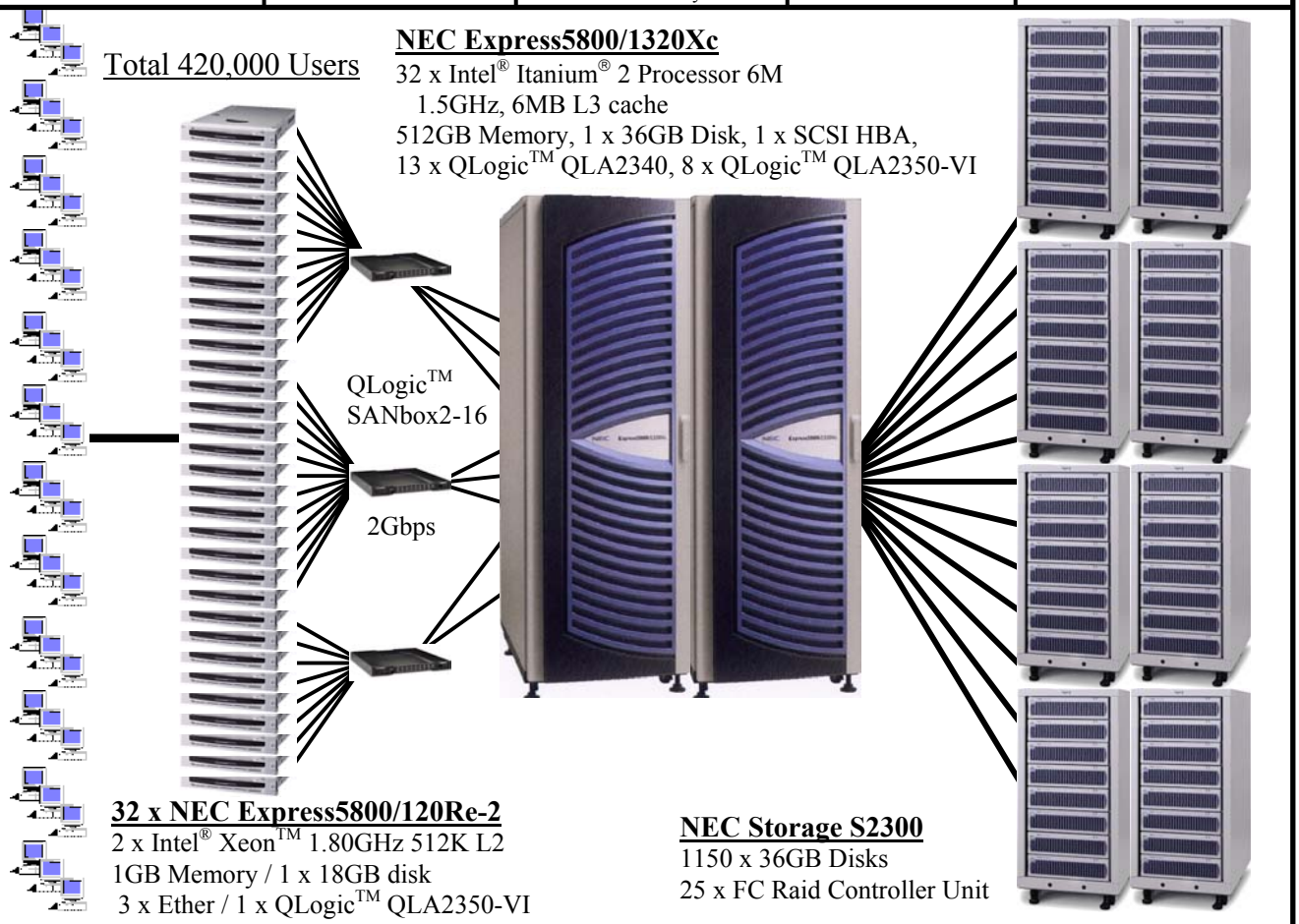
Printed in USA, 2003

NEC Express5800/1320Xc
C/S with Express5800/120Re-2

TPC-C Rev.5.1
Reported Date
April 23, 2003

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$5,907,028	514,034.72 tpmC	$11.50 per tpmC	October 22, 2003 Hardware Available on Aug 11, 2003

Processors	Database Manager	Operating System	Other Software	Number of Users
32 Intel® Itanium® 2 Processor 6M 1.5GHz for Server 62 Intel® Xeon™ 1.80GHz for Client	Microsoft® SQL Server™ 2000 Enterprise Edition(64-bit)	Microsoft® Windows® Server 2003, Datacenter Edition for 64-bit Itanium-based Systems	Windows® 2000 Server Microsoft® VC++ Microsoft® COM+	420,000



System Component	Server		Each Client	
Processors **Cache**	32	Intel® Itanium® 2 Processor 6M 1.5GHz, 6MB L3 Cache	2	Intel® Xeon™ 1.80GHz 512KB L2 Cache
Memory		2 GB x 256		512MB x 2
Disk Controllers	13 1	QLogic™ QLA2340 SCSI HBA	1	On-board SCSI
Disk Drives	1 1150	36GB (33.4GB Usable) 36GB (33.2GB Usable)	1	18GB
Total Storage		41,436GB		18GB
Others	1 1 8	DVD-ROM Drive 1Gbps Ether NIC QLogic™ QLA2350-VI 2Gb NIC	1 2 1 1	CD-ROM Drive On-board Ether controller 100Mbps Ether NIC QLogic™ QLA2350-VI 2Gb NIC

NEC

NEC Express5800/1320Xc
C/S with Express5800/120Re-2

TPC-C REV 5.1
Report Date: 04/23/03

Description	Part Number	Third Party Brand	Pricing	Unit Price	Qty	Extended Price	3-yr Mnt. Price
Server Hardware							
Express5800/1320Xc system	850200023	NEC	1	2,126,090	1	2,126,090	1,158,910
Base Original 32	Included	NEC	1	-	1	-	-
Cell Card Original	Included	NEC	1	-	8	-	-
CPU(1.5GHz/6MB)	Included	NEC	1	-	32	-	-
Memory(8GB)	Included	NEC	1	-	64	-	-
Base PCI Box	Included	NEC	1	-	3	-	-
Core PCI Box(Windows Model)	Included	NEC	1	-	1	-	-
PCI Cable(2.5m)	Included	NEC	1	-	6	-	-
PCI Cable(3.0m)	Included	NEC	1	-	2	-	-
Expansion Cabinet	Included	NEC	1	-	1	-	-
18GB 10K rpm HDD	Included	NEC	1	-	1	-	-
Gbit Ether NIC	Included	NEC	1	-	1	-	-
Mouse/Keyboard	Included	NEC	1	-	1	-	-
SCSI HBA with VHDCI cables	Included	NEC	1	-	1	-	-
Windows Server 2003, Datacenter Edition	Included	NEC	1	-	1	-	-
NEC Express5800/120Re-2 (for System Maintenance)							
Base System with 1 x Xeon Processor 1.8GHz/512KB	850162001	NEC	1	2,859	1	2,859	-
1 x Xeon Processor 1.8GHz/512KB BTO Option,	062-02163-000	NEC	1	399	1	399	-
1GB(2 x 512MB) DDR200 SDRAM memory,	062-02168-000	NEC	1	599	1	599	-
1 x 18.2GB 10K rpm Ultra160 HDD,	062-02011-000	NEC	1	329	1	329	-
CD-ROM, 2 x On-board LAN, KB/MS	Included	NEC	1	-	1	-	-
3 years of warranty service to 4-hour response, 7x24	WG-0000-1095-7244	NEC	1	1,399	1	-	1,399
NEC AccuSync50 (15" monitor)	AS50	NEC	3	120	2	239	-
36GB 10K rpm SCSI HDD (+2 spares)	DK32DJ-36MC	HITACHI	3	239	3	717	
FC HBA QLA2340 (+2 spares)	QLA2340-CK	QLogic	3	1,329	15	19,940	-
FC Adapter QLA2350 (+2 spares)	QLA2350	QLogic	4	2,095	11	23,045	-
FC Cable 10M LC-LC (+2 spares)	062-02304-000	NEC	1	200	11	2,200	-
					Subtotal	2,176,417	1,160,309
Disk Subsystem							
NEC Storage S2300 Basic Unit 2 RAID_Cntrl, 1GB Cache, 1DE, 3 x 36GB Drive	850161002	NEC	1	46,296	25	1,157,400	-
1GB(2 x 512MB) PC133 SDRAM memory	062-02301-000	NEC	1	1,999	2	3,998	-
S2300 DE (+10% spares)	062-02302-000	NEC	1	6,999	58	405,942	-
36GB 15K rpm FC HDD (+10% spares)	062-02303-000	NEC	1	1,099	1183	1,300,117	-
3 yr Telephone Upgrade Hdw & Storage Manager SW Support; 7days/ 24hours per day	IS-THSU-1095-0724	NEC	1	1499	25		37,475
42U Rackframe	050-01790-000	NEC	1	1,799	8	14,392	-
FC Cable 10M LC-LC (+10% spares)	062-02304-000	NEC	1	200	28	5,600	-
					Subtotal	2,887,449	37,475
Server Software							
SQL Server2000 Ent. Edition(64-bit) ,Processor License	810-00560	Microsoft	2	16,541	32	529,312	5,850
					Subtotal	529,312	5,850
Client Hardware							
NEC Express5800/120Re-2							
Base System with 1 x Xeon Processor 1.8GHz/512KB	850162001	NEC	1	2,859	31	88,629	-
1 x Xeon Processor 1.8GHz/512KB BTO Option,	062-02163-000	NEC	1	399	31	12,369	-
1GB(2 x 512MB) DDR200 SDRAM memory,	062-02168-000	NEC	1	599	31	18,569	-
1 x 18.2GB 10K rpm Ultra160 HDD,	062-02011-000	NEC	1	329	31	10,199	-
CD-ROM, 2 x On-board LAN, KB/MS	Included	NEC	1	-	31	-	-
3 years of warranty service to 4-hour response, 7x24	WG-0000-1095-7244	NEC	1	1,399	31	-	43,369
NEC AccuSync50 (15" monitor)	AS50	NEC	3	120	31	3,710	-
42U Rackframe	050-01790-000	NEC	1	1,799	2	3,598	-
FC Adapter QLA2350 (+10% spares)	QLA2350	QLogic	4	2,095	35	73,325	-
FC Cable 10M LC-LC (+10% spares)	062-02304-000	NEC	1	200	35	7,000	-
Intel PRO/100 S Server Adapter, 5-pack (+10% spares)	PILA8470C3PAK5	Intel	3	286	7	2,002	-
Fast Ether Cable 25' RJ45-RJ45 (+10% spares)	C5E-114GY-25FB	AESP	3	11	103	1,133	-
					Subtotal	220,535	43,369
Client Software							
Windows2000 Server, Server License	C11-00821	Microsoft	2	738	32	23,616	(Included)
Visual C++ Standard	254-00170	Microsoft	2	109	1	109	(Included)
					Subtotal	23,725	
User Connectivity							
SAN BOX2-16 16-ports 2-Gbit FC Switch(+2 spares)	SB2A-16A	QLogic	4	17,995	5	89,975	-
SFP-Shortwave LC Optic / 8 SFPs per pack (+10% spares)	SFP2-SW-01	QLogic	4	2,150	6	12,900	-
					Subtotal	102,875	0
					TOTAL	5,940,313	1,247,003
NEC brand total(Pricing 1-NEC)						5,160,289	1,241,153
NEC brand Large Purchase Cash Prepay Discount(-20%)						-1,032,058	-248,231

Notes:
Client software maintenance is covered by the maintenance costs of Microsoft SQL Server 2000
Pricing: 1-NEC 2-Microsoft 3-CDW 4-QLogic

3-Yr. Cost of Ownership: **$5,907,028**
tpmC Rating: **514034.72**

$ / tpmC: $11.50

Results and methodology audited by Francois Raab of InfoSizing, Inc. (www.sizing.com)

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.

Numerical Quantities Summary

MQTh, Computed Maximum Qualified Throughput 514,034.72 tpmC

Response Times(in seconds)	Average	Maximum	90%
New-Order	0.89	21.45	1.73
Payment	0.86	24.88	1.68
Delivery(interactive portion)	0.10	2.52	0.11
Stock-Level	0.94	22.06	1.82
Order-status	0.88	22.59	1.75
Delivery(deferred portion)	0.10	3.20	0.14
Menu	0.10	2.53	0.11

Response time delay added for emulated components 0.1

Transaction Mix , in percent of total transaction

New-Order	44.91%
Payment	43.03%
Delivery	4.01%
Stock-Level	4.01%
Order-Status	4.03%

Keying/Think Times (in seconds)	Average		Min		Max	
New-Order	18.02	12.07	18.00	0.00	18.06	120.71
Payment	3.02	12.07	3.00	0.00	3.06	120.71
Delivery	2.02	5.08	2.00	0.00	2.06	50.71
Stock-Level	2.02	5.07	2.00	0.00	2.06	50.71
Order-Status	2.02	10.08	2.00	0.00	2.06	100.71

Test Duration

Ramp-up time	145 minutes
Measurement interval	120 minutes
Number of checkpoints	4
Checkpoint interval	30 minutes
Number of transactions (all types) completed in measurement interval	142,855,735

Abstract

This report documents the compliance of NEC Corporation's TPC Benchmark™ C tests on the NEC Express5800/1320Xc client/server system with version 5.1 of the TPC Benchmark C Standard Specification. 31 Clients (NEC Express5800/120Re-2) were used as the front-end clients.

The operating system and the DBMS used on the server were Microsoft® Windows® Server 2003, Datacenter Edition and Microsoft® SQL Server™ 2000, Enterprise Edition(64-bit). The operating system on the clients was Microsoft® Windows® 2000 Server SP2. Those clients ran Microsoft® 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 (three year capital cost per measured tpmC) are reported.

System	SW	Total System Cost	tpmC	$ per tpmC	Availability Date
NEC Express5800 /1320Xc	Microsoft® Windows® Server 2003, Datacenter Edition for 64-bit Itanium-based Systems Microsoft® SQL Server™ 2000, Enterprise Edition(64-bit)	$5,907,028	514,034.72	$11.50	October 22, 2003 Hardware Available on August 11, 2003

Standard and Executive Summary Statements

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

Auditor

The benchmark configuration, environment and methodology were audited by Francois Raab of Info Sizing Inc. to verify compliance with the relevant TPC specifications.

Preface

The TPC Benchmark™ C was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry. This full disclosure report is based on the TPC Benchmark™ C Standard Specifications Version 5.1.

TPC Benchmark™ C Overview

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

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

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

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

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 Sponsor's Full Disclosure report must correspond with the order and titles of sections from the TPC-C standard specification (i.e., this document). 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 report.

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,*
- *longest checkpoint interval in minutes,*
- *number of transactions (all types) completed within the measurement interval,*
- *computed Maximum Qualified Throughput in tpmC,*
- *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,*
- *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 Express5800/1320Xc. Price quotations contained in Appendix E correspond to the NEC Express5800/1320Xc server.

Parameters and Options

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

- *Database tuning options.*
- *Recovery/commit options.*
- *Consistency/locking options.*
- *Operating system and application configuration parameters.*
- *Compilation and linkage options and run-time optimizations used to create/install applications, OS, and/or databases.*

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

Configuration Diagrams

Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

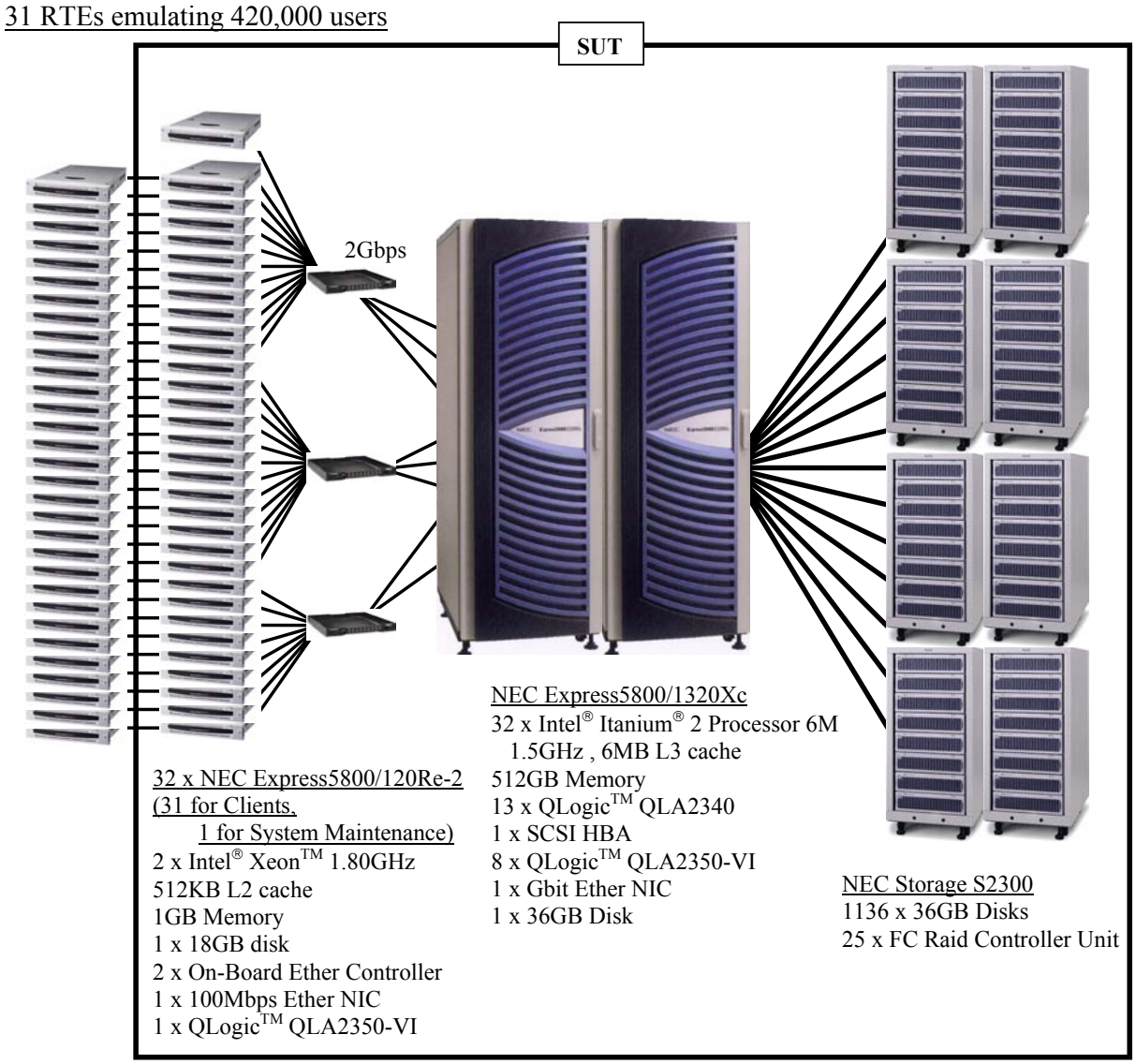
- *Number and type of processors*
- *Size of allocated memory, and any specific mapping/partitioning of memory unique to the test.*
- *Number and type of disk units (and controllers, if applicable).*
- *Number of channels or bus connections to disk units, including their protocol type.*
- *Number of LAN (e.g., Ethernet) connections, including routers, workstations, terminals, etc., that were physically used in the test or are incorporated into the pricing structure (see Clause 8.1.8).*
- *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.

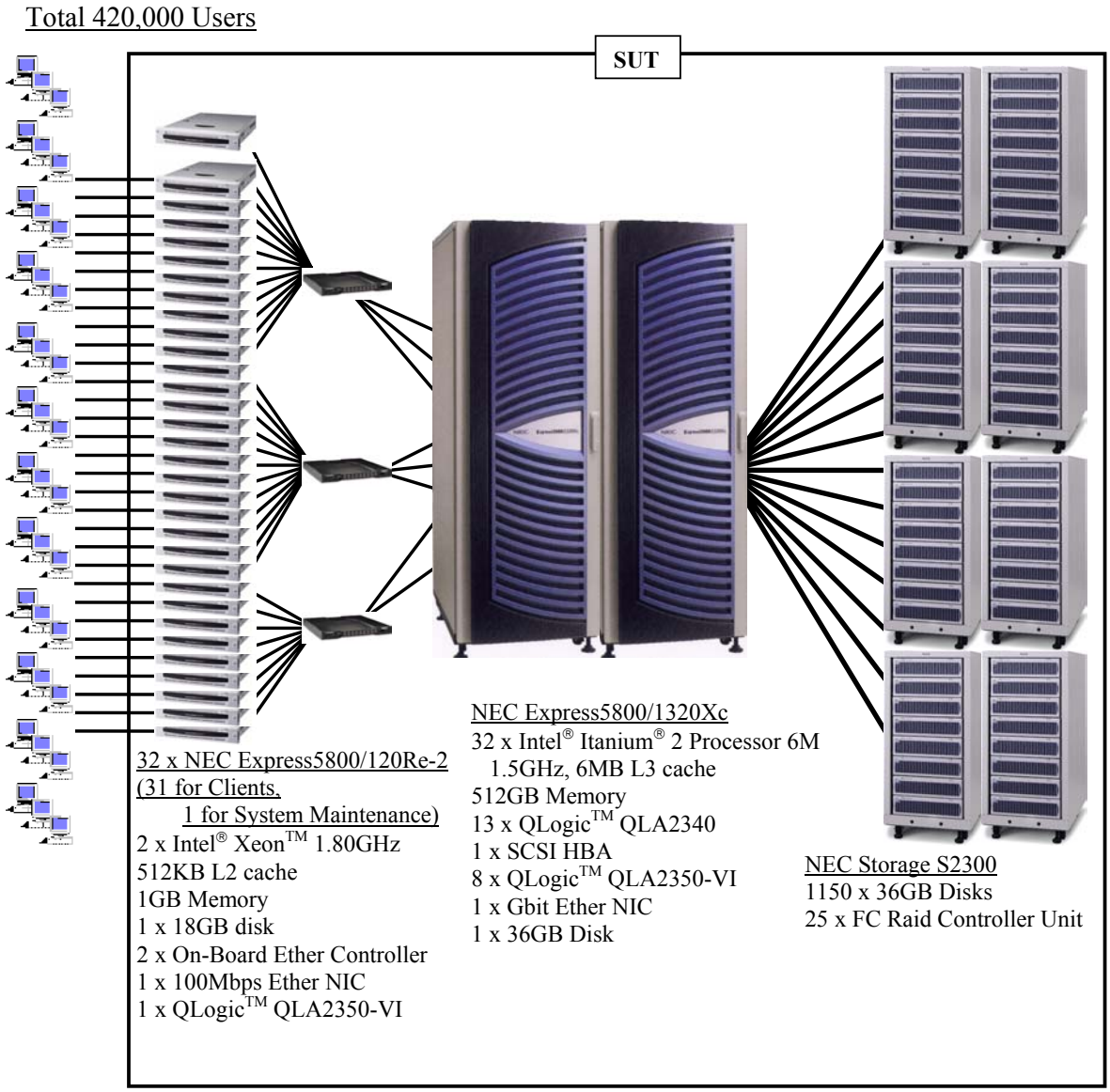
Figure 1.1 Express5800/1320Xc, Measured Configuration Diagram



Priced System Configuration

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

Figure1.2: Express5800/1320Xc, Priced Configuration Diagram



Clause 1 : Logical Database Design and Related Items

Table Definitions

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

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

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 restriction in the SUT database implementation that precludes inserts beyond the limits defined in Clause 1.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the maximum key value for these new rows

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 (see Clause 1.4.6).

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 (see Clause 1.4.7).

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 layouts 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.

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 transactions must be disclosed.*
- *The percentage of Payment and Order-Status transactions that used non-primary key (C_LAST) access to the database must be disclosed.*
- *The percentage of Delivery transactions that were skipped as a result of an insufficient number of rows in the NEW-ORDER table must be disclosed.*

Table 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 Clause 8.1.3.5 through 8.1.3.11

Table 1 Transaction Statistics

	Statistic	Value
New Order	Home warehouse order lines	**99.00%**
	Remote warehouse order lines	**1.00%**
	Rolled back transactions	**1.00%**
	Average items per order	**10.00**
Payment	Home warehouse payments	**85.00%**
	Remote warehouse payments	**15.00%**
	Accessed by last name	**60.01%**
Order Status	Accessed by last name	**60.05%**
Delivery	Skipped deliveries	**0**
Transaction Mix	New Order	**44.91%**
	Payment	**43.03%**
	Delivery	**4.01%**
	Stock Level	**4.01%**
	Order Status	**4.03%**

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 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 transaction for a 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 a 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 the 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 20 minutes under 210,000 users (21,000 active warehouse) condition. A checkpoint generated in the test. The shell script of consistency was executed before and after the run. The result of the same queries verified that the database remained consistent after the run.

Isolation Tests

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, 8 and 9.

Durability Tests

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

- *Permanent irrecoverable failure of any single durable medium during the Measurement Interval containing TPC-C database tables or recovery log data.*
- *Instantaneous interruption (system or subsystem crash/system hang) in processing which causes all or part of the processing of atomic transactions to halt..*

- *Failure of all or part of memory(loss of contents)*

Loss of Memory and 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.

The following steps were performed on a database of 42,000 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. 420,000 users were logged in to the database and start transactions.
3. Waited for all emulated users to be activated and the number of tpmC exceed 90% of reported tpmC.
4. Removed one of mirrored log disk. The running continued without any interruptions.
5. Keep running more 5 minutes.
6. The system was powered off.
7. The RTE was shut down.
8. The system was powered up. SQL Server™ 2000 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 4,400 Warehouse database. The standard driving mechanism was used to generate the transaction load of 44,000 users for the test. To demonstrate recovery from a permanent failure of durable media containing TPC-C tables, the following steps were performed.

1. A 4,400 Warehouse database was built having similar characteristics to the large database.
2. The database was backed up using SQL Server™ 2000 backup facilities.
3. A sum of D_NEXT_O_ID was taken.
4. 44,000 users were logged in to the database and kept running transactions about 5 minutes in steady state.
5. One disk drive for data part in the array was removed causing SQL Server™ 2000 error. Shut down SQL Server™ 2000.
6. SQL Server™ 2000 was restarted and a dump of the transaction log was taken.
7. The 4,400 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.

Loss of mirrored write-back cache

The Fibre Array system used for this benchmark has integrated feature of mirrored write-back cache. When a LUN is configured to enable write-back caching, the data on the cache is automatically mirrored on the RAMs in two controller modules, which are powered and protected from loss of power by the controller BBU. Loss of write-back cache was demonstrated on 4,400 warehouse database, by pulling off one of the controller module. A fully scaled database would also pass this test.

1. A sum of D_NEXT_O_ID of all rows in the district table was taken.
2. 44,000 users were logged in to the database and kept running transactions about 5 minutes in steady state.
3. A controller module, which manages write-back cache of mirrored drives, was pulled off.
4. Fibre system reported system error resulted in IO error for SQL database.
5. Contents on mirrored cache on another controller were automatically saved to temporal disk area and the Fibre Array house was eventually stopped. System was shut down.
6. The Fibre array was re-powered.
7. Saved data was automatically restored to mirrored cache and flushed to drives.
8. SQL Server™ 2000 was started up and database was recovered automatically.
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., the number of rows) of each table, as it existed at the start of the benchmark run (see Clause 4.2), must be disclosed. If the database was over-scaled and inactive rows of the WAREHOUSE table were deleted (see Clause 4.2.2), the cardinality of the WAREHOUSE table as initially configured and the number of rows deleted must be disclosed.

The TPC-C database was originally built with 44,000 warehouses.

Table 2 Number of Rows for Server

Table	Cardinality as benchmarked
Warehouse	44,000
District	440,000
Customer	1,320,000,000
History	1,320,000,000
Orders	1,320,000,000
New Order	396,000,000
Order Line	13,199,956,988
Stock	4,400,000,000
Item	100,000
Deleted Warehouse Rows	2,000

2,000 warehouses were deleted before the measurement. 42,000 warehouses and their associated tables were accessed during the measurement.

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 the tested and priced systems.

Table 3 depicts the distribution of the database over the disks of the tested and priced system.
Figure 1.1, 1.2 shows the disk configuration for measured and priced system.

Table 3 : Data Distribution

HBA#	storage system#	SP#	DEU#	# of Disks	RAID Level	Capacity (GB)	Partition1	Partition2	Partition3	Partition4
Partitions for DB Log							Logfiles (raw)	Junction point (NTFS)		
0	0	0	0	14	1/0	232.82	z:\dev\log (200GB)	z: (32.81GB)		
			1	14	1/0	232.82	z:\dev\log2 (200GB)	unused (32.81GB)		
			2	14	1/0	232.82	z:\dev\log3 (200GB)	unused (32.81GB)		
			3	14	1/0	232.82	z:\dev\log4 (200GB)	unused (32.81GB)		

Partitions for DB Data							misc fg (raw)	cs fg (raw)	freespace	backup file (NTFS)
1	1	0	0	15	0	498.92	z:\dev\m001 (20GB)	z:\dev\c001 (30GB)	(30GB)	z:\dev\b001 (418.91GB)
			1	15	0	498.92	z:\dev\m002 (20GB)	z:\dev\c002 (30GB)	(30GB)	z:\dev\b002 (418.91GB)
			2	15	0	498.92	z:\dev\m003 (20GB)	z:\dev\c003 (30GB)	-	z:\dev\b003 (443.92GB)
	2	0	0	15	0	498.92	z:\dev\m005 (20GB)	z:\dev\c005 (30GB)	(30GB)	z:\dev\b005 (418.91GB)
			1	15	0	498.92	z:\dev\m006 (20GB)	z:\dev\c006 (30GB)	(30GB)	z:\dev\b006 (418.91GB)
			2	15	0	498.92	z:\dev\m007 (20GB)	z:\dev\c007 (30GB)	(30GB)	z:\dev\b007 (418.91GB)
2	3	0	0	15	0	498.92	z:\dev\m009 (20GB)	z:\dev\c009 (30GB)	(30GB)	z:\dev\b009 (418.91GB)
			1	15	0	498.92	z:\dev\m010 (20GB)	z:\dev\c010 (30GB)	(30GB)	z:\dev\b010 (418.91GB)
			2	15	0	498.92	z:\dev\m011 (20GB)	z:\dev\c011 (30GB)	(30GB)	z:\dev\b011 (418.91GB)
	4	0	0	15	0	498.92	z:\dev\m013 (20GB)	z:\dev\c013 (30GB)	(30GB)	z:\dev\b013 (418.91GB)
			1	15	0	498.92	z:\dev\m014 (20GB)	z:\dev\c014 (30GB)	-	z:\dev\b014 (443.92GB)
			2	15	0	498.92	z:\dev\m015 (20GB)	z:\dev\c015 (30GB)	(30GB)	z:\dev\b015 (418.91GB)
3	5	0	0	15	0	498.92	z:\dev\m017 (20GB)	z:\dev\c017 (30GB)	(30GB)	z:\dev\b017 (418.91GB)
			1	15	0	498.92	z:\dev\m018 (20GB)	z:\dev\c018 (30GB)	(30GB)	z:\dev\b018 (418.91GB)
			2	15	0	498.92	z:\dev\m019 (20GB)	z:\dev\c019 (30GB)	(30GB)	z:\dev\b019 (418.91GB)
	6	0	0	15	0	498.92	z:\dev\m021 (20GB)	z:\dev\c021 (30GB)	(30GB)	z:\dev\b021 (418.91GB)
			1	15	0	498.92	z:\dev\m022 (20GB)	z:\dev\c022 (30GB)	(30GB)	z:\dev\b022 (418.91GB)
			2	15	0	498.92	z:\dev\m023 (20GB)	z:\dev\c023 (30GB)	(30GB)	z:\dev\b023 (418.91GB)
4	7	0	0	15	0	498.92	z:\dev\m025 (20GB)	z:\dev\c025 (30GB)	-	z:\dev\b025 (443.92GB)
			1	15	0	498.92	z:\dev\m026 (20GB)	z:\dev\c026 (30GB)	(30GB)	z:\dev\b026 (418.91GB)
			2	15	0	498.92	z:\dev\m027 (20GB)	z:\dev\c027 (30GB)	-	z:\dev\b027 (443.92GB)
	8	0	0	15	0	498.92	z:\dev\m029 (20GB)	z:\dev\c029 (30GB)	(30GB)	z:\dev\b029 (418.91GB)
			1	15	0	498.92	z:\dev\m030 (20GB)	z:\dev\c030 (30GB)	(30GB)	z:\dev\b030 (418.91GB)
			2	15	0	498.92	z:\dev\m031 (20GB)	z:\dev\c031 (30GB)	(30GB)	z:\dev\b031 (418.91GB)
5	9	0	0	15	0	498.92	z:\dev\m033 (20GB)	z:\dev\c033 (30GB)	(30GB)	z:\dev\b033 (418.91GB)
			1	15	0	498.92	z:\dev\m034 (20GB)	z:\dev\c034 (30GB)	(30GB)	z:\dev\b034 (418.91GB)
			2	15	0	498.92	z:\dev\m035 (20GB)	z:\dev\c035 (30GB)	-	z:\dev\b035 (443.92GB)
	10	0	0	15	0	498.92	z:\dev\m037 (20GB)	z:\dev\c037 (30GB)	(30GB)	z:\dev\b037 (418.91GB)
			1	15	0	498.92	z:\dev\m038 (20GB)	z:\dev\c038 (30GB)	(30GB)	z:\dev\b038 (418.91GB)
			2	15	0	498.92	z:\dev\m039 (20GB)	z:\dev\c039 (30GB)	-	z:\dev\b039 (443.92GB)
6	11	0	0	15	0	498.92	z:\dev\m041 (20GB)	z:\dev\c041 (30GB)	-	z:\dev\b041 (443.92GB)
			1	15	0	498.92	z:\dev\m042 (20GB)	z:\dev\c042 (30GB)	(30GB)	z:\dev\b042 (418.91GB)
			2	15	0	498.92	z:\dev\m043 (20GB)	z:\dev\c043 (30GB)	(30GB)	z:\dev\b043 (418.91GB)
	12	0	0	15	0	498.92	z:\dev\m045 (20GB)	z:\dev\c045 (30GB)	(30GB)	z:\dev\b045 (418.91GB)
			1	15	0	498.92	z:\dev\m046 (20GB)	z:\dev\c046 (30GB)	-	z:\dev\b046 (443.92GB)
			2	15	0	498.92	z:\dev\m047 (20GB)	z:\dev\c047 (30GB)	-	z:\dev\b047 (443.92GB)
7	13	0	0	15	0	498.92	z:\dev\m049 (20GB)	z:\dev\c049 (30GB)	(30GB)	z:\dev\b049 (418.91GB)
			1	15	0	498.92	z:\dev\m050 (20GB)	z:\dev\c050 (30GB)	(30GB)	z:\dev\b050 (418.91GB)
			2	15	0	498.92	z:\dev\m051 (20GB)	z:\dev\c051 (30GB)	(30GB)	z:\dev\b051 (418.91GB)
	14	0	0	15	0	498.92	z:\dev\m053 (20GB)	z:\dev\c053 (30GB)	(30GB)	z:\dev\b053 (418.91GB)
			1	15	0	498.92	z:\dev\m054 (20GB)	z:\dev\c054 (30GB)	(30GB)	z:\dev\b054 (418.91GB)
			2	15	0	498.92	z:\dev\m055 (20GB)	z:\dev\c055 (30GB)	(30GB)	z:\dev\b055 (418.91GB)

8	15	0	0	15	0	498.92	z:\dev\m057 (20GB)	z:\dev\c057 (30GB)	(30GB)	z:\dev\b057 (418.91GB)
			1	15	0	498.92	z:\dev\m058 (20GB)	z:\dev\c058 (30GB)	(30GB)	z:\dev\b058 (418.91GB)
			2	15	0	498.92	z:\dev\m059 (20GB)	z:\dev\c059 (30GB)	(30GB)	z:\dev\b059 (418.91GB)
	16	0	0	15	0	498.92	z:\dev\m061 (20GB)	z:\dev\c061 (30GB)	(30GB)	z:\dev\b061 (418.91GB)
			1	15	0	498.92	z:\dev\m062 (20GB)	z:\dev\c062 (30GB)	(30GB)	z:\dev\b062 (418.91GB)
			2	15	0	498.92	z:\dev\m063 (20GB)	z:\dev\c063 (30GB)	(30GB)	z:\dev\b063 (418.91GB)
9	17	0	0	15	0	498.92	z:\dev\m065 (20GB)	z:\dev\c065 (30GB)	(30GB)	z:\dev\b065 (418.91GB)
			1	15	0	498.92	z:\dev\m066 (20GB)	z:\dev\c066 (30GB)	(30GB)	z:\dev\b066 (418.91GB)
			2	15	0	498.92	z:\dev\m067 (20GB)	z:\dev\c067 (30GB)	(30GB)	z:\dev\b067 (418.91GB)
	18	0	0	15	0	498.92	z:\dev\m069 (20GB)	z:\dev\c069 (30GB)	(30GB)	z:\dev\b069 (418.91GB)
			1	15	0	498.92	z:\dev\m070 (20GB)	z:\dev\c070 (30GB)	(30GB)	z:\dev\b070 (418.91GB)
			2	15	0	498.92	z:\dev\m071 (20GB)	z:\dev\c071 (30GB)	(30GB)	z:\dev\b071 (418.91GB)
10	19	0	0	15	0	498.92	z:\dev\m004 (20GB)	z:\dev\c004 (30GB)	-	z:\dev\b004 (443.92GB)
			1	15	0	498.92	z:\dev\m008 (20GB)	z:\dev\c008 (30GB)	-	z:\dev\b008 (443.92GB)
			2	15	0	498.92	z:\dev\m012 (20GB)	z:\dev\c012 (30GB)	-	z:\dev\b012 (443.92GB)
	20	0	0	15	0	498.92	z:\dev\m016 (20GB)	z:\dev\c016 (30GB)	-	z:\dev\b016 (443.92GB)
			1	15	0	498.92	z:\dev\m020 (20GB)	z:\dev\c020 (30GB)	-	z:\dev\b020 (443.92GB)
			2	15	0	498.92	z:\dev\m024 (20GB)	z:\dev\c024 (30GB)	-	z:\dev\b024 (443.92GB)
11	21	0	0	15	0	498.92	z:\dev\m028 (20GB)	z:\dev\c028 (30GB)	-	z:\dev\b028 (443.92GB)
			1	15	0	498.92	z:\dev\m032 (20GB)	z:\dev\c032 (30GB)	-	z:\dev\b032 (443.92GB)
			2	15	0	498.92	z:\dev\m036 (20GB)	z:\dev\c036 (30GB)	-	z:\dev\b036 (443.92GB)
	22	0	0	15	0	498.92	z:\dev\m040 (20GB)	z:\dev\c040 (30GB)	-	z:\dev\b040 (443.92GB)
			1	15	0	498.92	z:\dev\m044 (20GB)	z:\dev\c044 (30GB)	-	z:\dev\b044 (443.92GB)
			2	15	0	498.92	z:\dev\m048 (20GB)	z:\dev\c048 (30GB)	-	z:\dev\b048 (443.92GB)
12	23	0	0	15	0	498.92	z:\dev\m052 (20GB)	z:\dev\c052 (30GB)	-	z:\dev\b052 (443.92GB)
			1	15	0	498.92	z:\dev\m056 (20GB)	z:\dev\c056 (30GB)	-	z:\dev\b056 (443.92GB)
			2	15	0	498.92	z:\dev\m060 (20GB)	z:\dev\c060 (30GB)	-	z:\dev\b060 (443.92GB)
	24	0	0	15	0	498.92	z:\dev\m064 (20GB)	z:\dev\c064 (30GB)	-	z:\dev\b064 (443.92GB)
			1	15	0	498.92	z:\dev\m068 (20GB)	z:\dev\c068 (30GB)	-	z:\dev\b068 (443.92GB)
			2	15	0	498.92	z:\dev\m072 (20GB)	z:\dev\c072 (30GB)	-	z:\dev\b072 (443.92GB)

Type of Database

A statement must be provided that describes:

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

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

Database Mapping

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

No partitioning or replication was used.

60-Days Space

Details of the 60-day space computations along with proof that the database is configured to sustain 8 hours of growth for the dynamic tables (Order, Order-Line, and History) must be disclosed (see Clause 4.2.3).

The detail of 60-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 space used 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 1097.70 GB of logspace (i.e 2,195.39 GB of mirrored transaction log volume) to be available to sustain 8 hours. Total space of priced disks for the transaction log volume was 2,328.13 GB. It indicates 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

514,034.72 tpmC

Response Times

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

Table 5 : Response Times (in seconds)

Type	Average	Maximum	90th %
New-Order	0.89	21.45	1.73
Payment	0.86	24.88	1.68
Interactive Delivery	0.10	2.52	0.11
Stock Level	0.94	22.06	1.82
Order Status	0.88	22.59	1.75
Deferred Delivery	0.10	3.20	0.14
Menu	0.10	2.53	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	Average	Minimum	Maximum
New-Order	18.02	18.00	18.06
Payment	3.02	3.00	3.06
Interactive Delivery	2.02	2.00	2.06
Stock Level	2.02	2.00	2.06
Order Status	2.02	2.00	2.06

Table 7 : Think Times

Type	Average	Minimum	Maximum
New-Order	12.07	0.00	120.71
Payment	12.07	0.00	120.71
Interactive Delivery	5.08	0.00	50.71
Stock Level	5.07	0.00	50.71
Order Status	10.08	0.00	100.71

Response Time Frequency Distribution Curves

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

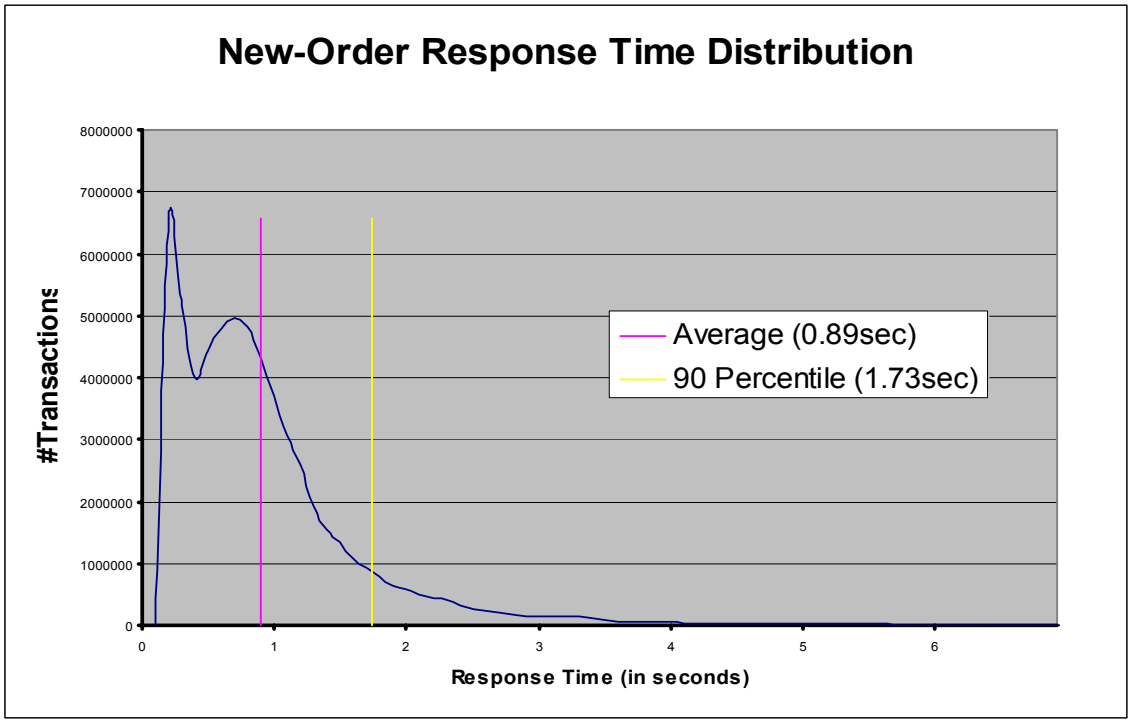


Figure 2.2 : Payment Response Time Distribution

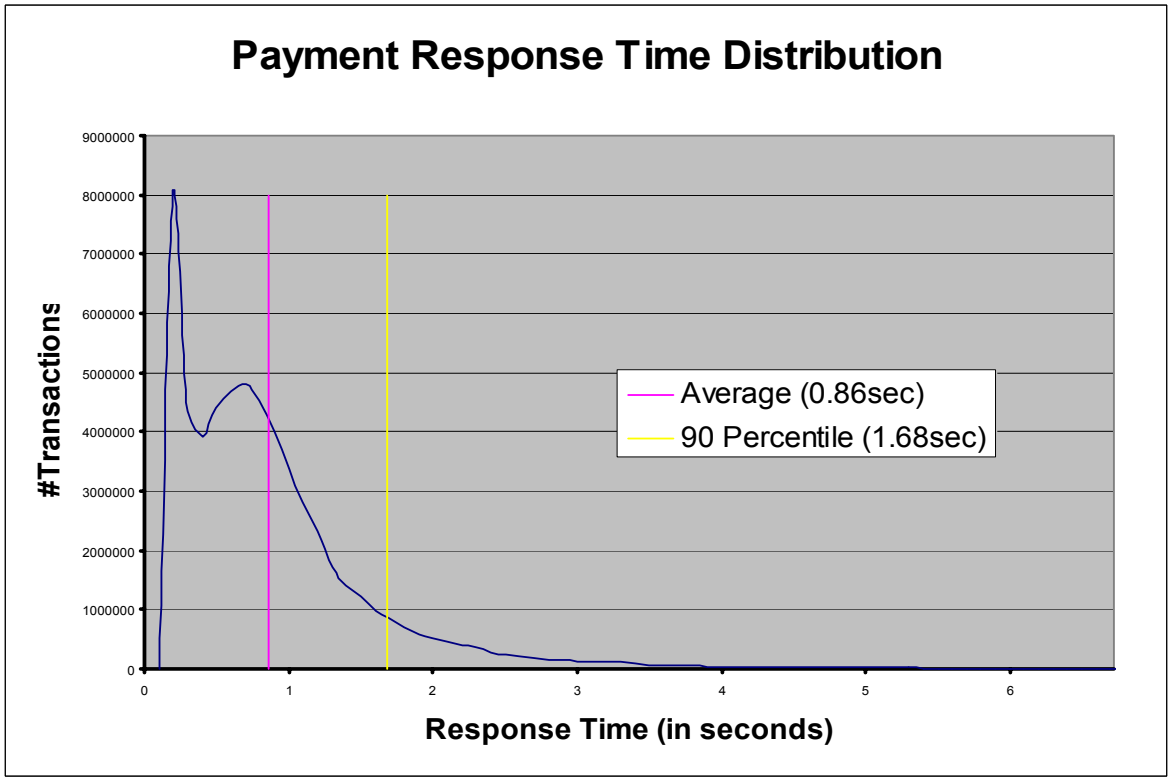


Figure 2.3 : Order-Status Response Time Distribution

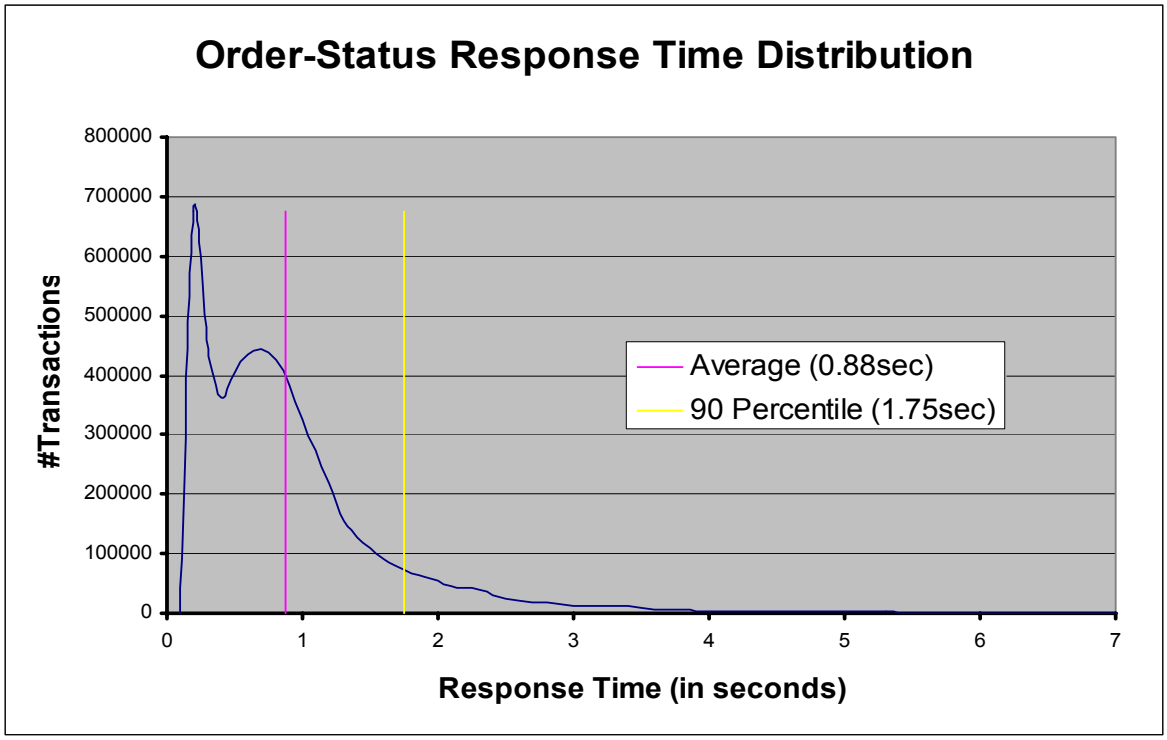


Figure 2.4 : Delivery Response Time Distribution

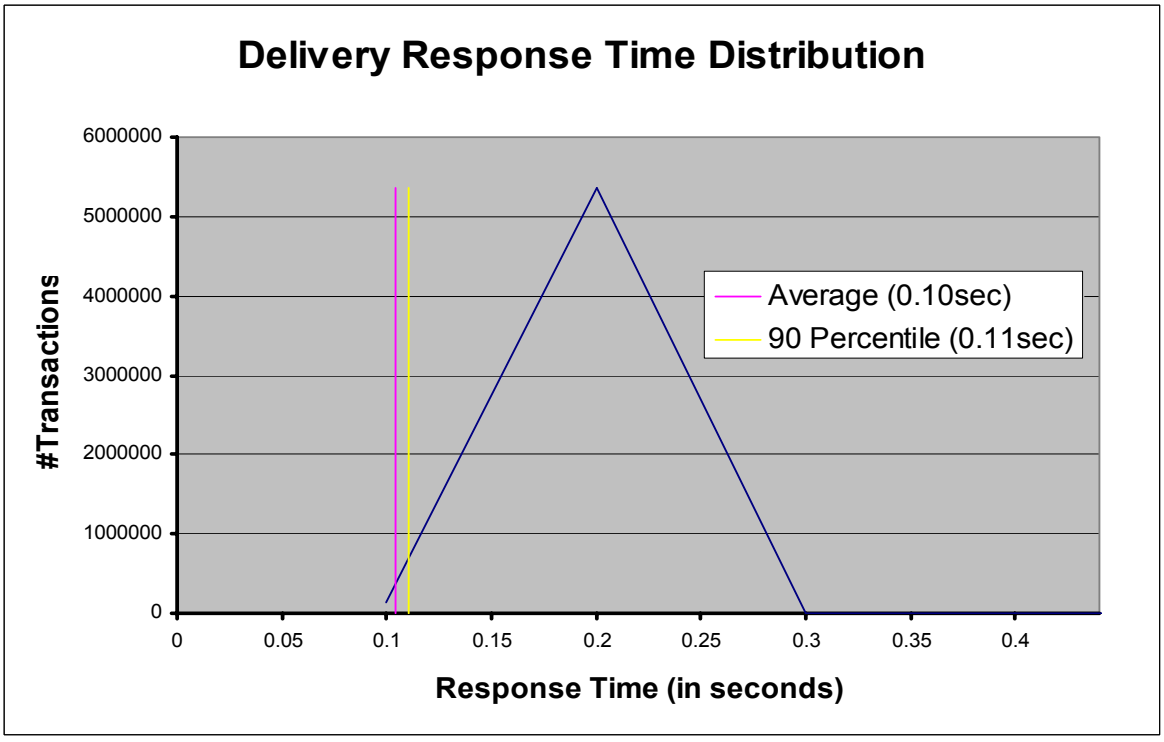
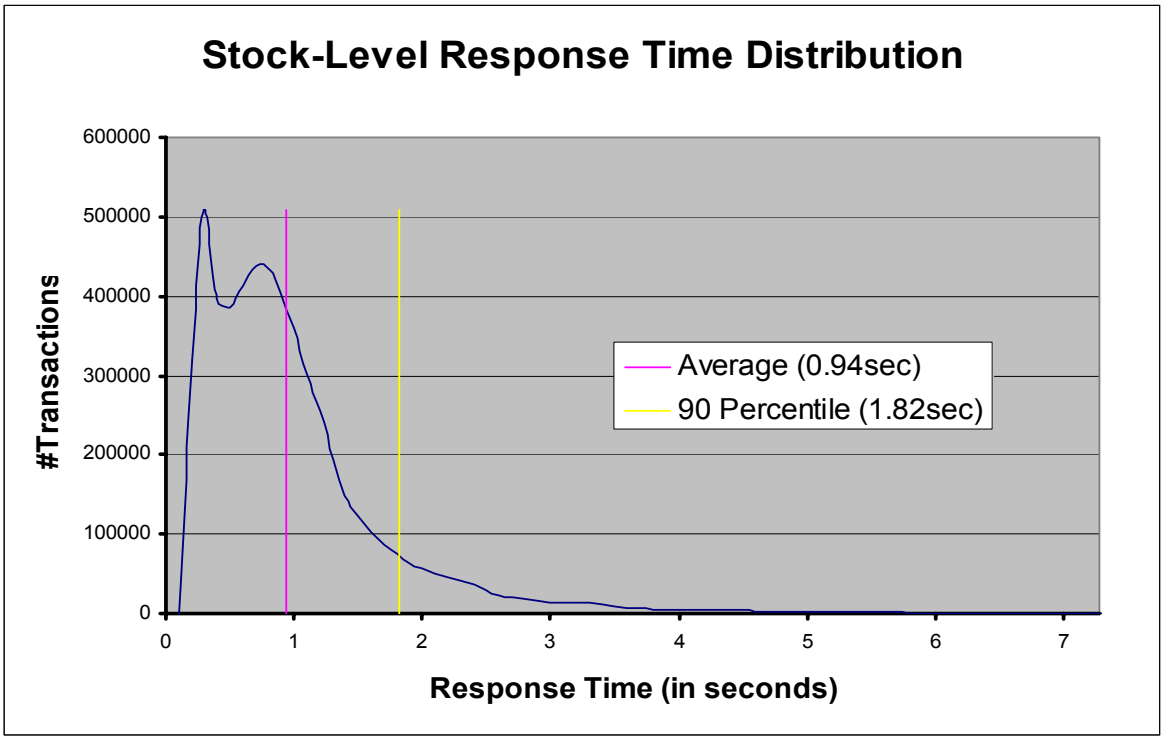


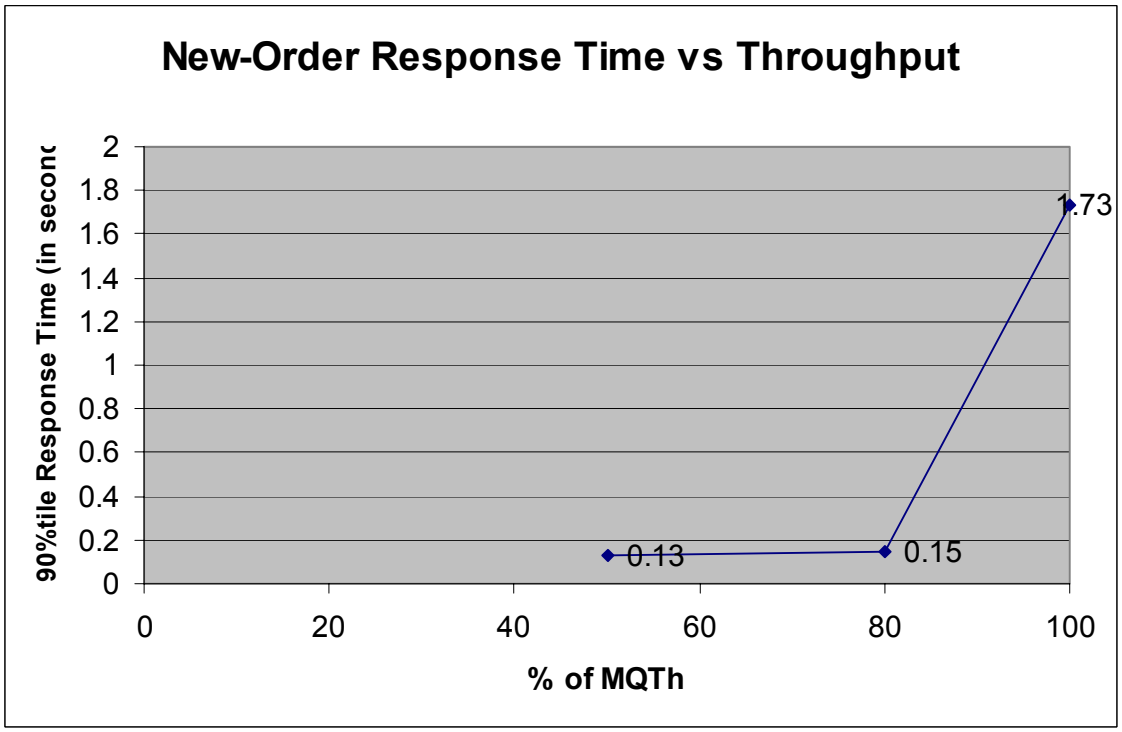
Figure 2.5 : Stock-Level Response Time Distribution



Response time versus Throughput Curve

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

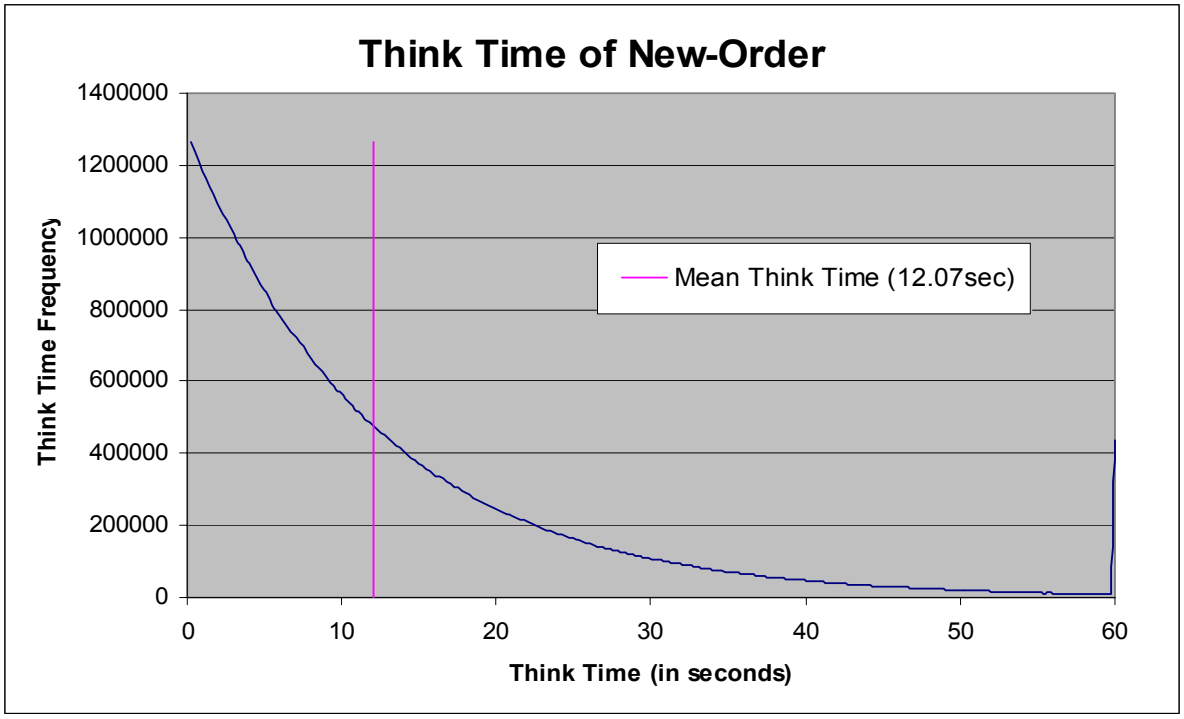
Figure 2.6 Response Time vs. Throughput Curve



New-Order Think Time Frequency Distribution

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

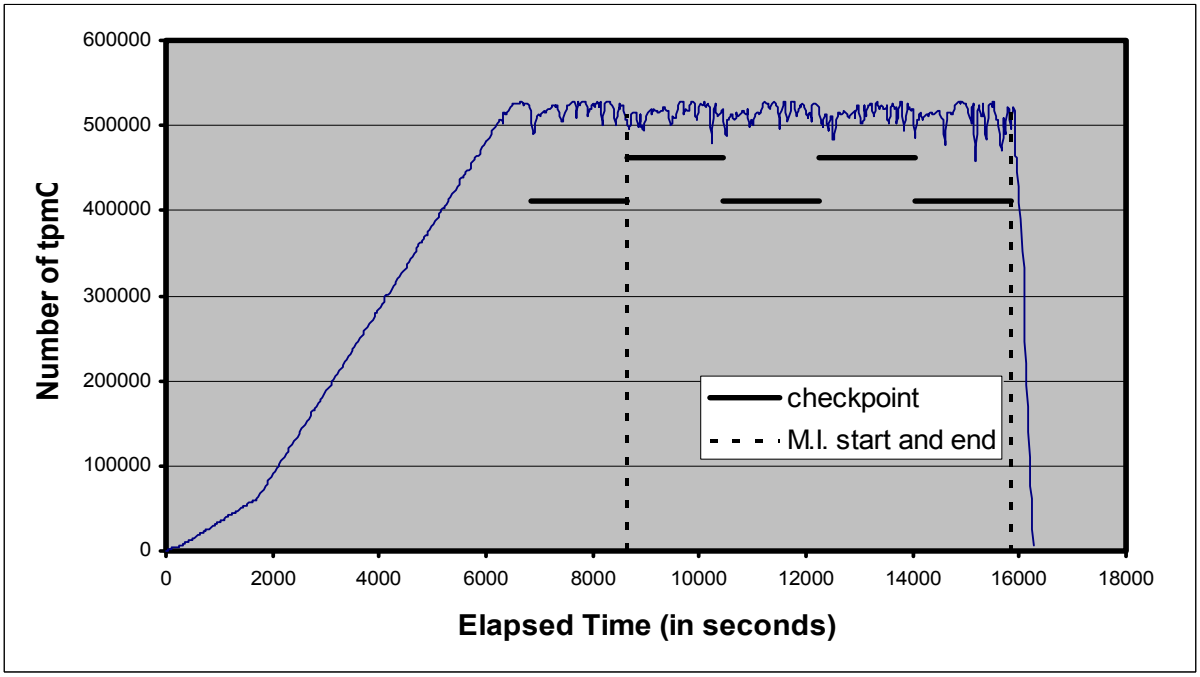
Figure 2.7 New-Order Think Time



New-Order Throughput vs. Elapsed Time

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

Figure 2.8 New Order Throughput vs. Elapsed Time



Steady State

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

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™ 2000 writes to disk all updated memory pages that have not been yet actually written to disk. SQL Server™ 2000 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.

Measurement Period Duration and Checkpoint Duration

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

	Start	End	Duration (in second)
M.I.	00:40:30	02:40:30	7200
1st Checkpoint	00:40:40	01:10:25	1785
2nd Checkpoint	01:10:39	01:40:24	1785
3rd Checkpoint	01:40:38	02:10:23	1785
4th Checkpoint	02:10:36	02:40:21	1785

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 entered 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.

There was one checkpoint before measurement and four checkpoints during measurement.
The time of the first checkpoint during the measurement interval is 10 seconds after the start of the measurement, and the checkpoint interval is 30 minutes.

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.

Loss of Terminal Connections

The number of terminal connections lost during the Measurement Interval must be disclosed (see Clause 6.6.2).

No terminal connections were lost.

Emulated Components

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

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

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

Network configurations and Driver system

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

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

Network Bandwidth

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

The Database server contains eight VI-NIC adapters. The VI-NIC adapters were connected to three FC switches with 2Gbps bandwidth. 31 front-end clients were also connected to these FC switches with 2Gbps bandwidth. Each client has three Ethernet adapters to connect to RTE system . The network bandwidth between RTE system and the front-end clients is 100Mbps. 93 segments were used for the connection of this tested configuration.

Operator Intervention

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

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

Clause 7 : 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 date. If package-pricing is used, vendor part number of the package and a description uniquely identifying each of the components of the package must be disclosed. Pricing source(s) and effective date(s) of price(s) must also be reported.

The total 3-year price of the entire configuration must be reported, including: hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

The 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 calculations must be reported. When the priced system includes products with different availability dates, the reported availability date for the priced system must be the date at which all components are committed to be available. This 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.

The total system as priced will be available October 22, 2003.

Throughput, and Price Performance

A statement of the measured tpmC, as well as the respective calculations for 3-year pricing, price/performance (price/tpmC), and the availability date must be included.

- Maximum Qualified Throughput : 514,034.72 tpmC
- Price per tpmC : $11.50 per tpmC
- Total 3-year cost of ownership : $5,907,028

Country Specific Pricing

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

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

Usage Pricing

For any usage pricing, the sponsor must disclose:

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

The component pricing based on usage is shown below:

- 32 SQL Server™ 2000, Enterprise Edition(64-bit), Processor License
- 1 Windows® Server 2003, Datacenter Edition
- 32 Windows®2000 Server, Server License
- 1 Visual C++ Standard

System Pricing

System pricing should include subtotals for the following components: Server Hardware, Server Software, Client Hardware, Client Software, and Network Components.

System pricing must include line item indication where non-sponsoring companies' brands are used. System pricing must also include line item indication of third party pricing.

A detailed list of all hardware and software, including the 3-year price, is provided in the Executive Summary at the front of this report. All third-party quotations are included in Appendix E at the end of this document.

Clause 8 : 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 InfoSizing 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 charges for similar documents by that 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
Presidio of San Francisco
Building 572B (surface)
P.O. Box 29920 (mail) San Francisco, CA 94129-0920
Voice: 415-561-6272
Fax: 415-561-6120
Email: info@tpc.org

Auditor's letter

INFO SIZING

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL
CERTIFIED AUDITOR

Sponsor: Jun Suzuki
NEC Corporation
1-10 Nisshincho
Fuchu City, Tokyo 183-8501

April 22, 2003

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

Platform: NEC Express5800/1320Xc c/s
Operating system: Microsoft Windows Server 2003, Datacenter Edition for 64-bit Itanium Based System
Database Manager: Microsoft SQL Server 2000 Enterprise Edition (64-bit)
Transaction Manager: Microsoft COM +

The results were:

CPU's Speed	Memory	Disks	NewOrder 90% Response Time	tpmC
Server: NEC Express5800/1320Xc				
32 x Itanium2 (1.5 GHz)	512 GB (6 MB Cache/cpu)	1 x 36 GB int. 1150 x 36 GB ext.	1.73 Seconds	514,034.72
Thirty-one Clients: NEC Express 5800/120Re-2 (Specification for each)				
2 x Xeon (1.8 GHz)	1 GB (512 KB cache/cpu)	1 x 18 GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC requirements for Revision 5.1 of the benchmark.

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

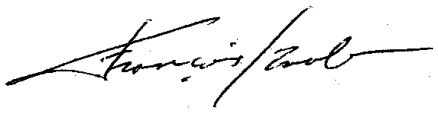
The following verification items were given special attention:

- The transactions were correctly implemented
- The database records were the proper size
- The database was properly scaled and populated
- The ACID properties were met
- 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.
- At least 90% of all delivery transactions met the 80 Second completion time limit
- All 90% response times were under the specified maximums
- The measurement interval was representative of steady state conditions
- The reported measurement interval was 120 minutes
- Four checkpoints were taken during the measurement interval
- The 60 day storage requirement was correctly computed
- The system pricing was verified for major components and maintenance

Additional Audit Notes:

None.

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 **

# TARGTYPE "Win32 (x86) Application" 0x0101

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

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

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

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

odbc32.lib odbccp32.lib /nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib

odbccp32.lib /nologo /subsystem:windows /machine:I386

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir ".\Debug"
# PROP BASE Intermediate_Dir ".\Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\Debug"
# PROP Intermediate_Dir ".\Debug"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /c
# ADD CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib 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, Format Version 6.00
# WARNING: DO NOT EDIT OR DELETE THIS WORKSPACE FILE!

###############################################################################

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

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

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

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

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

Package=<5>
{{{
}}}

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

###############################################################################

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

Package=<5>
{{{
}}}

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

###############################################################################

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

Package=<5>
{{{
}}}

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

###############################################################################

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

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

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

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

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

Package=<5>
{{{
}}}

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

###############################################################################

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

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

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

Package=<5>
{{{
}}}

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

###############################################################################

Global:

Package=<5>
{{{
}}}

Package=<3>
{{{
}}}

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

common/src/error.h

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

```
 *           PURPOSE:     Header file for error exception classes.
 *
 *  Change history:
 *                       4.20.000 - updated rev number to match kit
 *                       4.21.000 - fixed bug: ~CBaseErr needed to be declared
virtual
 */

#pragma once

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

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

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

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


#define ERR_TYPE_LOGIC
                        -1                      //logic error in program;
internal error
#define ERR_SUCCESS
                        0                       //success (a
non-error error)
#define ERR_BAD_ITEM_ID
                        1                       //expected abort record in
txnRecord
#define ERR_TYPE_DELIVERY_POST
            2                       //expected delivery post failed
#define       ERR_TYPE_WEBDLL
                        3                       //tpcc web
generated error
#define       ERR_TYPE_SQL
                        4                       //sql server generated
error
#define       ERR_TYPE_DBLIB
                        5                       //dblib
generated error
#define       ERR_TYPE_ODBC
                        6                       //odbc generated error
#define ERR_TYPE_SOCKET
                        7                       //error on communication
socket client rte only
#define ERR_TYPE_DEADLOCK
                        8                       //dblib and odbc only deadlock condition
#define ERR_TYPE_COM
                        9                       //error from COM call
#define ERR_TYPE_TUXEDO
                        10                      //tuxedo error
#define ERR_TYPE_OS
                        11                      //operating
system error
#define ERR_TYPE_MEMORY
                        12                      //memory allocation error
#define ERR_TYPE_TPCC_ODBC
            13                      //error from tpcc odbc txn module
#define ERR_TYPE_TPCC_DBLIB
                        14                      //error from tpcc dblib
txn module
#define ERR_TYPE_DELISRV
                        15                      //delivery server error
#define ERR_TYPE_TXNLOG
                        16                      //txn log error
#define ERR_TYPE_BCCONN
                        17                      //Benchcraft connection
class
#define ERR_TYPE_TPCC_CONN
                        18                      //Benchcraft connection class
#define ERR_TYPE_ENCINA
                        19                      //Encina error
#define ERR_TYPE_COMPONENT
                        20                      //error from COM component
#define ERR_TYPE_RTE
                        21                      //Benchcraft rte
#define ERR_TYPE_AUTOMATION
                        22                      //Benchcraft automation
errors
#define ERR_TYPE_DRIVER
                        23                      //Driver engine errors
#define ERR_TYPE_RTE_BASE
            24                      //Framework errors
#define       ERR_BUF_OVERFLOW
                        25                      //Buffer overflow during
receive
// TPC-W error types
#define ERR_TYPE_TPCW_CONN
                        50                      //Benchcraft connection class
#define ERR_TYPE_TPCW_HTML
                        51                      //error from TpcwHtml dll
#define ERR_TYPE_TPCW_USER
                        52                      //error from TPC-W user class
#define ERR_TYPE_TPCW_ENG_BASE
                        53
#define ERR_TYPE_TPCW_ENG_OS
                        54
#define ERR_TYPE_HTML_RESP
                        55
#define ERR_TYPE_TPCW_ODBC
                        56
#define ERR_TYPE_SCHANNEL
                        57

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


class CBaseErr
{
public:

            CBaseErr(LPCTSTR szLoc = NULL)
            {
                        m_idMsg                     = INV_ERROR_CODE;


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

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



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


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

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

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

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

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

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

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

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


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


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

            //short       m_errType;
};

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

            CSocketErr(Action eAction, LPCTSTR szLocation = NULL);

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

            Action        m_eAction;
            char          *m_szErrorText;

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

};

class CSystemErr : public CBaseErr
{
public:
            enum Action
            {
                        eNone = 0,
                        eTransactNamedPipe,
                        eWaitNamedPipe,
                        eSetNamedPipeHandleState,
                        eCreateFile,
                        eCreateProcess,
                        eCallNamedPipe,
                        eCreateEvent,
                        eCreateThread,
                        eVirtualAlloc,
                        eReadFile = 10,
                        eWriteFile,
                        eMapViewOfFile,
                        eCreateFileMapping,
                        eInitializeSecurityDescriptor,
                        eSetSecurityDescriptorDacl,
                        eCreateNamedPipe,
                        eConnectNamedPipe,
                        eWaitForSingleObject,
                        eRegOpenKeyEx,
                        eRegQueryValueEx = 20,
                        ebeginthread,
                        eRegEnumValue,
                        eRegSetValueEx,
                        eRegCreateKeyEx,
                        eWaitForMultipleObjects,
                        eRegisterClassEx,
                        eCreateWindow,
                        eCreateSemaphore,
                        eFSeek,
                        eFRead,
                        eFWrite,
                        eTmpFile,
                        eSetFilePointer,
                        eNew,
            };

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

            Action        m_eAction;

private:
            char m_szMsg[ERR_MSG_BUF_SIZE];

};

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

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


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

            int ErrorType() {return ERR_BUF_OVERFLOW;}
```

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

common/src/ReadRegistry.cpp

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


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

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

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

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

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

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

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

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

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

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

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

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

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

            RegCloseKey(hKey);

            return FALSE;
}
```

common/src/ReadRegistry.h

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

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

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

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

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );
```

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 dblib, so redefined here.  Note: we are using the symbol
"__SQLTYPES"
// (declared in sqltypes.h) as a way to determine if TIMESTAMP_STRUCT has been
declared.
#ifndef __SQLTYPES
            typedef struct
            {
                                                short
            /* SQLSMALLINT */           year;
                                                unsigned short          /*
SQLUSMALLINT */             month;
                                                unsigned short          /*
SQLUSMALLINT */             day;
                                                unsigned short          /*
SQLUSMALLINT */             hour;
                                                unsigned short          /*
SQLUSMALLINT */             minute;
                                                unsigned short          /*
SQLUSMALLINT */             second;
                                                unsigned long/* SQLUINTEGER */
            fraction;
            } TIMESTAMP_STRUCT;
#endif

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

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

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

typedef struct
{
            // input params
            short                       w_id;
            short                       d_id;
            long                        c_id;
            short                       o_ol_cnt;

            // output params
            EXEC_STATUS                                 exec_status_code;
            char                        c_last[LAST_NAME_LEN+1];
            char                        c_credit[CREDIT_LEN+1];
            double                      c_discount;
            double                      w_tax;
            double                      d_tax;
            long                                        o_id;
            short
            o_commit_flag;
            TIMESTAMP_STRUCT            o_entry_d;
            short                       o_all_local;
            double                      total_amount;
            OL_NEW_ORDER_DATA           OL[MAX_OL_NEW_ORDER_ITEMS];
} NEW_ORDER_DATA, *PNEW_ORDER_DATA;

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

            // output params
            EXEC_STATUS                                 exec_status_code;
            TIMESTAMP_STRUCT                    h_date;
            char
w_street_1[ADDRESS_LEN+1];
            char
w_street_2[ADDRESS_LEN+1];
            char                                        w_city[ADDRESS_LEN+1];
            char                                        w_state[STATE_LEN+1];
            char                                        w_zip[ZIP_LEN+1];
            char
d_street_1[ADDRESS_LEN+1];
            char
d_street_2[ADDRESS_LEN+1];
            char                                        d_city[ADDRESS_LEN+1];
            char                                        d_state[STATE_LEN+1];
```

```
            char                            d_zip[ZIP_LEN+1];
            char
c_first[FIRST_NAME_LEN+1];
            char                                    c_middle[MIDDLE_NAME_LEN
+ 1];
            char
c_street_1[ADDRESS_LEN+1];
            char
c_street_2[ADDRESS_LEN+1];
            char                                    c_city[ADDRESS_LEN+1];
            char                                    c_state[STATE_LEN+1];
            char                                    c_zip[ZIP_LEN+1];
            char                                    c_phone[PHONE_LEN+1];
            TIMESTAMP_STRUCT            c_since;
            char                                    c_credit[CREDIT_LEN+1];
            double                                  c_credit_lim;
            double                                  c_discount;
            double                                  c_balance;
            char                                    c_data[200+1];
} PAYMENT_DATA, *PPAYMENT_DATA;

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

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

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


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

            // output params
            EXEC_STATUS                             exec_status_code;
            SYSTEMTIME                              queue_time;
            long                                            o_id[10];
            // id's of delivered orders for districts 1 to 10
} DELIVERY_DATA, *PDELIVERY_DATA;

//This structure is used for posting delivery transactions and for writing them
to the delivery server.
typedef struct _DELIVERY_TRANSACTION
{
            SYSTEMTIME    queue;                                    //time
delivery transaction queued
            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/txn_base.h

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

#pragma once

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

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

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

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

common/txnlog/include/rtetime.h

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

//FILE: RTETIME.H

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

common/txnlog/include/spinlock.h

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

#ifndef _INC_Spinlock

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

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

            class Spinlock
            {
                        // Private data.
                        HANDLE                          Semaphore;
                        volatile LONG                   m_Spinlock;
                        volatile LONG                   Waiting;

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

                        public:
                                    // Public functions.

                                    Spinlock( void );

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

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

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

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

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

            inline BOOL Spinlock::ClaimLock( BOOL Wait )
            {
                        if ( ! ClaimSpinlock( (volatile LONG*) &
m_Spinlock ) )
                        {
                                    if ( Wait )
                                                WaitForLock();
                                    return Wait;
                        }
                        return TRUE;
            }

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

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

            #define _INC_Spinlock

#endif
```

common/txnlog/include/txnlog.h

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

#pragma once

    typedef struct _TXN_NEWORDER
            {
```

```
        BYTE  OL_Count;           //range 0 to 31
        BYTE  OL_Remote_Count;    //range 0 to 31
        WORD  c_id;
        int                 o_id;
              } TXN_NEWORDER;

    typedef struct _TXN_PAYMENT
              {
        BYTE  CustByName;
        BYTE  IsRemote;
              } TXN_PAYMENT;

    typedef struct _TXN_ORDERSTATUS
              {
                        BYTE          CustByName;
              } TXN_ORDERSTATUS;

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


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

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


              typedef struct _TXN_RECORD_CONTROL
              {
                        // common header; must exactly match
TXN_RECORD_HEADER
                        JULIAN_TIME   TxnStartTO;               // start of
txn
                        BYTE          TxnType;
              // = TXN_REC_TYPE_CONTROL
                        BYTE          TxnSubType;
              // depends on TxnType
                                  // end of common header

                        DWORD         Len;
              // number of bytes after this field
              } TXN_RECORD_CONTROL, *PTXN_RECORD_CONTROL;


              // TPC-C Txn Record Layout:
              //
    //'TxnStartTO' is a Julian timestamp corresponding to the moment the
    //txn is sent to the SUT, i.e., beginning of response time.  Deltas
    //are in milliseconds.  Note that if RTDelay > 0, then the txn was
    //delayed by this amount.  The delay occurs at the beginning of the
    //response time.  So if RTDelay > 0, then the txn was actually sent
    //at TxnStartTO + RTDelay.
    //
    //Graphically:
    //
    //  time -->
    //
    //    |--- Menu  ---|-- Keying  --|-- Response --|--- Think  --|
    //     <- DeltaT1 -> <- DeltaT2 -> <- DeltaT4  -> <- DeltaT3 ->
    //                               ^
    //                               ^ TxnStartTO
    //
    //RTDelay is the amount of response time delay included in DeltaT4.
    //RTDelay is recorded per txn because this value can be changed on
    //the fly, and so may vary from txn to txn.
    //
    //TxnStatus is the txn completion code.  It is used to indicate errors.
    //For example, in the New Order txn, 1% of txns abort.  TxnStatus will
    //reflect this.

              typedef struct _TXN_RECORD_TPCC
              {
                        // common header; must exactly match
TXN_RECORD_HEADER
                        JULIAN_TIME   TxnStartTO;               // start of
txn
                        BYTE          TxnType;
              // = TXN_REC_TYPE_TPCC
                        BYTE          TxnSubType;
              // depends on TxnType
                                  // end of common header

                        int     DeltaT1;                        // menu time
(ms)
                        int     DeltaT2;                        // keying
time (ms)
                        int     DeltaT3;                        // think time
(ms)
                        int     DeltaT4;                        // response
time (ms)
                        int     RTDelay;                        // response
time delay (ms)
                        int                           TxnError;
              // error code providing more detail for TxnStatus
                        int                           w_id;
              // warehouse ID
                        BYTE          d_id;
              // assigned district ID for this thread
                        BYTE          d_id_ThisTxn; // district ID chosen for
this particular
                        BYTE          TxnStatus;                // completion
status for txn to indicate errors
                        BYTE          reserved;                 // for word
alignment
                        TXN_DETAILS   TxnDetails;  //
              } TXN_RECORD_TPCC, *PTXN_RECORD_TPCC;

              // TPC-C Deferred Delivery Txn Record Layout:
              //
    //Incorporating delivery transaction information into the above
              //structure would increase the size of TXN_DETAILS from 8 to 42
bytes.
    //Hence, we store delivery transaction details in a separate structure.
    //
              typedef struct _TXN_RECORD_TPCC_DELIV_DEF
              {
                        // common header; must exactly match
TXN_RECORD_HEADER
                        JULIAN_TIME   TxnStartTO;               // start of
txn
                        BYTE          TxnType;
              // = TXN_REC_TYPE_TPCC_DELIV_DEF
                        BYTE          TxnSubType;
              // = 0
                                  // end of common header

                        int     DeltaT4;                        // response
time (ms)
                        int                           DeltaTxnExec; // execution
time (ms)
                        int                           w_id;
              // warehouse ID
                        BYTE          TxnStatus;                // completion
status for txn to indicate errors
                        BYTE          reserved;                 // for word
alignment
                        short   o_carrier_id;     // carrier id
                        long          o_id[10];                 // returned
delivery transaction ids
              } TXN_RECORD_TPCC_DELIV_DEF, *PTXN_RECORD_TPCC_DELIV_DEF;



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


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

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

              } TXN_LOG_HEADER, *PTXN_LOG_HEADER;


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



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

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

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

#define       TXN_LOG_OS_ERROR          1
#define       TXN_LOG_NOT_SORTED        2

#define SKIP_CTRL_RECS                  1

class CTxnLog
{
              private:
                        DWORD                           iBufferSize;
              //buffer allocated size
                        DWORD                           iBytesFreeInBuffer;
              //total bytes available for use in buffer
                        int                                             iNumBuffers;
                        //buffers in use
                        int
              iActiveBuffer;                                            //indicates
which buffer is active: 0 or 1
                        int                                             iIoBuffer;
                                  //buffer for any pending IO operation
//                      int                                             iFilePointer;
                        //position in file.
                        LARGE_INTEGER lFilePointer;             //position in
file.
                        int                                             iNextRec;
                                  //when reading, ordinal value of next
record
                        // A "save point" is remembered each time
GetNextRecord is called with a start time specified.
                        // The next time it is called, if start time is after
the save point, we start scanning from the
                        // save point.  This is particularly useful in
FindBestInterval, where the log is scanned repeatedly.
                        JULIAN_TIME               SavePtTime;
//                      int
              iSavePtFilePointer;
                        LARGE_INTEGER lSavePtFilePointer;
                        int
              iSavePtNextRec;

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

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

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

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

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

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

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

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

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

                        int Write(BYTE *ptr, DWORD Size);
```

```
                        static void LogFileIO(CTxnLog *);

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

            public:

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

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

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

                        void CloseTransactionLogFile(void);

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

                        int Sort(void);
                        PTXN_RECORD_HEADER GetSortedRecord();

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

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

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

                        int ErrorType() {return ERR_TYPE_TXNLOG;};

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

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

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

db_dblib_dll/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 **

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

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

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

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

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

odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib

/nologo /subsystem:windows /dll /machine:I386 /out:".\bin/tpcc_dblib.dll"

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

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

odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386
/pdbtype:sept
# ADD LINK32 ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib

/nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_dblib.dll"
/pdbtype:sept

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "db_dblib"
# PROP BASE Intermediate_Dir "db_dblib"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D
"_WINDOWS" /YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /Gm /GX /Zi /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS"
/D "ICECAP" /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib

uuid.lib /nologo /subsystem:windows /dll /debug /machine:I386
/out:".\bin/tpcc_dblib.dll" /pdbtype:sept
# ADD LINK32 icap.lib ntwdblib.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib

uuid.lib /nologo /subsystem:windows /dll /debug /machine:I386
/out:".\bin/tpcc_dblib.dll" /pdbtype:sept

!ENDIF

# Begin Target

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

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

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

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

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

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

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

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

db_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 <sqldb.h>

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

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

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

#define       DEFCLPACKSIZE                         4096

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

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

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

BOOL APIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID
lpReserved)
{
            switch( ul_reason_for_call )
            {
                        case DLL_PROCESS_ATTACH:
                                    DisableThreadLibraryCalls(hModule);
```

```
                                        dbinit();                       // initialize
dblib
                                        break;

                                case DLL_PROCESS_DETACH:
                                        dbexit();                       // close all
dblib structures/connections
                                        break;

                                default:
                                        /* nothing */;
                }
                return TRUE;
}


int err_handler(DBPROCESS *dbproc, int severity, int dberr, int oserr, LPCSTR
dberrstr, LPCSTR oserrstr)
{
                CTPCC_DBLIB                             *pConn;

                assert(dbproc != NULL);
                pConn = (CTPCC_DBLIB*)dbgetuserdata(dbproc);

                if (pConn != NULL)
                {
                        pConn->SetDbLibError( severity, dberr, oserr,
dberrstr, oserrstr );
                }
                return INT_CANCEL;
}

/* FUNCTION: int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int
severity, char *msgtext)
 *
 * PURPOSE:   This function handles DB-Library SQL Server error messages
 *
 * ARGUMENTS: DBPROCESS                    *dbproc
                DBPROCESS id pointer
 *              msgno                                   DBINT
                                                        message number
 *                                                      int
                        msgstate                        message state
 *                                                      int
                        severity                        message severity
 *                                                      char
                *msgtext                        printable message description
 *
 * RETURNS:               int
                INT_CONTINUE  continue if error is SQLETIME else INT_CANCEL action
 *
                        INT_CANCEL                      cancel operation
 *
 * COMMENTS: This function also sets the dead lock dbproc variable if necessary.
 *
 */

// typedef INT (SQLAPI *DBMSGHANDLE_PROC)(PDBPROCESS, DBINT, INT, INT, LPCSTR,
LPCSTR, LPCSTR, DBUSMALLINT);

int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int severity,
                                                        LPCSTR msgtext, LPCSTR
srvname, LPCSTR procname, DBUSMALLINT line)
{
                CTPCC_DBLIB                             *pConn;

                assert(dbproc != NULL);
                pConn = (CTPCC_DBLIB*)dbgetuserdata(dbproc);

                if (pConn != NULL)
                {
                        pConn->SetSqlError( msgno, msgstate, severity,
msgtext );
                }

                return 0;
}

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

inline static void UtilStrCpy(char * pDest, const BYTE * pSrc, int n)
{
    strncpy(pDest, (char *)pSrc, n);
    pDest[n] = '\0';

                return;
}

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
 *
 */
```

```
char* CTPCC_DBLIB_ERR::ErrorText(void)
{
                int i;

                static SERRORMSG errorMsgs[] =
                {
                                { ERR_WRONG_SP_VERSION,                 "Wrong
version of stored procs on database server"     },
                                { ERR_INVALID_CUST,
                "Invalid Customer id,name."
                                                                        },
                                { ERR_NO_SUCH_ORDER,                    "No orders
found for customer."
                                },
                                { ERR_RETRIED_TRANS,                    "Retries
before transaction succeeded."
                                },
                                { 0,
                                ""


                                                }
                };

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

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

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


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

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

                m_MaxRetries = 10;                      // how many retries on
deadlock

                // increase max number of connections if getting close
                if ( dbgetmaxprocs() < (iConnectionCount+5) )
                {
                                if ( dbsetmaxprocs(iConnectionCount+10) == FAIL )
                                                ThrowError(CDBLIBERR::eDbSetMaxProcs);
                }

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

                // register error and message handler functions
                if (dbprocerrhandle(login, err_handler) == NULL)
                                ThrowError(CDBLIBERR::eDbProcHandler);

                if (dbprocmsghandle(login, msg_handler) == NULL)
                                ThrowError(CDBLIBERR::eDbProcHandler);


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

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

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

                m_dbproc = dbopen(login, szServer);
```

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

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

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

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

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

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

                DiscardNextResults(2);

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

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

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

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

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


                DiscardNextRows(0);
                DiscardNextResults(0);
}


CTPCC_DBLIB::~CTPCC_DBLIB( void )
{
                // close db connection and deallocate resources
                dbclose(m_dbproc);
                InterlockedDecrement( &iConnectionCount );
                if (m_DbLibErr != NULL)
                                delete m_DbLibErr;
                if (m_SqlErr != NULL)
                                delete m_SqlErr;
}

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

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

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

}

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

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

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

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

```
		// check for SQL Server error first;  if yes, throw it and ignore
any DBLib error.
		if (m_SqlErr != NULL)
		{
			CSQLERR			*pSqlErr;
			pSqlErr = m_SqlErr;
			m_SqlErr = NULL;		// clear our pointer to
instance; catch handler will delete
			throw pSqlErr;
		}

		CDBLIBERR	*pDbLibErr;
		if (m_DbLibErr == NULL)
			// this case isn't expected to happen, since it means
that an error was returned
			// but the error handlers were not called.
			pDbLibErr = new CDBLIBERR(eAction);
		else
		{
			pDbLibErr = m_DbLibErr;
			pDbLibErr->m_eAction = eAction;
			m_DbLibErr = NULL;				// clear our
pointer to instance; catch handler will delete
		}

		throw pDbLibErr;
}

// Read and discard rows until no more.  Throw an exception if number of rows
read doesn't
// match number of rows expected.  The row count will be ignored if the expected
count value
// passed in is negative.  A typical use of this routine is to verify that there
are no more
// rows to be read.
void CTPCC_DBLIB::DiscardNextRows(int iExpectedCount)
{
		int			iRowsRead = 0;
		RETCODE		rc;

		while (TRUE)
		{
			rc = dbnextrow(m_dbproc);
			if (rc == NO_MORE_ROWS)
				break;
			if (rc == FAIL)
			{
				if (iExpectedCount >= 0)

		ThrowError(CDBLIBERR::eDbNextRow);
				else
					break;
			}
			iRowsRead++;
		}

		if ((iExpectedCount >= 0) &&
			(iExpectedCount != iRowsRead))
			ThrowError(CDBLIBERR::eWrongRowCount);
}

// Read and discard results until no more.  Throw an exception if number of
result sets read doesn't
// match number expected.  The result set count will be ignored if the expected
count value
// passed in is negative.  A typical use of this routine is to verify that there
are no more
// result sets to be read.
void CTPCC_DBLIB::DiscardNextResults(int iExpectedCount)
{
		int			iResultsRead = 0;
		RETCODE		rc;

		while (TRUE)
		{
			rc = dbresults(m_dbproc);
			if (rc == NO_MORE_RESULTS)
				break;
			if (rc == FAIL)
			{
				if (iExpectedCount >= 0)

		ThrowError(CDBLIBERR::eDbResults);
				else
					break;
			}

			DiscardNextRows(-1);
			iResultsRead++;
		}

		if ((iExpectedCount >= 0) &&
			(iExpectedCount != iResultsRead))
			ThrowError(CDBLIBERR::eWrongRowCount);
}

void CTPCC_DBLIB::StockLevel()
{
		int					iTryCount = 0;
		const BYTE	*pData;

		ResetError();

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

				dbrpcparam(m_dbproc, NULL, 0, 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);	// @threshhold smallint

				if (dbrpcexec(m_dbproc) == FAIL)

		ThrowError(CDBLIBERR::eDbRpcExec);

				if (dbresults(m_dbproc) != SUCCEED)

		ThrowError(CDBLIBERR::eDbResults);

				if (dbnextrow(m_dbproc) != REG_ROW)

		ThrowError(CDBLIBERR::eDbNextRow);

				if (pData=dbdata(m_dbproc, 1))
					m_txn.StockLevel.low_stock
= *((long *) pData);

				DiscardNextRows(0);
				DiscardNextResults(0);

				m_txn.StockLevel.exec_status_code = eOK;
				return;
			}
			catch (CSQLERR *e)
			{
				if ((e->m_msgno == 1205 ||
					(e->m_msgno ==
iErrOleDbProvider &&
					strstr(e->m_msgtext,
sErrTimeoutExpired) != NULL)) &&
					(++iTryCount <=
iMaxRetries))
				{
					// hit deadlock; backoff
for increasingly longer period
					delete e;
					Sleep(10 * iTryCount);
				}
				else
					throw;
			}
			// while (TRUE)

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


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

		int					iTryCount = 0;
		const BYTE	*pData;

		ResetError();

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

				dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -
1, -1, (BYTE *) &m_txn.NewOrder.w_id);	dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -
1, -1, (BYTE *) &m_txn.NewOrder.d_id);	dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -
1, -1, (BYTE *) &m_txn.NewOrder.c_id);	dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -
1, -1, (BYTE *) &m_txn.NewOrder.o_ol_cnt);

				// check whether any order lines are for
a remote warehouse
				m_txn.NewOrder.o_all_local = 1;
				for (i = 0; i < m_txn.NewOrder.o_ol_cnt;
i++)
				{
					if
(m_txn.NewOrder.OL[i].ol_supply_w_id != m_txn.NewOrder.w_id)
					{
		m_txn.NewOrder.o_all_local = 0;  // at least one remote warehouse
						break;
					}
				}
				dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -
1, -1, (BYTE *) &m_txn.NewOrder.o_all_local);

				for (i = 0; i < m_txn.NewOrder.o_ol_cnt;
i++)
				{
					dbrpcparam(m_dbproc, NULL,
0, SQLINT4, -1, -1, (BYTE *) &m_txn.NewOrder.OL[i].ol_i_id);
					dbrpcparam(m_dbproc, NULL,
0, SQLINT2, -1, -1, (BYTE *) &m_txn.NewOrder.OL[i].ol_supply_w_id);
					dbrpcparam(m_dbproc, NULL,
0, SQLINT2, -1, -1, (BYTE *) &m_txn.NewOrder.OL[i].ol_quantity);
				}

				if (dbrpcexec(m_dbproc) == FAIL)

		ThrowError(CDBLIBERR::eDbRpcExec);

				// Get order line results
				m_txn.NewOrder.total_amount = 0;
				for (i = 0; i<m_txn.NewOrder.o_ol_cnt;
i++)
				{
					if (dbresults(m_dbproc) !=
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 = (*(DBTINYINT
*) pData);

                                    DiscardNextRows(0);
                                    DiscardNextResults(0);


                                    if (commit_flag == 1)
                                    {

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

m_txn.NewOrder.c_discount));

            m_txn.NewOrder.exec_status_code = eOK;
                                    }
                                    else

            m_txn.NewOrder.exec_status_code = eInvalidItem;

                                    return;
                        }
                        catch (CSQLERR *e)
                        {
                                    if ((e->m_msgno == 1205 ||
                                                (e->m_msgno ==
iErrOleDbProvider &&
                                                strstr(e->m_msgtext,
sErrTimeoutExpired) != NULL)) &&
                                                (++iTryCount <=
iMaxRetries))
                                    {
                                                // hit deadlock; backoff
for increasingly longer period
                                                delete e;
                                                Sleep(10 * iTryCount);
                                    }
                                    else
                                                throw;
                        }
            }           // while (TRUE)

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


void CTPCC_DBLIB::Payment()
{
            DBDATETIME    datetime;
            DBDATEREC     daterec;

            int                                 iTryCount = 0;
            const BYTE    *pData;

            ResetError();

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

                                    dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -
1, -1, (BYTE *) &m_txn.Payment.w_id);
                                    dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -
1, -1, (BYTE *) &m_txn.Payment.c_w_id);
                                    dbrpcparam(m_dbproc, NULL, 0, SQLFLT8, -
1, -1, (BYTE *) &m_txn.Payment.h_amount);
                                    dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -
1, -1, (BYTE *) &m_txn.Payment.d_id);
                                    dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -
1, -1, (BYTE *) &m_txn.Payment.c_d_id);
                                    dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -
1, -1, (BYTE *) &m_txn.Payment.c_id);

                                    // if customer id is zero, then payment
is by name
                                    if (m_txn.Payment.c_id == 0)
                                                dbrpcparam(m_dbproc, NULL,
0, SQLCHAR, -1, strlen(m_txn.Payment.c_last), (unsigned char
*)m_txn.Payment.c_last);

                                    if (dbrpcexec(m_dbproc) == FAIL)

            ThrowError(CDBLIBERR::eDbRpcExec);

                                    if (dbresults(m_dbproc) != SUCCEED)

            ThrowError(CDBLIBERR::eDbResults);

                                    if (dbnextrow(m_dbproc) != REG_ROW)

            ThrowError(CDBLIBERR::eDbNextRow);

                                    if (dbnumcols(m_dbproc) != 27)

            ThrowError(CDBLIBERR::eWrongNumCols);

                                    if (pData=dbdata(m_dbproc, 1))
                                                m_txn.Payment.c_id =
*((DBINT *) pData);
                                    if (pData=dbdata(m_dbproc, 2))

            UtilStrCpy(m_txn.Payment.c_last, pData, dbdatlen(m_dbproc, 2));
                                    if (pData=dbdata(m_dbproc, 3))
                                    {
                                                datetime = *((DBDATETIME
*) pData);
                                                dbdatecrack(m_dbproc,
&daterec, &datetime);
                                                m_txn.Payment.h_date.year
= daterec.year;
                                                m_txn.Payment.h_date.month
= daterec.month;
                                                m_txn.Payment.h_date.day
= daterec.day;
                                                m_txn.Payment.h_date.hour
= daterec.hour;

            m_txn.Payment.h_date.minute = daterec.minute;

            m_txn.Payment.h_date.second = daterec.second;
                                    }
                                    if (pData=dbdata(m_dbproc, 4))

            UtilStrCpy(m_txn.Payment.w_street_1, pData, dbdatlen(m_dbproc,
4));
                                    if (pData=dbdata(m_dbproc, 5))

            UtilStrCpy(m_txn.Payment.w_street_2, pData, dbdatlen(m_dbproc,
5));
                                    if (pData=dbdata(m_dbproc, 6))

            UtilStrCpy(m_txn.Payment.w_city, pData, dbdatlen(m_dbproc, 6));
                                    if (pData=dbdata(m_dbproc, 7))

            UtilStrCpy(m_txn.Payment.w_state, pData, dbdatlen(m_dbproc, 7));
                                    if (pData=dbdata(m_dbproc, 8))

            UtilStrCpy(m_txn.Payment.w_zip, pData, dbdatlen(m_dbproc, 8));
                                    if (pData=dbdata(m_dbproc, 9))

            UtilStrCpy(m_txn.Payment.d_street_1, pData, dbdatlen(m_dbproc,
9));
                                    if (pData=dbdata(m_dbproc, 10))

            UtilStrCpy(m_txn.Payment.d_street_2, pData, dbdatlen(m_dbproc,
10));
                                    if (pData=dbdata(m_dbproc, 11))

            UtilStrCpy(m_txn.Payment.d_city, pData, dbdatlen(m_dbproc, 11));
                                    if (pData=dbdata(m_dbproc, 12))

            UtilStrCpy(m_txn.Payment.d_state, pData, dbdatlen(m_dbproc, 12));
                                    if (pData=dbdata(m_dbproc, 13))

            UtilStrCpy(m_txn.Payment.d_zip, pData, dbdatlen(m_dbproc, 13));
                                    if (pData=dbdata(m_dbproc, 14))

            UtilStrCpy(m_txn.Payment.c_first, pData, dbdatlen(m_dbproc, 14));
                                    if (pData=dbdata(m_dbproc, 15))

            UtilStrCpy(m_txn.Payment.c_middle, pData, dbdatlen(m_dbproc, 15));
                                    if (pData=dbdata(m_dbproc, 16))

            UtilStrCpy(m_txn.Payment.c_street_1, pData, dbdatlen(m_dbproc,
16));
                                    if (pData=dbdata(m_dbproc, 17))

            UtilStrCpy(m_txn.Payment.c_street_2, pData, dbdatlen(m_dbproc,
17));
                                    if (pData=dbdata(m_dbproc, 18))

            UtilStrCpy(m_txn.Payment.c_city, pData, dbdatlen(m_dbproc, 18));
                                    if (pData=dbdata(m_dbproc, 19))

            UtilStrCpy(m_txn.Payment.c_state, pData, dbdatlen(m_dbproc, 19));
                                    if (pData=dbdata(m_dbproc, 20))

            UtilStrCpy(m_txn.Payment.c_zip, pData, dbdatlen(m_dbproc, 20));
                                    if (pData=dbdata(m_dbproc, 21))

            UtilStrCpy(m_txn.Payment.c_phone, pData, dbdatlen(m_dbproc, 21));
                                    if (pData=dbdata(m_dbproc, 22))
                                    {
                                                datetime = *((DBDATETIME
*) pData);
                                                dbdatecrack(m_dbproc,
&daterec, &datetime);
                                                m_txn.Payment.c_since.year
= daterec.year;

            m_txn.Payment.c_since.month  = daterec.month;
                                                m_txn.Payment.c_since.day
= daterec.day;
                                                m_txn.Payment.c_since.hour
= daterec.hour;

            m_txn.Payment.c_since.minute = daterec.minute;

            m_txn.Payment.c_since.second = daterec.second;
                                    }
                                    if(pData=dbdata(m_dbproc, 23))

            UtilStrCpy(m_txn.Payment.c_credit, pData, dbdatlen(m_dbproc, 23));

                                    if(pData=dbdata(m_dbproc, 24))
                                                dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,24), SQLFLT8, (BYTE
*)&m_txn.Payment.c_credit_lim, 8);
                                    if(pData=dbdata(m_dbproc, 25))
                                                dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,25), SQLFLT8, (BYTE
*)&m_txn.Payment.c_discount, 8);
                                    if(pData=dbdata(m_dbproc, 26))
                                                dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,26), SQLFLT8, (BYTE
*)&m_txn.Payment.c_balance, 8);
                                    if(pData=dbdata(m_dbproc, 27))

            UtilStrCpy(m_txn.Payment.c_data, pData, dbdatlen(m_dbproc, 27));

                                    DiscardNextRows(0);
                                    DiscardNextResults(0);


                                    if (m_txn.Payment.c_id == 0)
                                                throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
                                    else

            m_txn.Payment.exec_status_code = eOK;

                                    return;
                        }
                        catch (CSQLERR *e)
                        {
                                    if ((e->m_msgno == 1205 ||
                                                (e->m_msgno ==
iErrOleDbProvider &&
                                                strstr(e->m_msgtext,
sErrTimeoutExpired) != NULL)) &&
                                                (++iTryCount <=
iMaxRetries))
                                    {
                                                // hit deadlock; backoff
for increasingly longer period
                                                delete e;
                                                Sleep(10 * iTryCount);
                                    }
                                    else
                                                throw;
                        }
            }           // while (TRUE)

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


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

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

            ResetError();

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

                                    dbrpcparam(m_dbproc, NULL, 0, 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) != SUCCEED)
                                    {
                                                if ((m_DbLibErr == NULL)
&& (m_SqlErr == NULL))
                                                            throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
                                                else

            ThrowError(CDBLIBERR::eDbResults);
                                    }

                                    if (dbnumcols(m_dbproc) != 5)
```

```
		ThrowError(CDBLIBERR::eWrongNumCols);

                                        i = 0;
                                        while (TRUE)
                                        {
                                                rc = dbnextrow(m_dbproc);
                                                if (rc == NO_MORE_ROWS)
                                                        break;
                                                if (rc != REG_ROW)

		ThrowError(CDBLIBERR::eDbNextRow);

                                                if(pData=dbdata(m_dbproc,
1))

		m_txn.OrderStatus.OL[i].ol_supply_w_id = (*(DBSMALLINT *) pData);
                                                if(pData=dbdata(m_dbproc,
2))

		m_txn.OrderStatus.OL[i].ol_i_id = (*(DBINT *) pData);
                                                if(pData=dbdata(m_dbproc,
3))

		m_txn.OrderStatus.OL[i].ol_quantity = (*(DBSMALLINT *) pData);
                                                if(pData=dbdata(m_dbproc,
4))

		dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,4),

		  SQLFLT8, (BYTE *)&m_txn.OrderStatus.OL[i].ol_amount, 8);
                                                if(pData=dbdata(m_dbproc,
5))

                                                {
                                                        datetime =
*((DBDATETIME *) pData);

		dbdatecrack(m_dbproc, &daterec, &datetime);

		m_txn.OrderStatus.OL[i].ol_delivery_d.year   = daterec.year;

		m_txn.OrderStatus.OL[i].ol_delivery_d.month  = daterec.month;

		m_txn.OrderStatus.OL[i].ol_delivery_d.day    = daterec.day;

		m_txn.OrderStatus.OL[i].ol_delivery_d.hour   = daterec.hour;

		m_txn.OrderStatus.OL[i].ol_delivery_d.minute = daterec.minute;

		m_txn.OrderStatus.OL[i].ol_delivery_d.second = daterec.second;
                                                }
                                                i++;
                                        }
                                        m_txn.OrderStatus.o_ol_cnt = i;


                                        if (dbresults(m_dbproc) != SUCCEED)

		ThrowError(CDBLIBERR::eDbResults);

                                        if (dbnextrow(m_dbproc) != REG_ROW)

		ThrowError(CDBLIBERR::eDbNextRow);

                                        if (dbnumcols(m_dbproc) != 8)

		ThrowError(CDBLIBERR::eWrongNumCols);

                                        if(pData=dbdata(m_dbproc, 1))
                                                m_txn.OrderStatus.c_id =
(*(DBINT *) pData);
                                        if(pData=dbdata(m_dbproc, 2))

		UtilStrCpy(m_txn.OrderStatus.c_last, pData, dbdatlen(m_dbproc,2));
                                        if(pData=dbdata(m_dbproc, 3))

		UtilStrCpy(m_txn.OrderStatus.c_first, pData,
dbdatlen(m_dbproc,3));
                                        if(pData=dbdata(m_dbproc, 4))

		UtilStrCpy(m_txn.OrderStatus.c_middle, pData, dbdatlen(m_dbproc,
4));
                                        if(pData=dbdata(m_dbproc, 5))
                                        {
                                                datetime = *((DBDATETIME
*) pData);
                                                dbdatecrack(m_dbproc,
&daterec, &datetime);

		m_txn.OrderStatus.o_entry_d.year   = daterec.year;

		m_txn.OrderStatus.o_entry_d.month  = daterec.month;

		m_txn.OrderStatus.o_entry_d.day    = daterec.day;

		m_txn.OrderStatus.o_entry_d.hour   = daterec.hour;

		m_txn.OrderStatus.o_entry_d.minute = daterec.minute;

		m_txn.OrderStatus.o_entry_d.second = daterec.second;
                                        }
                                        if(pData=dbdata(m_dbproc, 6))

		m_txn.OrderStatus.o_carrier_id = (*(DBSMALLINT *) pData);
                                        if(pData=dbdata(m_dbproc, 7))
                                                dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,7),
SQLFLT8, (BYTE *)&m_txn.OrderStatus.c_balance, 8);
                                        if(pData=dbdata(m_dbproc, 8))
                                                m_txn.OrderStatus.o_id =
(*(DBINT *) pData);
                                        DiscardNextRows(0);
                                        DiscardNextResults(0);


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

                m_txn.OrderStatus.exec_status_code = eOK;

                                        return;
                                }
                                catch (CSQLERR *e)
                                {
                                        if ((e->m_msgno == 1205 ||
                                                (e->m_msgno ==
iErrOleDbProvider &&
                                                strstr(e->m_msgtext,
sErrTimeoutExpired) != NULL)) &&
                                                (++iTryCount <=
iMaxRetries))
                                        {
                                                // hit deadlock; backoff
for increasingly longer period
                                                delete e;
                                                Sleep(10 * iTryCount);
                                        }
                                        else
                                                throw;
                                }
                        }               // while (TRUE)

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


void CTPCC_DBLIB::Delivery()
{
                int                                     i;
                int                                     iTryCount = 0;
                const BYTE      *pData;

                ResetError();

                while (TRUE)
                {
                        try
                        {
                                dbrpcinit(m_dbproc, "tpcc_delivery", 0);
                                dbrpcparam(m_dbproc, NULL, 0, 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 &&
                                        strstr(e->m_msgtext,
sErrTimeoutExpired) != NULL)) &&
                                        (++iTryCount <=
iMaxRetries))
                                {
                                        // hit deadlock; backoff
for increasingly longer period
                                        delete e;
                                        Sleep(10 * iTryCount);
                                }
                                else
                                        throw;
                        }
                }               // while (TRUE)

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

void CTPCC_DBLIB::ResetError()
{
                if (m_DbLibErr != NULL)
                {
                        delete m_DbLibErr;
                        m_DbLibErr = (CDBLIBERR*)NULL;
                }

                if (m_SqlErr != NULL)
                {
                        delete m_SqlErr;
                        m_SqlErr = (CSQLERR*)NULL;
                }
                return;
}
```

db_dblib_dll/src/tpcc_dblib.h

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

#ifndef PDBPROCESS
#define DBPROCESS void   // dbprocess structure type
typedef DBPROCESS * PDBPROCESS;
#endif

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

class CSQLERR : public CBaseErr
{
                public:

                                CSQLERR(void)
                                {
                                        m_msgno = 0;
                                        m_msgstate = 0;
                                        m_severity = 0;
                                        m_msgtext = NULL;
                                };

                                ~CSQLERR()
                                {
                                        delete [] m_msgtext;
                                };

                                int                             m_msgno;
                                int                             m_msgstate;
                                int                             m_severity;
                                char    *m_msgtext;

                                int ErrorType() {return ERR_TYPE_SQL;};
                                int ErrorNum() {return m_msgno;};
                                char *ErrorText() {return m_msgtext;};

};

class CDBLIBERR : public CBaseErr
{
                public:
                                enum ACTION
                                {
                                        eNone,
                                        eUnknown,
                                        eLogin,
                // error from dblogin
                                        eDbOpen,
                // error from dbopen
                                        eDbUse,
                // error from dbuse             eDbSqlExec,
                // error from dbsqlexec
                                        eDbSet,
                // error from one of the dbset* routines
                                        eDbNextRow,
                // error from dbnextrow         eWrongRowCount,
                // more or less rows returned than expected
                                        eWrongNumCols,
                // more or less columns returned than expected
                                        eDbResults,
                // error from dbresults
                                        eDbRpcExec,
                // error from dbrpcexec
                                        eDbSetMaxProcs,
```

```
        // error from dbsetmaxprocs
                                EDbProcHandler
        // error from either dbprocerrhandle or dbprocmsghandle
                        };

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

                                m_dberrstr = NULL;
                                m_oserrstr = NULL;
                        };

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

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

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

};

class CTPCC_DBLIB_ERR : public CBaseErr
{
        public:
                        enum CTPCC_DBLIB_ERRS
                        {
                                ERR_WRONG_SP_VERSION = 1,  // "Wrong
version of stored procs on database server"
                                ERR_INVALID_CUST,
        // "Invalid Customer id,name."
                                ERR_NO_SUCH_ORDER,
        // "No orders found for customer."
                                ERR_RETRIED_TRANS,
        // "Retries before transaction succeeded."
                        };

                        CTPCC_DBLIB_ERR( int iErr ) { m_errno = iErr;
m_iTryCount = 0; };

                        CTPCC_DBLIB_ERR( int iErr, int iTryCount ) { m_errno
= iErr; m_iTryCount = iTryCount; };

                        int                        m_errno;
                        int                        m_iTryCount;

                        int ErrorType() {return ERR_TYPE_TPCC_DBLIB;};
                        int ErrorNum() {return m_errno;};

                        char *ErrorText();
};

class DllDecl CTPCC_DBLIB : public CTPCC_BASE
{
        private:
                        // declare variables and private functions here...
                        PDBPROCESS    m_dbproc;
                        CDBLIBERR     *m_DbLibErr;                // not
allocated until needed (maybe never)
                        CSQLERR                        *m_SqlErr;
        // not allocated until needed (maybe never)
                        int                                        m_MaxRetries;
        // retry count on deadlock

                        void DiscardNextRows(int iExpectedCount);
                        void DiscardNextResults(int iExpectedCount);
                        void ThrowError( CDBLIBERR::ACTION eAction );
                        void ResetError();

                        union
                        {
                                NEW_ORDER_DATA
        NewOrder;
                                PAYMENT_DATA                Payment;
                                DELIVERY_DATA               Delivery;
                                STOCK_LEVEL_DATA            StockLevel;
                                ORDER_STATUS_DATA           OrderStatus;
                        }                                   m_txn;

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

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

```
                        void NewOrder             ();
                        void Payment              ();
                        void Delivery             ();
                        void StockLevel                        ();
                        void OrderStatus          ();

                        // these are public because they must be called from
the dblib err_handler and msg_hangler
                        // outside of the class
                        void SetDbLibError(int severity, int dberr, int oserr,
LPCSTR dberrstr, LPCSTR oserrstr);
                        void SetSqlError( int msgno, int msgstate, int
severity, LPCSTR msgtext );

};

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

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

db_odbc_dll/db_odbc_dll.dsp

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

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

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

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

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

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

odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib

odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
/out:".\bin/tpcc_odbc.dll"

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

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

odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386
/pdbtype:sept
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib

odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386
/out:".\bin/tpcc_odbc.dll" /pdbtype:sept

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "db_odbc_"
# PROP BASE Intermediate_Dir "db_odbc_"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D
"_WINDOWS" /YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /Gm /GX /Zi /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS"
/D "ICECAP" /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib

odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386
/out:".\bin/tpcc_odbc.dll" /pdbtype:sept
# ADD LINK32 icap.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib

odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386
/out:".\bin/tpcc_odbc.dll" /pdbtype:sept

!ENDIF

# Begin Target

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

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

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

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

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

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

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

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

db_odbc_dll/src/tpcc_odbc.cpp

```
/*            FILE:                        TPCC_ODBC.CPP
 *                                                      Microsoft TPC-C Kit Ver.
4.20.000
 *                                                      Copyright Microsoft, 1999
 *            All Rights Reserved
```

```
 *
 *                                                  Version 4.10.000 audited
by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *             PURPOSE:      Implements ODBC calls for TPC-C txns.
 *             Contact:      Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                           4.20.000 - updated rev number to match kit
 *                           4.10.001 - not deleting error class in catch handler
on deadlock retry;
 *                                      not a functional bug,
but a memory leak
 */

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

#define DBNTWIN32
#include <sqltypes.h>
#include <sql.h>
#include <sqlext.h>
#include <odbcss.h>

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

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

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

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

const iMaxRetries = 10;                 // how many retries on deadlock

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

static SQLHENV henv = SQL_NULL_HENV;
            // ODBC environment handle

BOOL APIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID
lpReserved)
{
            switch( ul_reason_for_call )
            {
                        case DLL_PROCESS_ATTACH:
                                    DisableThreadLibraryCalls(hModule);
                                    if ( SQLAllocHandleStd(SQL_HANDLE_ENV,
SQL_NULL_HANDLE, &henv) != SQL_SUCCESS )
                                                return FALSE;
                                    break;

                        case DLL_PROCESS_DETACH:
                                    if (henv != NULL)
                                                SQLFreeEnv(henv);
                                    break;

                        default:
                                    /* nothing */;
            }
            return TRUE;
}

/* FUNCTION: CTPCC_ODBC_ERR::ErrorText
 *
 */

char* CTPCC_ODBC_ERR::ErrorText(void)
{
            int i;

            static SERRORMSG errorMsgs[] =
            {
                        { ERR_WRONG_SP_VERSION,                 "Wrong
version of stored procs on database server"       },
                        { ERR_INVALID_CUST,
            "Invalid Customer id,name."
                                                                },
                        { ERR_NO_SUCH_ORDER,                    "No orders
found for customer."
                        },
                        { ERR_RETRIED_TRANS,                    "Retries
before transaction succeeded."
                        },
                        { 0,
                        ""


                                    }
            };

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

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

```
// wrapper routine for class constructor
__declspec(dllexport) CTPCC_ODBC* CTPCC_ODBC_new(
            LPCSTR szServer,                        // name of SQL server
            LPCSTR szUser,                                  // user name
for login
            LPCSTR szPassword,                      // password for login
            LPCSTR szHost,                                  // not used
            LPCSTR szDatabase )                     // name of database to use
{
            return new CTPCC_ODBC( szServer, szUser, szPassword, szHost,
szDatabase );
}


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

            // initialization
            m_hdbc = SQL_NULL_HDBC;
            m_hstmt = SQL_NULL_HSTMT;

            m_hstmtNewOrder = SQL_NULL_HSTMT;
            m_hstmtPayment = SQL_NULL_HSTMT;
            m_hstmtDelivery = SQL_NULL_HSTMT;
            m_hstmtOrderStatus = SQL_NULL_HSTMT;
            m_hstmtStockLevel = SQL_NULL_HSTMT;

            m_descNewOrderCols1 = SQL_NULL_HDESC;
            m_descNewOrderCols2 = SQL_NULL_HDESC;
            m_descOrderStatusCols1 = SQL_NULL_HDESC;
            m_descOrderStatusCols2 = SQL_NULL_HDESC;


            if ( SQLAllocHandle(SQL_HANDLE_DBC, henv, &m_hdbc) !=
SQL_SUCCESS )
                        ThrowError(CODBCERR::eAllocHandle);

            if ( SQLSetConnectOption(m_hdbc, SQL_PACKET_SIZE, 4096) !=
SQL_SUCCESS )
                        ThrowError(CODBCERR::eConnOption);

            {
                        char
            szConnectStr[256];
                        char
            szOutStr[1024];
                        SQLSMALLINT               iOutStrLen;

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

                        rc = SQLDriverConnect(m_hdbc, NULL,
(SQLCHAR*)szConnectStr, sizeof(szConnectStr),
                                    (SQLCHAR*)szOutStr, sizeof(szOutStr),
&iOutStrLen, SQL_DRIVER_NOPROMPT );

                        if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                                    ThrowError(CODBCERR::eConnect);
            }

            if (SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmt) !=
SQL_SUCCESS)
                        ThrowError(CODBCERR::eAllocHandle);

            {
                        char                        buffer[128];

                        // set some options affecting connection behavior
                        strcpy(buffer, "set nocount on set XACT_ABORT ON");
                        rc = SQLExecDirect(m_hstmt, (unsigned char *)buffer,
SQL_NTS);
                        if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                                    ThrowError(CODBCERR::eExecDirect);

                        // verify that version of stored procs on server is
correct
                        char db_sp_version[10];
                        strcpy(buffer, "{call tpcc_version}");
                        rc = SQLExecDirect(m_hstmt, (unsigned char *)buffer,
SQL_NTS);
                        if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                                    ThrowError(CODBCERR::eExecDirect);
                        if ( SQLBindCol(m_hstmt, 1, SQL_C_CHAR,
&db_sp_version, sizeof(db_sp_version), NULL) != SQL_SUCCESS )
                                    ThrowError(CODBCERR::eBindCol);
                        if ( SQLFetch(m_hstmt) == SQL_ERROR )
                                    ThrowError(CODBCERR::eFetch);
                        if (strcmp(db_sp_version,sVersion))
                                    throw new
CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_WRONG_SP_VERSION );

                        SQLFreeHandle(SQL_HANDLE_STMT, m_hstmt);
            }

            // Bind parameters for each of the transactions
            InitNewOrderParams();
            InitPaymentParams();
            InitOrderStatusParams();
            InitDeliveryParams();
            InitStockLevelParams();
}

CTPCC_ODBC::~CTPCC_ODBC( void )
{
            // note: descriptors are automatically released when the
connection is dropped
            SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtNewOrder);
            SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtPayment);
            SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtDelivery);
            SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtOrderStatus);
            SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtStockLevel);

            SQLDisconnect(m_hdbc);
            SQLFreeHandle(SQL_HANDLE_DBC, m_hdbc);
}

void CTPCC_ODBC::ThrowError( CODBCERR::ACTION eAction )
{
            RETCODE                     rc;
            SDWORD                      lNativeError;
            char                        szState[6];
            char                        szMsg[SQL_MAX_MESSAGE_LENGTH];
            char                        szTmp[6*SQL_MAX_MESSAGE_LENGTH];
            CODBCERR   *pODBCErr;                                   // not
allocated until needed (maybe never)

            pODBCErr = new CODBCERR();

            pODBCErr->m_NativeError = 0;
            pODBCErr->m_eAction = eAction;
            pODBCErr->m_bDeadLock = FALSE;

            szTmp[0] = 0;
            while (TRUE)
            {
                        rc = SQLError(henv, m_hdbc, m_hstmt, (BYTE *)&szState,
&lNativeError,
                                                                (BYTE
*)&szMsg, sizeof(szMsg), NULL);
                        if (rc == SQL_NO_DATA)
                                    break;

                        // check for deadlock
                        if (lNativeError == 1205 || (lNativeError ==
iErrOleDbProvider &&
                                    strstr(szMsg, sErrTimeoutExpired) !=
NULL))
                                    pODBCErr->m_bDeadLock = TRUE;

                        // capture the (first) database error
                        if (pODBCErr->m_NativeError == 0 && lNativeError !=
0)
                                    pODBCErr->m_NativeError = lNativeError;

                        // quit if there isn't enough room to concatenate
error text
                        if ( (strlen(szMsg) + 2) > (sizeof(szTmp) -
strlen(szTmp)) )
                                    break;

                        // include line break after first error msg
                        if (szTmp[0] != 0)
                                    strcat( szTmp, "\n");
                        strcat( szTmp, szMsg );
            }

            if (pODBCErr->m_odbcerrstr != NULL)
            {
                        delete [] pODBCErr->m_odbcerrstr;
                        pODBCErr->m_odbcerrstr = NULL;
            }

            if (strlen(szTmp) > 0)
            {
                        pODBCErr->m_odbcerrstr = new char[ strlen(szTmp)+1 ];
                        strcpy( pODBCErr->m_odbcerrstr, szTmp );
            }

            SQLFreeStmt(m_hstmt, SQL_CLOSE);
            throw pODBCErr;
}

void CTPCC_ODBC::InitStockLevelParams()
{
            if (   SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtStockLevel) != SQL_SUCCESS )
                        ThrowError(CODBCERR::eAllocHandle);

            m_hstmt = m_hstmtStockLevel;

            int i = 0;
            if (   SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SSHORT, SQL_SMALLINT, 0, 0, &m_txn.StockLevel.w_id, 0, NULL) !=
SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.StockLevel.d_id, 0, NULL) !=
SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SSHORT, SQL_SMALLINT, 0, 0, &m_txn.StockLevel.threshold, 0, NULL) !=
SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eBindParam);

            if ( SQLBindCol(m_hstmt, 1, SQL_C_SLONG,
&m_txn.StockLevel.low_stock, 0, NULL) != SQL_SUCCESS )
                        ThrowError(CODBCERR::eBindCol);
}
```

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

            m_hstmt = m_hstmtStockLevel;

            while (TRUE)
            {
                        try
                        {
                                    rc = SQLExecDirectW(m_hstmt,
(SQLWCHAR*)L"{call tpcc_stocklevel(?,?,?)}", SQL_NTS);
                                    if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

            ThrowError(CODBCERR::eExecDirect);

                                    if ( SQLFetch(m_hstmt) == SQL_ERROR )

            ThrowError(CODBCERR::eFetch);

                                    SQLFreeStmt(m_hstmt, SQL_CLOSE);

                                    m_txn.StockLevel.exec_status_code = eOK;
                                    break;
                        }
                        catch (CODBCERR *e)
                        {
                                    if ((!e->m_bDeadLock) || (++iTryCount >
iMaxRetries))
                                                throw;

                                    // hit deadlock; backoff for
increasingly longer period
                                    delete e;
                                    Sleep(10 * iTryCount);
                        }
            }

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

void CTPCC_ODBC::InitNewOrderParams()
{
            if (   SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtNewOrder) != SQL_SUCCESS
                        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderCols1) != SQL_SUCCESS
                        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderCols2) != SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eAllocHandle);

            m_hstmt = m_hstmtNewOrder;

            if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descNewOrderCols1, SQL_IS_POINTER ) != SQL_SUCCESS )
                        ThrowError(CODBCERR::eSetStmtAttr);

            int i = 0;
            if (   SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SSHORT, SQL_SMALLINT, 0, 0, &m_txn.NewOrder.w_id, 0, NULL) != SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.NewOrder.d_id, 0, NULL) !=

SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SLONG, SQL_INTEGER, 0, 0, &m_txn.NewOrder.c_id, 0, NULL) !=

SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.NewOrder.o_ol_cnt, 0, NULL) !=

SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.NewOrder.o_all_local, 0, NULL) !=

SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eBindParam);

            for (int j=0; j<MAX_OL_NEW_ORDER_ITEMS; j++)
            {
                        if (   SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SLONG, SQL_INTEGER, 0, 0, &m_txn.NewOrder.OL[j].ol_i_id, 0, NULL)

!= SQL_SUCCESS
                                    || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,

&m_txn.NewOrder.OL[j].ol_supply_w_id, 0, NULL) != SQL_SUCCESS
                                    || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
&m_txn.NewOrder.OL[j].ol_quantity,

0, NULL) != SQL_SUCCESS
                                    )
                                    ThrowError(CODBCERR::eBindParam);
            }

            // set the bind offset pointer
            if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_ROW_BIND_OFFSET_PTR,
&m_BindOffset, SQL_IS_POINTER ) != SQL_SUCCESS )
                        ThrowError(CODBCERR::eSetStmtAttr);

            i = 0;
            if (   SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.OL[0].ol_i_name, sizeof(m_txn.NewOrder.OL[0].ol_i_name),
NULL) !=

SQL_SUCCESS
                                    || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.NewOrder.OL[0].ol_stock, 0, NULL) != SQL_SUCCESS
                                    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.OL[0].ol_brand_generic,

sizeof(m_txn.NewOrder.OL[0].ol_brand_generic), NULL) != SQL_SUCCESS
                                    || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.OL[0].ol_i_price, 0, NULL) != SQL_SUCCESS
                                    || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.OL[0].ol_amount, 0, NULL) != SQL_SUCCESS
                                    )
                                    ThrowError(CODBCERR::eBindCol);

            // associate the column bindings for the second result set
            if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descNewOrderCols2, SQL_IS_POINTER ) != SQL_SUCCESS )
                        ThrowError(CODBCERR::eSetStmtAttr);

            i = 0;
            if (   SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.w_tax, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.d_tax, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.NewOrder.o_id, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.c_last, sizeof(m_txn.NewOrder.c_last), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.c_discount, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.c_credit, sizeof(m_txn.NewOrder.c_credit), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.NewOrder.o_entry_d, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_no_commit_flag, 0, NULL) != SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eBindCol);
}

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

                                                                    // 0         1         2
                                                                    //
0123456789012345678901234567890123456789
            wchar_t
            szSqlTemplate[] = L"{call tpcc_neworder(?,?,?,?,?,"

L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,"

L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,"

L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)}";

            m_hstmt = m_hstmtNewOrder;

            // associate the parameter and column bindings for this
transaction
            if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descNewOrderCols1, SQL_IS_POINTER ) != SQL_SUCCESS )
                        ThrowError(CODBCERR::eSetStmtAttr);

            // clip statement buffer based on number of parameters
            // fixed part is 29 chars and variable part is 6 chars per line
item
            i = 29 + m_txn.NewOrder.o_ol_cnt*6;
            wcscpy( &szSqlTemplate[i], L")}" );

            // check whether any order lines are for a remote warehouse
            m_txn.NewOrder.o_all_local = 1;
            for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++)
            {
                        if (m_txn.NewOrder.OL[i].ol_supply_w_id !=
m_txn.NewOrder.w_id)
                        {
                                    m_txn.NewOrder.o_all_local = 0;  // at
least one remote warehouse
                                    break;
                        }
            }

            while (TRUE)
            {
                        try
                        {
                                    m_BindOffset = 0;
                                    rc = SQLExecDirectW(m_hstmt,
(SQLWCHAR*)szSqlTemplate, SQL_NTS);
                                    if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

            ThrowError(CODBCERR::eExecDirect);

                                    // Get order line results
                                    m_txn.NewOrder.total_amount = 0;
                                    for (i = 0; i<m_txn.NewOrder.o_ol_cnt;
i++)
                                    {
                                                // set the bind offset
value...
                                                m_BindOffset = i *
sizeof(m_txn.NewOrder.OL[0]);

                                    if ( SQLFetch(m_hstmt) ==
SQL_ERROR)

            ThrowError(CODBCERR::eFetch);

                                    // move to the next
resultset
                                    if
( SQLMoreResults(m_hstmt) == SQL_ERROR )

            ThrowError(CODBCERR::eMoreResults);


            m_txn.NewOrder.total_amount += m_txn.NewOrder.OL[i].ol_amount;
                                    }
                                    // associate the column bindings for the
second result set
                                    if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descNewOrderCols2, SQL_IS_POINTER ) != SQL_SUCCESS )

            ThrowError(CODBCERR::eSetStmtAttr);

                                    if ( SQLFetch(m_hstmt) == SQL_ERROR)

            ThrowError(CODBCERR::eFetch);

                                    SQLFreeStmt(m_hstmt, SQL_CLOSE);

                                    if (m_no_commit_flag == 1)
                                    {

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

m_txn.NewOrder.c_discount));

            m_txn.NewOrder.exec_status_code = eOK;
                                    }
                                    else

            m_txn.NewOrder.exec_status_code = eInvalidItem;

                                    break;
                        }
                        catch (CODBCERR *e)
                        {
                                    if ((!e->m_bDeadLock) || (++iTryCount >
iMaxRetries))
                                                throw;

                                    // hit deadlock; backoff for
increasingly longer period
                                    delete e;
                                    Sleep(10 * iTryCount);
                        }
            }

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

void CTPCC_ODBC::InitPaymentParams()
{
            if (   SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmtPayment) !=
SQL_SUCCESS )
                        ThrowError(CODBCERR::eAllocHandle);

            m_hstmt = m_hstmtPayment;

            int i = 0;
            if (   SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SSHORT, SQL_SMALLINT, 0, 0, &m_txn.Payment.w_id, 0, NULL) != SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SSHORT, SQL_SMALLINT, 0, 0, &m_txn.Payment.c_w_id, 0, NULL) !=

SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_DOUBLE, SQL_NUMERIC, 6, 2, &m_txn.Payment.h_amount, 0, NULL) !=

SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.Payment.d_id, 0, NULL) !=

SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.Payment.c_d_id, 0, NULL) !=

SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SLONG, SQL_INTEGER, 0, 0, &m_txn.Payment.c_id, 0, NULL) != SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_CHAR, SQL_CHAR, sizeof(m_txn.Payment.c_last), 0,

&m_txn.Payment.c_last, sizeof(m_txn.Payment.c_last), NULL) != SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eBindParam);

            i = 0;
            if (   SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.Payment.c_id,                    0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_last,                  sizeof(m_txn.Payment.c_last), NULL) !=

SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.Payment.h_date,                  0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
```

```
&m_txn.Payment.w_street_1, sizeof(m_txn.Payment.w_street_1), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_street_2, sizeof(m_txn.Payment.w_street_2), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_city,                  sizeof(m_txn.Payment.w_city), NULL) !=

SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_state,                 sizeof(m_txn.Payment.w_state), NULL) !=

SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_zip,                   sizeof(m_txn.Payment.w_zip), NULL) !=

SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_street_1, sizeof(m_txn.Payment.d_street_1), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_street_2, sizeof(m_txn.Payment.d_street_2), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_city,                  sizeof(m_txn.Payment.d_city), NULL) !=

SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_state,                 sizeof(m_txn.Payment.d_state), NULL) !=

SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_zip,                   sizeof(m_txn.Payment.d_zip), NULL) !=

SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_first,                 sizeof(m_txn.Payment.c_first), NULL) !=

SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_middle,  sizeof(m_txn.Payment.c_middle), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_street_1, sizeof(m_txn.Payment.c_street_1), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_street_2, sizeof(m_txn.Payment.c_street_2), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_city,                  sizeof(m_txn.Payment.c_city), NULL) !=

SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_state,                 sizeof(m_txn.Payment.c_state), NULL) !=

SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_zip,                   sizeof(m_txn.Payment.c_zip), NULL) !=

SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_phone,                 sizeof(m_txn.Payment.c_phone), NULL) !=

SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.Payment.c_since,                 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_credit,  sizeof(m_txn.Payment.c_credit), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.Payment.c_credit_lim,0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.Payment.c_discount, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.Payment.c_balance,  0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_data,                  sizeof(m_txn.Payment.c_data), NULL) !=

SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eBindCol);
}

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

        m_hstmt = m_hstmtPayment;

        if (m_txn.Payment.c_id != 0)
                m_txn.Payment.c_last[0] = 0;

        while (TRUE)
        {
                try
                {
                        rc = SQLExecDirectW(m_hstmt,
(SQLWCHAR*)L"{call tpcc_payment(?,?,?,?,?,?,?,?)}", SQL_NTS);
                        if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

        ThrowError(CODBCERR::eExecDirect);

                        if ( SQLFetch(m_hstmt) == SQL_ERROR)

        ThrowError(CODBCERR::eFetch);

                        SQLFreeStmt(m_hstmt, SQL_CLOSE);

                        if (m_txn.Payment.c_id == 0)
                                throw new
CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_INVALID_CUST );
                        else
                m_txn.Payment.exec_status_code = eOK;

                        break;
                }
                catch (CODBCERR *e)
                {
                        if ((!e->m_bDeadLock) || (++iTryCount >
iMaxRetries))
                                throw;

                        // hit deadlock; backoff for
increasingly longer period
                        delete e;
                        Sleep(10 * iTryCount);
                }
        }

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

void CTPCC_ODBC::InitOrderStatusParams()
{
        if (   SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtOrderStatus) != SQL_SUCCESS
                        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols1) != SQL_SUCCESS
                        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols2) != SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eAllocHandle);

        m_hstmt = m_hstmtOrderStatus;

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

        int i = 0;
        if (   SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SSHORT, SQL_SMALLINT, 0, 0, &m_txn.OrderStatus.w_id, 0, NULL) !=

SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.OrderStatus.d_id, 0, NULL) !=

SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SLONG, SQL_INTEGER, 0, 0, &m_txn.OrderStatus.c_id, 0, NULL) !=

SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_CHAR, SQL_CHAR, sizeof(m_txn.OrderStatus.c_last), 0,

&m_txn.OrderStatus.c_last, sizeof(m_txn.OrderStatus.c_last), NULL) !=
SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eBindParam);

        // configure block cursor
        if (   SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_BIND_TYPE,
(SQLPOINTER)sizeof(m_txn.OrderStatus.OL[0]), 0) != SQL_SUCCESS
                        || SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROWS_FETCHED_PTR,
&m_RowsFetched, 0) != SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eSetStmtAttr);

        i = 0;
        if (   SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.OrderStatus.OL[0].ol_supply_w_id, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.OL[0].ol_i_id, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.OrderStatus.OL[0].ol_quantity, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.OrderStatus.OL[0].ol_amount, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.OrderStatus.OL[0].ol_delivery_d, 0, NULL) != SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eBindCol);


        if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols2, SQL_IS_POINTER ) != SQL_SUCCESS )
                        ThrowError(CODBCERR::eSetStmtAttr);

        i = 0;
        if (   SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.c_id, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_last, sizeof(m_txn.OrderStatus.c_last), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_first, sizeof(m_txn.OrderStatus.c_first), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_middle, sizeof(m_txn.OrderStatus.c_middle), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.OrderStatus.o_entry_d, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.OrderStatus.o_carrier_id, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.OrderStatus.c_balance, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.o_id, 0, NULL) != SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eBindCol);
}

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

        m_hstmt = m_hstmtOrderStatus;

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

        if (m_txn.OrderStatus.c_id != 0)
                        m_txn.OrderStatus.c_last[0] = 0;

        while (TRUE)
        {
                try
                {
                        // configure block cursor
                        if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_ARRAY_SIZE, (SQLPOINTER)1, 0) != SQL_SUCCESS )

        ThrowError(CODBCERR::eSetStmtAttr);

                        rc = SQLExecDirectW(m_hstmt,
(SQLWCHAR*)L"{call tpcc_orderstatus(?,?,?,?)}", SQL_NTS);
                        if ( ((rc == SQL_SUCCESS_WITH_INFO) &&
(m_RowsFetched != 0)) || (rc == SQL_ERROR) )

        ThrowError(CODBCERR::eExecDirect);

                        // configure block cursor
                        if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_ARRAY_SIZE, (SQLPOINTER)MAX_OL_ORDER_STATUS_ITEMS, 0) !=
SQL_SUCCESS )

        ThrowError(CODBCERR::eSetStmtAttr);

                        rc = SQLFetchScroll( m_hstmt,
SQL_FETCH_NEXT, 0 );
                        if ( ((rc == SQL_SUCCESS_WITH_INFO) &&
(m_RowsFetched != 0)) || (rc == SQL_ERROR) )

        ThrowError(CODBCERR::eFetchScroll);

                        m_txn.OrderStatus.o_ol_cnt =
(short)m_RowsFetched;

                        if (m_txn.OrderStatus.o_ol_cnt != 0)
                        {
                                if
( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols2,
SQL_IS_POINTER ) != SQL_SUCCESS )

        ThrowError(CODBCERR::eSetStmtAttr);

                                if
( SQLMoreResults(m_hstmt) == SQL_ERROR )

        ThrowError(CODBCERR::eMoreResults);

                                if ( (rc =
SQLFetch(m_hstmt)) == SQL_ERROR)

        ThrowError(CODBCERR::eFetch);
                                }

                        SQLFreeStmt(m_hstmt, SQL_CLOSE);

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

                m_txn.OrderStatus.exec_status_code = eOK;

                        break;
                }
                catch (CODBCERR *e)
                {
                        if ((!e->m_bDeadLock) || (++iTryCount >
iMaxRetries))
                                throw;

                        // hit deadlock; backoff for
increasingly longer period
                        delete e;
                        Sleep(10 * iTryCount);
                }
        }

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

void CTPCC_ODBC::InitDeliveryParams()
{
        if (   SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtDelivery) != SQL_SUCCESS )
                        ThrowError(CODBCERR::eAllocHandle);

        m_hstmt = m_hstmtDelivery;

        int i = 0;
        if (   SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
```

```
SQL_C_SSHORT, SQL_SMALLINT, 0, 0, &m_txn.Delivery.w_id, 0, NULL) != SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SSHORT, SQL_SMALLINT, 0, 0, &m_txn.Delivery.o_carrier_id, 0, NULL) !=

SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eBindParam);

            for (i=0;i<10;i++)
            {
                        if ( SQLBindCol(m_hstmt, (UWORD)(i+1), SQL_C_SLONG,
&m_txn.Delivery.o_id[i], 0, NULL) != SQL_SUCCESS )
                                    ThrowError(CODBCERR::eBindCol);
            }
}

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

            m_hstmt = m_hstmtDelivery;

            while (TRUE)
            {
                        try
                        {
                                    rc = SQLExecDirectW(m_hstmt,
(SQLWCHAR*)L"{call tpcc_delivery(?,?)}", SQL_NTS);
                                    if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

            ThrowError(CODBCERR::eExecDirect);

                                    if ( SQLFetch(m_hstmt) == SQL_ERROR )

            ThrowError(CODBCERR::eFetch);

                                    SQLFreeStmt(m_hstmt, SQL_CLOSE);
                                    m_txn.Delivery.exec_status_code = eOK;
                                    break;
                        }
                        catch (CODBCERR *e)
                        {
                                    if ((!e->m_bDeadLock) || (++iTryCount >
iMaxRetries))
                                                throw;

                                    // hit deadlock; backoff for
increasingly longer period
                                    delete e;
                                    Sleep(10 * iTryCount);
                        }
            }

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

db_odbc_dll/src/tpcc_odbc.h

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

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

class CODBCERR : public CBaseErr
{
            public:
                        enum ACTION
                        {
                                    eNone,
                                    eUnknown,
                                    eAllocConn,
            // error from SQLAllocConnect       eAllocHandle,           // error from
SQLAllocHandle
                                    eConnOption,            // error from
SQLSetConnectOption
                                    eConnect,
            // error from SQLConnect
                                    eAllocStmt,
            // error from SQLAllocStmt
                                    eExecDirect,            // error from
SQLExecDirect
                                    eBindParam,
            // error from SQLBindParameter
                                    eBindCol,
            // error from SQLBindCol
                                    eFetch,
            // error from SQLFetch
                                    eFetchScroll,           // error from
SQLFetchScroll
                                    eMoreResults,           // error from
SQLMoreResults
                                    ePrepare,
            // error from SQLPrepare
                                    eExecute,
            // error from SQLExecute
                                    eSetEnvAttr,            // error from
SQLSetEnvAttr
                                    eSetStmtAttr            // error from
SQLSetStmtAttr
                        };

                        CODBCERR(void)
                        {
                                    m_eAction = eNone;
                                    m_NativeError = 0;
                                    m_bDeadLock = FALSE;
                                    m_odbcerrstr = NULL;
                        };

                        ~CODBCERR()
                        {
                                    if (m_odbcerrstr != NULL)
                                                delete [] m_odbcerrstr;
                        };

                        ACTION      m_eAction;
                        int                     m_NativeError;
                        BOOL        m_bDeadLock;
                        char    *m_odbcerrstr;

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

};

class CTPCC_ODBC_ERR : public CBaseErr
{
            public:
                        enum TPCC_ODBC_ERRS
                        {
                                    ERR_WRONG_SP_VERSION = 1,  // "Wrong
version of stored procs on database server"
                                    ERR_INVALID_CUST,
            // "Invalid Customer id,name."
                                    ERR_NO_SUCH_ORDER,
            // "No orders found for customer."
                                    ERR_RETRIED_TRANS,
            // "Retries before transaction succeeded."
                        };

                        CTPCC_ODBC_ERR( int iErr ) { m_errno = iErr;
m_iTryCount = 0; };

                        CTPCC_ODBC_ERR( int iErr, int iTryCount ) { m_errno =
iErr; m_iTryCount = iTryCount; };

                        int                     m_errno;
                        int                     m_iTryCount;

                        int ErrorType() {return ERR_TYPE_TPCC_ODBC;};
                        int ErrorNum() {return m_errno;};

                        char *ErrorText();
};

class DllDecl CTPCC_ODBC : public CTPCC_BASE
{
            private:
                        // declare variables and private functions here...
                        BOOL                    m_bDeadlock;
            // transaction was selected as deadlock victim
                        int                                 m_MaxRetries;
            // retry count on deadlock

                        SQLHENV                 m_henv;
                        // ODBC environment handle
                        SQLHDBC                 m_hdbc;
                        SQLHSTMT    m_hstmt;
            // the current hstmt

                        SQLHSTMT    m_hstmtNewOrder;
                        SQLHSTMT    m_hstmtPayment;
                        SQLHSTMT    m_hstmtDelivery;
                        SQLHSTMT    m_hstmtOrderStatus;
                        SQLHSTMT    m_hstmtStockLevel;

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

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

                        void ThrowError( CODBCERR::ACTION eAction );

                        void InitNewOrderParams();
                        void InitPaymentParams();
                        void InitDeliveryParams();
                        void InitStockLevelParams();
                        void InitOrderStatusParams();

                        union
                        {
                                    NEW_ORDER_DATA
            NewOrder;
                                    PAYMENT_DATA            Payment;
                                    DELIVERY_DATA           Delivery;
                                    STOCK_LEVEL_DATA        StockLevel;
                                    ORDER_STATUS_DATA       OrderStatus;
                        }                                   m_txn;

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

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

                        void NewOrder           ();
                        void Payment            ();
                        void Delivery           ();
                        void StockLevel                     ();
                        void OrderStatus        ();

};


// wrapper routine for class constructor
extern "C" DllDecl CTPCC_ODBC* CTPCC_ODBC_new
                        ( LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword, LPCSTR szHost,
LPCSTR szDatabase );

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

install/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 **

# TARGTYPE "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
!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 "install.mak" CFG="install - Win32 Release"
!MESSAGE
!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
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib

odbc32.lib odbccp32.lib /nologo /subsystem:windows /machine:I386
# ADD LINK32 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 /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
# 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;hpj;bat;for;f90"
# Begin Source File

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

SOURCE=.\src\install.rc
# ADD BASE RSC /l 0x409 /i "src"
# ADD RSC /l 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=..\db_dblib_dll\bin\tpcc_dblib.dll
# End Source File
# Begin Source File

SOURCE=..\db_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
```

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                               versionDllMS;
DWORD                               versionDllLS;

// TPC-C registry settings
TPCCREGISTRYDATA        Reg;

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

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


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

BOOL install_com(char *szDllPath);

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

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

            hInst = hInstance;

            InitCommonControls();

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

            iRc = DialogBox(hInstance, MAKEINTRESOURCE(IDD_DIALOG4),
GetDesktopWindow(), LicenseDlgProc);
            if ( iRc )
            {
                        iRc = DialogBox(hInstance,
MAKEINTRESOURCE(IDD_DIALOG1), GetDesktopWindow(), MainDlgProc);
                        if ( iRc )
                        {
                                    DialogBoxParam(hInstance,
MAKEINTRESOURCE(IDD_DIALOG2), GetDesktopWindow(), UpdatedDlgProc, (LPARAM)iRc);
                        }
            }
            DestroyIcon(hIcon);
            return 0;
}

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

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

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

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

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

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

                                    if ( GetInstallPath(szDllPath) )
                                    {
                                                MessageBox(hwnd, "Error
internet service inetsrv is not installed.", NULL, MB_ICONSTOP | MB_OK);
                                                EndDialog(hwnd, FALSE);
                                                return TRUE;
                                    }

                                    // set default values
                                    ZeroMemory( &Reg, sizeof(Reg) );
                                    Reg.dwNumberOfDeliveryThreads = 4;
                                    Reg.dwMaxConnections = 100;
                                    Reg.dwMaxPendingDeliveries = 100;
                                    Reg.eDB_Protocol = DBLIB;
```

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

                                iPoolThreadLimit =  iMaxPhysicalMemory *
2;
                                iThreadTimeout = 86400;
                                iListenBackLog = 15;
                                iAcceptExOutstanding = 40;

                                ReadTPCCRegistrySettings( &Reg );
                                ReadRegistrySettings();

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

                               wsprintf(szTmp, "Version %d.%2.2d.%3.3d",
versionExeMS, versionExeMM, versionExeLS);
                                SetDlgItemText(hwnd, IDC_VERSION,
szTmp);

                                SetDlgItemText(hwnd, IDC_PATH,
szDllPath);

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

                                SetDlgItemInt(hwnd, ED_THREADS,
Reg.dwNumberOfDeliveryThreads, FALSE);
                                SetDlgItemInt(hwnd, ED_MAXCONNECTION,
Reg.dwMaxConnections, FALSE);
                                SetDlgItemInt(hwnd, ED_MAXDELIVERIES,
Reg.dwMaxPendingDeliveries, FALSE);
                                SetDlgItemInt(hwnd,
ED_IIS_MAX_THEAD_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;

                                                case IDOK:

                                                ProcessOK(hwnd, szDllPath);

                                                return TRUE;

                                                case
IDCANCEL:

                                                EndDialog(hwnd, FALSE);

                                                return TRUE;

                                                default:

                                                return FALSE;

                                        }
                                        break;
                                default:
                                        break;
                        }
                        return FALSE;
}

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

                        char            szFullName[256];
                        char            szErrTxt[128];

                        // read settings from dialog
                        Reg.dwNumberOfDeliveryThreads  = GetDlgItemInt(hwnd, ED_THREADS,
&d, FALSE);
                        Reg.dwMaxConnections = GetDlgItemInt(hwnd, ED_MAXCONNECTION, &d,
FALSE);
                        Reg.dwMaxPendingDeliveries = GetDlgItemInt(hwnd, ED_MAXDELIVERIES,
&d, FALSE);

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

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

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

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


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


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

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

                        // register com proxy stub
                        strcpy(szFullName, szDllPath);
                        strcat(szFullName, "tpcc_com_ps.dll");
                        if (!RegisterDLL(szFullName))
                        {
                                ShowWindow(hwnd, SW_SHOWNA);
                                DestroyWindow(hDlg);
                                strcpy( szErrTxt, "Error occured when registering
" );
                                strcat( szErrTxt, szFullName );
                                MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP |
MB_OK);
                                EndDialog(hwnd, 0);
                                return;
                        }

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

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

                        Sleep(100);

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

                        EndDialog(hwnd, rc);
                        return;
}

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

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

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

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

                                size = sizeof(iListenBackLog);
                                if ( RegQueryValueEx(hKey, "ListenBackLog", 0, &type,
(char *)&iListenBackLog, &size) == ERROR_SUCCESS )
                                        if ( !iListenBackLog )
                                                iListenBackLog = 15;

                                RegCloseKey(hKey);
                        }

                        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters", 0, KEY_READ, &hKey) ==
ERROR_SUCCESS )
                        {
                                size = sizeof(iAcceptExOutstanding);
                                if ( RegQueryValueEx(hKey, "AcceptExOutstanding", 0,
&type, (char *)&iAcceptExOutstanding, &size) == ERROR_SUCCESS )
                                        if ( !iAcceptExOutstanding )
                                                iAcceptExOutstanding = 40;

                                RegCloseKey(hKey);
                        }
}

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

```
            if ( RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE\\Microsoft\\TPCC", 0, NULL, REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS,
NULL, &hKey,

&dwDisposition) == ERROR_SUCCESS )
            {
                        strcpy(szTmp, szDllPath);
                        ptr = strstr(szTmp, "tpcc");
                        if ( ptr )
                                    *ptr = 0;

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

                        RegSetValueEx(hKey, "NumberOfDeliveryThreads", 0,
REG_DWORD, (char *)&Reg.dwNumberOfDeliveryThreads,

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

sizeof(Reg.dwMaxPendingDeliveries));

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

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

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

                        RegFlushKey(hKey);
                        RegCloseKey(hKey);
            }

            if ( (iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Parameters", 0, NULL,

REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition)) ==
ERROR_SUCCESS )
            {
                        RegSetValueEx(hKey, "PoolThreadLimit", 0, REG_DWORD,
(char *)&iPoolThreadLimit, sizeof(iPoolThreadLimit));
                        RegSetValueEx(hKey, "ThreadTimeout", 0, REG_DWORD,
(char *)&iThreadTimeout, sizeof(iThreadTimeout));
                        RegSetValueEx(hKey, "ListenBackLog", 0, REG_DWORD,
(char *)&iListenBackLog, sizeof(iListenBackLog));

                        RegFlushKey(hKey);
                        RegCloseKey(hKey);
            }

            if ( (iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters", 0, NULL,

REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition)) ==
ERROR_SUCCESS )
            {
                        RegSetValueEx(hKey, "AcceptExOutstanding", 0,
REG_DWORD, (char *)&iAcceptExOutstanding, sizeof(iAcceptExOutstanding));

                        RegFlushKey(hKey);
                        RegCloseKey(hKey);
            }

            return;
}

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

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

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

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

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

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

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

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

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

            CloseHandle(hFile);

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

            return 1;
}

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


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

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

                        RegCloseKey(hKey);
            }

            return bRc;
}

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

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

            versionExeMS = 0x7FFF;
            versionExeLS = 0x7FFF;
            dwSize = GetFileVersionInfoSize(szExePath, &d);
            if ( dwSize )
            {
                        ptr = (char *)malloc(dwSize);
                        GetFileVersionInfo(szExePath, 0, dwSize, ptr);
                        VerQueryValue(ptr, "\\",&vs, &dwBytes);

                        versionExeMS = vs->dwProductVersionMS;
                        versionExeLS = LOWORD(vs->dwProductVersionLS);
```

```
                        versionExeMM = HIWORD(vs->dwProductVersionLS);
                        free(ptr);
                }
                return;
}

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

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

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

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

                CloseServiceHandle(schService);
                return TRUE;

ServiceNotRunning:

                CloseServiceHandle(schService);
                return FALSE;
}

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

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

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

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

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

                CloseServiceHandle(schService);
                return TRUE;

StartWWWebErr:
                CloseServiceHandle(schService);
                return FALSE;
}

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

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

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

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

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

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

                CloseServiceHandle(schService);
                return TRUE;

StopWWWebErr:
                CloseServiceHandle(schService);
                return FALSE;
}

static void UpdateDialog(HWND hDlg)
{
                MSG msg;

                UpdateWindow(hDlg);
                while( PeekMessage(&msg, hDlg, 0, 0, PM_REMOVE) )
                {
                        TranslateMessage(&msg);
                        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_MAXCONNECTION                                1014
#define ED_IIS_MAX_THEAD_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

#if !defined(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 "TPC-C Web Client Installation Utility"
FONT 8, "MS Sans Serif"
BEGIN
    EDITTEXT        ED_THREADS,164,45,34,12,ES_RIGHT | ES_NUMBER,
                    WS_EX_RTLREADING
    EDITTEXT        ED_MAXDELIVERIES,164,59,34,12,ES_RIGHT | ES_NUMBER,
                    WS_EX_RTLREADING
    EDITTEXT        ED_MAXCONNECTION,164,73,34,12,ES_RIGHT | ES_NUMBER,
                    WS_EX_RTLREADING
    CONTROL         "None",IDC_TM_NONE,"Button",BS_AUTORADIOBUTTON |
                    WS_GROUP | WS_TABSTOP,43,100,33,10
    CONTROL         "COM",IDC_TM_MTS,"Button",BS_AUTORADIOBUTTON |
                    WS_TABSTOP,43,113,32,10
    CONTROL         "TUXEDO",IDC_TM_TUXEDO,"Button",BS_AUTORADIOBUTTON |
                    WS_TABSTOP,106,100,46,10
    CONTROL         "ENCINA",IDC_TM_ENCINA,"Button",BS_AUTORADIOBUTTON |
                    WS_DISABLED | WS_TABSTOP,106,113,43,10
    EDITTEXT        ED_DB_SERVER,131,152,67,12,ES_AUTOHSCROLL
    EDITTEXT        ED_DB_USER_ID,131,165,67,12,ES_AUTOHSCROLL
    EDITTEXT        ED_DB_PASSWORD,131,178,67,12,ES_AUTOHSCROLL
    EDITTEXT        ED_DB_NAME,131,191,67,12,ES_AUTOHSCROLL
    CONTROL         "DBLIB",IDC_DBLIB,"Button",BS_AUTORADIOBUTTON | WS_GROUP |
                    WS_TABSTOP,45,219,39,12
    CONTROL         "ODBC",IDC_ODBC,"Button",BS_AUTORADIOBUTTON | WS_TABSTOP,
                    91,219,39,12
    EDITTEXT        ED_IIS_MAX_THEAD_POOL_LIMIT,164,263,34,12,ES_RIGHT |
                    ES_NUMBER,WS_EX_RTLREADING
    EDITTEXT        ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,164,277,34,12,ES_RIGHT |
                    ES_NUMBER,WS_EX_RTLREADING
    EDITTEXT        ED_IIS_THREAD_TIMEOUT,164,291,34,12,ES_RIGHT | ES_NUMBER,
                    WS_EX_RTLREADING
    EDITTEXT        ED_IIS_LISTEN_BACKLOG,164,305,34,12,ES_RIGHT | ES_NUMBER,
                    WS_EX_RTLREADING
    DEFPUSHBUTTON   "OK",IDOK,53,331,50,14
    PUSHBUTTON      "Cancel",IDCANCEL,119,331,50,14
    EDITTEXT        IDC_PATH,106,26,91,13,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "Number of Delivery Threads:",IDC_STATIC,35,45,115,12
    LTEXT           "Max Number of Connections:",IDC_STATIC,35,73,115,12
    RTEXT           "Version 4.11",IDC_VERSION,120,4,89,9
    LTEXT           "IIS Max Thread Pool Limit:",IDC_STATIC,36,263,115,12
    LTEXT           "Web Service Backlog Queue Size:",IDC_STATIC,36,277,115,
                    12
    LTEXT           "IIS Thread Timeout (seconds):",IDC_STATIC,36,291,115,12
    LTEXT           "IIS Listen Backlog:",IDC_STATIC,36,307,115,10
    GROUPBOX        "Database Interface",IDC_STATIC,35,208,163,27,WS_GROUP
    LTEXT           "Installation directory:",IDC_STATIC,35,29,71,10
    GROUPBOX        "Transaction Monitor",IDC_STATIC,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 TPC-C Installation Successfull",IDC_RESULTS,7,22,
                    102,18,0,WS_EX_CLIENTEDGE
    ICON            IDI_ICON2,IDC_STATIC,50,7,18,20,SS_REALSIZEIMAGE,
                    WS_EX_TRANSPARENT
END

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

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


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

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
    IDD_DIALOG1, DIALOG
    BEGIN
        LEFTMARGIN, 22
        RIGHTMARGIN, 209
        VERTGUIDE, 35
        VERTGUIDE, 198
```

```
        TOPMARGIN, 4
        BOTTOMMARGIN, 345
    END

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

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

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


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

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

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

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

#endif    // APSTUDIO_INVOKED


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

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

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

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

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

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

#endif    // !_MAC


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

IDR_LICENSE1            LICENSE DISCARDABLE     "license.txt"

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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


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


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

install/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*         pCatalogCollectionItf    = NULL;
            ICatalogCollection*         pCatalogCollectionMethod = NULL;

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

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


            CoInitializeEx(NULL, COINIT_MULTITHREADED);

            HRESULT hr = CoCreateInstance(CLSID_COMAdminCatalog,

                                          NULL,

                                          CLSCTX_INPROC_SERVER,

                                          IID_ICOMAdminCatalog,

                                          (void**) &pCOMAdminCat);

            if (!SUCCEEDED(hr)) goto Error;

            bstrTemp = "Applications";

            // Attempt to connect to "Applications" in the Catalog
            hr = pCOMAdminCat->GetCollection(bstrTemp,

                                          (IDispatch**) &pCatalogCollectionApp);
            if (!SUCCEEDED(hr)) goto Error;

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

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

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

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

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

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

            // add the new application
            hr = pCatalogCollectionApp->Add((IDispatch**) &pCatalogObjectApp);
            if (!SUCCEEDED(hr)) goto Error;
            // set properties
            bstrTemp = "Name";
            vTmp = "TPC-C";
            hr = pCatalogObjectApp->put_Value(bstrTemp, vTmp);
            if (!SUCCEEDED(hr)) goto Error;

            // set as a library (in process) application
            bstrTemp = "Activation";
```

```
            lActProp = COMAdminActivationInproc;
            vTmp = lActProp;
            hr = pCatalogObjectApp->put_Value(bstrTemp, vTmp);
            if (!SUCCEEDED(hr)) goto Error;

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

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

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

            pCatalogObjectApp->Release();
            pCatalogObjectApp = NULL;

            bstrTemp  = "TPC-C";
            // app name
            bstrTemp2 =  bstrDllPath + "tpcc_com_all.dll";
            // DLL
            bstrTemp3 =  bstrDllPath + "tpcc_com_all.tlb";
            // type library (TLB)
            bstrTemp4 =  bstrDllPath + "tpcc_com_ps.dll";
            // proxy/stub dll

            hr = pCOMAdminCat->InstallComponent(bstrTemp,
                                                bstrTemp2,
                                                bstrTemp3,
                                                bstrTemp4);
            if (!SUCCEEDED(hr)) goto Error;


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

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

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

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

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

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

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

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

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

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



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

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

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

                        hr = pCatalogCollectionItf->get_Count(&lCountItf);
                        if (!SUCCEEDED(hr)) goto Error;

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

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

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

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

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

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

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

                                                pCatalogObjectMethod-
>Release();
                                                pCatalogObjectMethod =
NULL;

                                                lCountMethod--;
                                    }

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

                                    pCatalogObjectItf->Release();
                                    pCatalogObjectItf = NULL;

                                    lCountItf--;

                        }


                        pCatalogObjectCo->Release();
                        pCatalogObjectCo = NULL;

                        lCountCo--;
            }


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

            pCatalogCollectionApp->Release();
            pCatalogCollectionApp = NULL;

            pCatalogCollectionCo->Release();
            pCatalogCollectionCo = NULL;

            pCatalogCollectionItf->Release();
            pCatalogCollectionItf = NULL;

            pCatalogCollectionMethod->Release();
            pCatalogCollectionMethod = NULL;


Error:
            CoUninitialize();

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

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 IDR_COMTYPLIB_DLL               129
#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_THEAD_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        130
#define _APS_NEXT_COMMAND_VALUE         40001
#define _APS_NEXT_CONTROL_VALUE         1031
#define _APS_NEXT_SYMED_VALUE           101
#endif
#endif
```

isapi_dll/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 **

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

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

# Begin Project
# PROP AllowPerConfigDependencies 0
```

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

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

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

odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
# ADD
LINK32 ..\common\txnlog\lib\release\rtetime.lib ..\common\txnlog\lib\release\spi
nlock.lib ..\common\txnlog\lib\release\error.lib

..\common\txnlog\lib\release\txnlog.lib wsock32.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib

ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /machine:I386 /nodefaultlib:"LIBCMT"

/out:".\bin/tpcc.dll"
# SUBTRACT LINK32 /nodefaultlib

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

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

odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386
/pdbtype:sept
# ADD
LINK32 ..\common\txnlog\lib\debug\rtetime.lib ..\common\txnlog\lib\debug\spinloc
k.lib ..\common\txnlog\lib\debug\error.lib

..\common\txnlog\lib\debug\txnlog.lib wsock32.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib

ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /nodefaultlib:"LIBCMTD"

/out:".\bin/tpcc.dll" /pdbtype:sept
# SUBTRACT LINK32 /profile /pdb:none /nodefaultlib

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "isapi_dl"
# PROP BASE Intermediate_Dir "isapi_dl"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /GX /Zi /Od /D "_DEBUG" /D "WIN32" /D "_WINDOWS"
/FR /YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /GX /Zi /O2 /D "NDEBUG" /D "ICECAP" /D "WIN32" /D
"_WINDOWS" /FR /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib

odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386
/pdbtype:sept
# ADD
LINK32 ..\common\txnlog\lib\release\rtetime.lib ..\common\txnlog\lib\release\s
pinlock.lib ..\common\txnlog\lib\release\error.lib

..\common\txnlog\lib\release\txnlog.lib wsock32.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib

ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc.dll"

/pdbtype:sept
# SUBTRACT LINK32 /profile /pdb:none /map

!ENDIF

# Begin Target

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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
```

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 defination of structures specific to TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\ReadRegistry.h"

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

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

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

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


#define       LEN_ERR_STRING              256

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

char                        szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

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

// The WEBCLIENT_VERSION string specifies the version level of this web client
interface.
// The RTE must be synchronized with the interface level on login, otherwise the
login
// will fail.  This is a sanity check to catch problems resulting from
mismatched versions
// of the RTE and web client.
#define WEBCLIENT_VERSION "410"

static        CRITICAL_SECTION                        TermCriticalSection;

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

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

// For deferred Delivery txns:

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

HANDLE
            hWorkerSemaphore           = INVALID_HANDLE_VALUE;
```

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

// configuration settings from registry
TPCCREGISTRYDATA                        Reg;


DWORD
            dwNumDeliveryThreads = 4;
CRITICAL_SECTION                        DelBuffCriticalSection;
            //critical section for delivery transactions cache
DELIVERY_TRANSACTION        *pDelBuff                           = NULL;
DWORD                                                           dwDelBuffSize
            = 100;                      // size of circular buffer for delivery
txns
DWORD
            dwDelBuffFreeCount;
            // number of buffers free
DWORD
            dwDelBuffBusyIndex          = 0;                    // index
position of entry waiting to be delivered
DWORD
            dwDelBuffFreeIndex          = 0;                    // index
position of unused entry

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

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

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

// debugging....
// DebugBreak();

            try
            {
                        switch( ul_reason_for_call )
                        {
                                    case DLL_PROCESS_ATTACH:
                                                {                           DWORD dwSize
= MAX_COMPUTERNAME_LENGTH+1;

            GetComputerName(szMyComputerName, &dwSize);

            szMyComputerName[dwSize] = 0;
                                                }


            DisableThreadLibraryCalls((HMODULE)hModule);

            InitializeCriticalSection(&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_CTPCC_TUXEDO*)
GetProcAddress(hLibInstanceTm,"CTPCC_TUXEDO_new");
                                                                        if
(pCTPCC_TUXEDO_new == NULL)

            throw new CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED, szDllName,
GetLastError() );
                                                }
                                                else if (Reg.eTxnMon ==
ENCINA)
                                                {

            strcpy( szDllName, Reg.szPath );

            strcat( szDllName, "tpcc_encina.dll");

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

            throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED, szDllName,
GetLastError() );
                                                                        // get
function pointer to wrapper for class constructor

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

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

            throw new CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED, szDllName,
GetLastError() );
                                                }
                                                else if (Reg.eTxnMon ==
COM)
                                                {

            strcpy( szDllName, Reg.szPath );

            strcat( szDllName, "tpcc_com.dll");

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

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

                                                                        // get
function pointer to wrapper for class constructor

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

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

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

            strcpy( szDllName, Reg.szPath );

            strcat( szDllName, "tpcc_dblib.dll");

            hLibInstanceDb = LoadLibrary( szDllName );

            if (hLibInstanceDb == NULL)

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


            // get function pointer to wrapper for class constructor

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

            if (pCTPCC_DBLIB_new == NULL)

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

            strcpy( szDllName, Reg.szPath );

            strcat( szDllName, "tpcc_odbc.dll");

            hLibInstanceDb = LoadLibrary( szDllName );

            if (hLibInstanceDb == NULL)

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

            // get function pointer to wrapper for class constructor

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

            if (pCTPCC_ODBC_new == NULL)

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

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

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

                                                                        // allocate
structures for delivery buffers and thread mgmt
                                                                        pDeliHandles
= new HANDLE[dwNumDeliveryThreads];
                                                                        pDelBuff =
new DELIVERY_TRANSACTION[dwDelBuffSize];
                                                                        // launch
DeliveryWorkerThread to perform actual delivery txns
                                                                        for(i=0;
i<dwNumDeliveryThreads; i++)
                                                                        {

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

            if (pDeliHandles[i] == INVALID_HANDLE_VALUE)

            throw new CWEBCLNT_ERR( ERR_DELIVERY_THREAD_FAILED );
                                                                        }
                                                }

                                                break;

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

            //write event into txn log for STOP

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


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

            CTxnLog       *txnDelilogLocal = txnDelilog;

            txnDelilog= NULL;

            delete txnDelilogLocal;
                                                                        }

                                                                        delete []
pDeliHandles;                                                           delete []
pDelBuff;


            CloseHandle( hWorkerSemaphore );

            CloseHandle( hDoneEvent );

            DeleteCriticalSection(&DelBuffCriticalSection);
                                                }


            DeleteCriticalSection(&TermCriticalSection);

                                                if (hLibInstanceTm !=
NULL)
```

```
            FreeLibrary( hLibInstanceTm );
                                                hLibInstanceTm = NULL;

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

                                                Sleep(500);
                                                break;

                                    default:
                                                /* nothing */;
                        }
            }
            catch (CBaseErr *e)
            {
                        WriteMessageToEventLog( e->ErrorText() );
                        delete e;
                        TerminateExtension(0);
                        return FALSE;
            }
            catch (...)
            {
                        WriteMessageToEventLog(TEXT("Unhandled exception.
DLL could not load."));
                        TerminateExtension(0);
                        return FALSE;
            }

            return TRUE;
}


/* FUNCTION: GetExtensionVersion
 *
 * PURPOSE:  This function is called by the inet service when the DLL is first
loaded.
 *
 * ARGUMENTS: HSE_VERSION_INFO          *pVer         passed in structure in
which to place expected version number.
 *
 * RETURNS:                TRUE         inet service expected return value.
 */

BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO *pVer)
{
            pVer->dwExtensionVersion = MAKELONG(HSE_VERSION_MINOR,
HSE_VERSION_MAJOR);
            lstrcpyn(pVer->lpszExtensionDesc, "TPC-C Server.",
HSE_MAX_EXT_DLL_NAME_LEN);

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

            return TRUE;
}

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

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

            TermDeleteAll();
            return TRUE;
}


/* FUNCTION: HttpExtensionProc
 *
 * PURPOSE:  This function is the main entry point for the TPCC DLL. The
internet service
 *                                    calls this function passing in the http
string.
 *
 * ARGUMENTS: EXTENSION_CONTROL_BLOCK    *pECB         structure pointer to
passed in internet
 *
            service information.
 *
 * RETURNS:                DWORD        HSE_STATUS_SUCCESS
                                    connection can be dropped if error
 *
            HSE_STATUS_SUCCESS_AND_KEEP_CONN        keep connect valid comment
sent
 *
 * COMMENTS:  None
 *
 */

DWORD WINAPI HttpExtensionProc(EXTENSION_CONTROL_BLOCK *pECB)
{
            int                                 iCmd, FormId, TermId,
iSyncId;
            char                        szBuffer[4096];

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


#ifdef ICECAP
#endif
            StartCAP();

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

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

                                    WriteMessageToEventLog( szTmp );

                                                throw new
CWEBCLNT_ERR( ERR_INVALID_TERMID );
                                    }

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

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

                        switch(iCmd)
                        {
                        case 0:
                                    WelcomeForm(pECB, szBuffer);
                                    break;
                        case 1:
                                    switch( FormId )
                                    {
                                                case WELCOME_FORM:
                                                case MAIN_MENU_FORM:
                                                            break;
                                                case NEW_ORDER_FORM:

                        ProcessNewOrderForm(pECB, TermId, szBuffer);
                                                            break;
                                                case PAYMENT_FORM:

                        ProcessPaymentForm(pECB, TermId, szBuffer);
                                                            break;
                                                case DELIVERY_FORM:

                        ProcessDeliveryForm(pECB, TermId, szBuffer);
                                                            break;
                                                case ORDER_STATUS_FORM:

                        ProcessOrderStatusForm(pECB, TermId, szBuffer);
                                                            break;
                                                case STOCK_LEVEL_FORM:

                        ProcessStockLevelForm(pECB, TermId, szBuffer);
                                                            break;
                                    }
                                    break;
                        case 2:
                                    // new-order selected from menu; display
new-order input form
                                    MakeNewOrderForm(TermId, NULL,
INPUT_FORM, szBuffer);
                                    break;
                        case 3:
                                    // payment selected from menu; display
payment input form
                                    MakePaymentForm(TermId, NULL, INPUT_FORM,
szBuffer);
                                    break;
                        case 4:
                                    // delivery selected from menu; display
delivery input form
                                    MakeDeliveryForm(TermId, NULL,
INPUT_FORM, szBuffer);
                                    break;
                        case 5:
                                    // order-status selected from menu;
display order-status input form
                                    MakeOrderStatusForm(TermId, NULL,
INPUT_FORM, szBuffer);
                                    break;
                        case 6:
                                    // stock-level selected from menu;
display stock-level input form
                                    MakeStockLevelForm(TermId, NULL,
INPUT_FORM, szBuffer);
                                    break;
                        case 7:
                                    // ExitCmd
                                    TermDelete(TermId);
                                    WelcomeForm(pECB, szBuffer);
                                    break;
                        case 8:
                                    SubmitCmd(pECB, szBuffer);
                                    break;
                        case 9:
                                    // menu
                                    MakeMainMenuForm(TermId,
Term.pClientData[TermId].iSyncId, szBuffer);
                                    break;
                        case 10:
                                    // CMD=Clear
                                    // resets all connections; should only
be used when no other connections are active
                                    TermDeleteAll();
                                    TermInit();
                                    WelcomeForm(pECB, szBuffer);
                                    break;
                        case 11:    // CMD=Stats
                                    StatsCmd(pECB, szBuffer);
                                    break;
                        }
            }
            catch (CBaseErr *e)
            {
                        ErrorForm( pECB, e->ErrorType(), e->ErrorNum(),
TermId, iSyncId, e->ErrorText(), szBuffer );
                        delete e;
            }
            catch (...)
            {
                        ErrorForm( pECB, ERR_TYPE_WEBDLL, 0, TermId, iSyncId,
"Error: Unhandled exception in Web Client.", szBuffer );
            }

#ifdef ICECAP
            StopCAP();
#endif


            lpbSize = strlen(szBuffer);
            wsprintf(szHeader1,
                                    "Content-Type: text/html\r\n"
                                    "Content-Length: %d\r\n"
                                    "Connection: Keep-Alive\r\n\r\n" ,
lpbSize);
            strcat( szHeader1, szBuffer );

            (*pECB->ServerSupportFunction)(pECB->ConnID,
HSE_REQ_SEND_RESPONSE_HEADER, szHeader, (LPDWORD) &dwSize, (LPDWORD)szHeader1);

            //finish up and keep connection
            pECB->dwHttpStatusCode = 200;
            return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}


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

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

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

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

        (VOID) DeregisterEventSource(hEventSource);
    }
}


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

/*static*/ void DeliveryWorkerThread(void *ptr)
{
            CTPCC_BASE                  *pTxn = NULL;

            DELIVERY_TRANSACTION                    delivery;
            PDELIVERY_DATA
            pDeliveryData;
```

```
            TXN_RECORD_TPCC_DELIV_DEF  txnDeliRec;

            DWORD
            index;
            HANDLE
            handles[2];

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

            assert(txnDelilog != NULL);

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

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

                                                ZeroMemory(&txnDeliRec,
sizeof(txnDeliRec));
                                                txnDeliRec.TxnType =
TXN_REC_TYPE_TPCC_DELIV_DEF;
                                                // make a local copy of
current entry from delivery buffer and increment buffer index
            EnterCriticalSection(&DelBuffCriticalSection);
                                                delivery =
*(pDelBuff+dwDelBuffBusyIndex);
                                                dwDelBuffFreeCount++;
                                                dwDelBuffBusyIndex++;
                                                if (dwDelBuffBusyIndex ==
dwDelBuffSize)            // wrap-around if at end of buffer
            dwDelBuffBusyIndex = 0;

            LeaveCriticalSection(&DelBuffCriticalSection);

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

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


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

            GetLocalTime( &trans_end );

                                                //log txn
                                                txnDeliRec.TxnStatus =
ERR_SUCCESS;

                                                for (int i=0; i<10; i++)

            txnDeliRec.o_id[i] = pDeliveryData->o_id[i];
                                                txnDeliRec.DeltaT4 =
(int)(Get64BitTime(&trans_end) - txnDeliRec.TxnStartTO);
                                                txnDeliRec.DeltaTxnExec =
(int)(Get64BitTime(&trans_end) - Get64BitTime(&trans_start));

                                                if (txnDelilog != NULL)
                                                            txnDelilog-
>WriteToLog(&txnDeliRec);
                                    }
                                    catch (CBaseErr *e)
                                    {
                                                char szTmp[1024];
                                                wsprintf( szTmp, "Error in Delivery Txn
thread.  %s", e->ErrorText() );
                                                WriteMessageToEventLog( szTmp );

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

                                                delete e;
                                    }
                                    catch (...)
                                    {
                                                // unhandled exception; shouldn't
happen; not much we can do...
                                                WriteMessageToEventLog(TEXT("Unhandled
exception caught in DeliveryWorkerThread."));
                                    }
            }

ErrorExit:
            delete pTxn;
            _endthread();
}

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

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

            EnterCriticalSection(&DelBuffCriticalSection);
            if (dwDelBuffFreeCount > 0)
            {
                        bError = FALSE;
                        (pDelBuff+dwDelBuffFreeIndex)->w_id
= w_id;
                        (pDelBuff+dwDelBuffFreeIndex)->o_carrier_id
= o_carrier_id;
                        GetLocalTime(&(pDelBuff+dwDelBuffFreeIndex)->queue);

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

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

            return bError;
}

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

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

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

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

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

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

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

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

/* FUNCTION: void WelcomeForm
 *
 */

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

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

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

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

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

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

            "</PRE></font>"

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

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

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

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

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

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

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

            sprintf( szTmp,                "Configuration Settings: <BR><font
face=\"Courier New\" color=\"blue\"><PRE>"                                  "Txn Monitor
= <B>%s</B><BR>"                                                            "Database
protocol      = <B>%s</B><BR>"                                              "Max
Connections        = <B>%d</B><BR>"                                         "# of
Delivery Threads  = <B>%d</B><BR>"                                          "Max Pending
Deliveries = <B>%d</B><BR>"
                                    , szTxnMonNames[Reg.eTxnMon],
szDBNames[Reg.eDB_Protocol],                Reg.dwMaxConnections,
dwNumDeliveryThreads, dwDelBuffSize );
            strcat( szBuffer, szTmp);

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

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

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

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

```
                "DB User ID   = <INPUT NAME=\"db_user\" SIZE=20 VALUE=\"%s\"><BR>"
                "DB Password  = <INPUT NAME=\"db_passwd\" SIZE=20
VALUE=\"%s\"><BR>"
                "DB Name      = <INPUT NAME=\"db_name\" SIZE=20 VALUE=\"%s\"><BR>"
                "</PRE></font>"
                                                , Reg.szDbServer,
Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
                else
                        // if using a txn monitor, connection options are
determined from registry; can't
                        // set per user.  show options fyi
                        sprintf( szTmp,         "Database options which
will be used by the transaction monitor:<BR>"
                "<font face=\"Courier New\" color=\"blue\"><PRE>"
                "DB Server              = <B>%s</B><BR>"
                "DB User ID             = <B>%s</B><BR>"
                "DB Password            = <B>%s</B><BR>"
                "DB Name                = <B>%s</B><BR>"
                "</PRE></font>"
                                                , Reg.szDbServer,
Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
                strcat( szBuffer, szTmp);

                sprintf( szTmp,         "Please enter your Warehouse and
District for this session:<BR>"
                                                "<font
face=\"Courier New\" color=\"blue\"><PRE>" );
                strcat( szBuffer, szTmp);
                strcat( szBuffer,       "Warehouse ID = <INPUT NAME=\"w_id\"
SIZE=4><BR>"
                "District ID  = <INPUT NAME=\"d_id\" SIZE=2><BR>"
                "</PRE></font><HR>"
                "<INPUT TYPE=\"Submit\" NAME=\"CMD\" VALUE=\"Submit\">"
                "</FORM></BODY></HTML>");
}

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

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

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

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

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

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

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


                iNewTerm = TermAdd();

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

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

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

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

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

                EnterCriticalSection(&TermCriticalSection);

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

                LeaveCriticalSection(&TermCriticalSection);

                wsprintf( szBuffer,
                                        "<HTML><HEAD><TITLE>TPC-C Web Client
Stats</TITLE></HEAD>"
                                        "<BODY><B><BIG> Total Active
Connections: %d </BIG></B><BR></BODY></HTML>"
                                        , iTotal );
}

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



                        },
                        {               ERR_D_ID_INVALID,
                                                "Invalid District ID Must
be 1 to 10."
                                                                },
                        {               ERR_DELIVERY_CARRIER_ID_RANGE,
                        "Delivery Carrier ID out of range must be 1 - 10."


                        },
                        {               ERR_DELIVERY_CARRIER_INVALID,
                        "Delivery Carrier ID invalid must be numeric 1 - 10."


                        },             {               ERR_DELIVERY_MISSING_OCD_KEY,
                        "Delivery missing Carrier ID key \"OCD*\"."



                                        },
                        {               ERR_DELIVERY_THREAD_FAILED,
                                        "Could not start delivery worker
thread."
                                                        },
                        {               ERR_GETPROCADDR_FAILED,
                                        "Could not map proc in DLL.  GetProcAddr
error.  DLL="
                        },
                        {               ERR_HTML_ILL_FORMED,
                                        "Required key field is missing from HTML
string."
                                        },
                        {               ERR_INVALID_SYNC_CONNECTION,
                        "Invalid Terminal Sync ID."


                                                        },
                        {               ERR_INVALID_TERMID,
                                                "Invalid Terminal ID."
                        },
                        {               ERR_LOADDLL_FAILED,
                                                "Load of DLL failed.
DLL="

                        },
                        {               ERR_MAX_CONNECTIONS_EXCEEDED,
                        "No connections available.  Max Connections is
probably too low."   },
                        {               ERR_MISSING_REGISTRY_ENTRIES,
                        "Required registry entries are missing.  Rerun
INSTALL to correct."
                },
                        {               ERR_NEWORDER_CUSTOMER_INVALID,
                        "New Order customer id invalid data type, range = 1
to 3000."
                },
                        {               ERR_NEWORDER_CUSTOMER_KEY,
                        "New Order missing Customer key \"CID*\"."


                                                },
                        {               ERR_NEWORDER_DISTRICT_INVALID,
                        "New Order District ID Invalid range 1 - 10."


                        },
                        {               ERR_NEWORDER_FORM_MISSING_DID,
                        "New Order missing District key \"DID*\"."


                                        },
                        {               ERR_NEWORDER_ITEMID_INVALID,
                        "New Order Item Id is wrong data type, must be
numeric."
                        },
                        {               ERR_NEWORDER_ITEMID_RANGE,
                        "New Order Item Id is out of range. Range = 1 to
999999."
                                        },
                        {               ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
                "New Order Item_Id field entered without a corresponding Supp_W."
},
                        {               ERR_NEWORDER_MISSING_IID_KEY,
                        "New Order missing Item Id key \"IID*\"."


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


                                        },
                        {               ERR_NEWORDER_MISSING_SUPPW_KEY,
                        "New Order missing Supp_W key \"SP##*\"."


                                        },
                        {               ERR_NEWORDER_NOITEMS_ENTERED,
                        "New Order No order lines entered."


                                                },
                        {               ERR_NEWORDER_QTY_INVALID,
                        "New Order Qty invalid must be numeric range 1 - 99."


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


                },
                        {               ERR_ORDERSTATUS_CID_AND_CLT,
                        "Order Status Only Customer ID or Last Name may be
entered, not both."
                },
                        {               ERR_ORDERSTATUS_CID_INVALID,
                        "Order Status Customer ID invalid, range must be
numeric 1 - 3000."
                },
                        {               ERR_ORDERSTATUS_CLT_RANGE,
                        "Order Status Customer last name longer than 16
characters."
                },
                        {               ERR_ORDERSTATUS_DID_INVALID,
                        "Order Status District invalid, value must be numeric
```

```
1 - 10."
},
                    {           ERR_ORDERSTATUS_MISSING_CID_CLT,
          "Order Status Either Customer ID or Last Name must be entered."

},
                    {           ERR_ORDERSTATUS_MISSING_CID_KEY,
          "Order Status missing Customer key \"CID*\"."


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

          },
                    {           ERR_ORDERSTATUS_MISSING_DID_KEY,
          "Order Status missing District key \"DID*\"."


                    },
                    {           ERR_PAYMENT_CDI_INVALID,
                    "Payment Customer district invalid must be numeric."


                    },
                    {           ERR_PAYMENT_CID_AND_CLT,
                   "Payment Only Customer ID or Last Name may be entered,
not
both."      },
                    {           ERR_PAYMENT_CUSTOMER_INVALID,
                    "Payment Customer data type invalid, must be
numeric."
          },
                    {           ERR_PAYMENT_CWI_INVALID,
                    "Payment Customer Warehouse invalid, must be
numeric."

                    },
                    {           ERR_PAYMENT_DISTRICT_INVALID,
                    "Payment District ID is invalid, must be 1 - 10."


                    },
                    {           ERR_PAYMENT_HAM_INVALID,
                    "Payment Amount invalid data type must be numeric."


                              },
                    {           ERR_PAYMENT_HAM_RANGE,
                              "Payment Amount out of range, 0 -
9999.99."

                                        },
                    {           ERR_PAYMENT_LAST_NAME_TO_LONG,
                    "Payment Customer last name longer than 16
characters."
          },
                    {           ERR_PAYMENT_MISSING_CDI_KEY,
                    "Payment missing Customer district key \"CDI*\"."


                    },
                    {           ERR_PAYMENT_MISSING_CID_CLT,
                    "Payment Either Customer ID or Last Name must be
entered."
          },
                    {           ERR_PAYMENT_MISSING_CID_KEY,
                    "Payment missing Customer Key \"CID*\"."


                              },
                    {           ERR_PAYMENT_MISSING_CLT_KEY,
                    "Payment missing Customer Last Name key \"CLT*\"."


                    },
                    {           ERR_PAYMENT_MISSING_CWI_KEY,
                    "Payment missing Customer Warehouse key \"CWI*\"."


                    },
                    {           ERR_PAYMENT_MISSING_DID_KEY,
                    "Payment missing District Key \"DID*\"."


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


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


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


},
                    {           ERR_STOCKLEVEL_THRESHOLD_RANGE,
                    "Stock Level Threshold out of range, range must be 1
- 99."
          },
                    {           ERR_VERSION_MISMATCH,
                              "Invalid version field.  RTE and Web
Client are probably out of

sync." },
                    {           ERR_W_ID_INVALID,
                                        "Invalid Warehouse ID."




                    },
                    {           0,


                    ""



          }
          };

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

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

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

}

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

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

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

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

          *pQueryString = ptr;
          return;

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

/* FUNCTION: GetIntKeyValue
 *
 * PURPOSE:  This function parses a http formatted string for a specific key
value.
 *
 * ARGUMENTS: char                                   *pQueryString http string
from client browser
 *                                                   char
           *pKey                                   key value to look for
 *                                                   WEBERROR
           NoKeyErr                  error value to throw if key not found
 *                                                   WEBERROR
           NotIntErr                 error value to throw if value not
numeric
 *
 * RETURNS:               integer
 *
 * ERROR:                 if (the pKey value is not found) then
 *                                                             if
(NoKeyErr != NO_ERR)
 *
           throw CWEBCLNT_ERR(err)
 *                                                           else
 *
           return 0
 *                                                   else if (non-numeric char
found) then
 *                                                             if
(NotIntErr != NO_ERR) then
 *
           throw CWEBCLNT_ERR(err)
 *                                                           else
 *
           return 0
 *
 * COMMENTS:  http keys are formatted either KEY=value& or KEY=value\0. This DLL
formats
 *                                                   TPC-C input fields in such
a manner that the keys can be extracted in the
 *                                                   above manner.
 */

int GetIntKeyValue(char **pQueryString, char *pKey, WEBERROR NoKeyErr, WEBERROR
NotIntErr)
{
          char *ptr0;
          char *ptr;

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

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

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

          *pQueryString = ptr;
          return atoi(ptr0);

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

/* FUNCTION: TermInit
 *
 * PURPOSE:  This function initializes the client terminal structure; it is
called when the TPCC.DLL
 *                                   is first loaded by the inet service.
 *
 */

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

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

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

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

          Term.iFreeList                    = Term.iNumEntries-1;
          // build free list
          // note: Term.pClientData[0].iNextFree gets set to -1, which marks
it as "in use".
```

```
		//	This is intentional, as the zero entry is used as an
anchor and never
		//				allocated as an actual terminal.
		for(int i=0; i<Term.iNumEntries; i++)
			Term.pClientData[i].iNextFree = i-1;

		LeaveCriticalSection(&TermCriticalSection);
}

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

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

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

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

		LeaveCriticalSection(&TermCriticalSection);
}

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

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

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

		EnterCriticalSection(&TermCriticalSection);
		if (Term.iFreeList != 0)
		{		// position is available
			iNewTerm = Term.iFreeList;
			Term.iFreeList =
Term.pClientData[iNewTerm].iNextFree;
			Term.pClientData[iNewTerm].iNextFree = -1;  //
indicates this position is in use
		}
		else
		{		// no open slots, so find the slot that hasn't been
used in the longest time and reuse it
			for(iNewTerm=1, i=1, iTickCount=0x7FFFFFFF;
i<Reg.dwMaxConnections; i++)
			{
				if (iTickCount >
Term.pClientData[i].iTickCount)
				{
					iTickCount =
Term.pClientData[i].iTickCount;
					iNewTerm = i;
				}
			}
			// if oldest term is less than one minute old, it
probably means that more connections
			// are being attempted than were specified as "Max
Connections" at install.  In this case,
			// do not bump existing connection; instead, return
error to requestor.
			if ((GetTickCount() - iTickCount) < 60000)
			{

		LeaveCriticalSection(&TermCriticalSection);
				throw new
CWEBCLNT_ERR( ERR_MAX_CONNECTIONS_EXCEEDED );
			}
		}

		Term.pClientData[iNewTerm].iTickCount = GetTickCount();
		Term.pClientData[iNewTerm].iSyncId = Term.iMasterSyncId++;
		Term.pClientData[iNewTerm].pTxn = NULL;

		LeaveCriticalSection(&TermCriticalSection);
		return iNewTerm;
}

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

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

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

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

			LeaveCriticalSection(&TermCriticalSection);
		}
}


/* FUNCTION: MakeErrorForm
 */

void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int iErrorNum, int
iTermId, int iSyncId, char *szErrorText, char *szBuffer )
{
		wsprintf(szBuffer,
				"<HTML><HEAD><TITLE>TPC-C Error</TITLE></HEAD><BODY>"
				"<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
				"<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"%d\">"
				"<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"%d\">"
				"<INPUT TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"
				"<INPUT TYPE=\"hidden\" NAME=\"TERMID\"
VALUE=\"%d\">"
				"<INPUT TYPE=\"hidden\" NAME=\"SYNCID\"
VALUE=\"%d\">"
				"<BOLD>An Error Occurred</BOLD><BR><BR>"
				"%s"
				"<BR><BR><HR>"
				"<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
				"<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"
				"<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"
				"<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-
Status..\">"
				"<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-
Level..\">"
				"<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Exit..\">"
				"</FORM></BODY></HTML>"
				, iType, iErrorNum, MAIN_MENU_FORM, iTermId, iSyncId,
szErrorText );
}

/* FUNCTION: MakeMainMenuForm
 */

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

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

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

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

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

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

void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA *pNewOrderData, BOOL bInput,
char *szForm)
{
		int			i, c;
		BOOL		bValid;
		static		char szBR[] = " <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>";

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

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

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

		if ( bInput )
		{
			c += wsprintf(szForm+c, "Warehouse: %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=\"SPOO*\" SIZE=4>  <INPUT
```

```
NAME=\"IID00*\" SIZE=6>                         <INPUT NAME=\"Qty00*\"
SIZE=1><BR>"
                        "  <INPUT NAME=\"SP01*\" SIZE=4>  <INPUT
NAME=\"IID01*\" SIZE=6>                         <INPUT NAME=\"Qty01*\"
SIZE=1><BR>"
                        "  <INPUT NAME=\"SP02*\" SIZE=4>  <INPUT
NAME=\"IID02*\" SIZE=6>                         <INPUT NAME=\"Qty02*\"
SIZE=1><BR>"
                        "  <INPUT NAME=\"SP03*\" SIZE=4>  <INPUT
NAME=\"IID03*\" SIZE=6>                         <INPUT NAME=\"Qty03*\"
SIZE=1><BR>"
                        "  <INPUT NAME=\"SP04*\" SIZE=4>  <INPUT
NAME=\"IID04*\" SIZE=6>                         <INPUT NAME=\"Qty04*\"
SIZE=1><BR>"
                        "  <INPUT NAME=\"SP05*\" SIZE=4>  <INPUT
NAME=\"IID05*\" SIZE=6>                         <INPUT NAME=\"Qty05*\"
SIZE=1><BR>"
                        "  <INPUT NAME=\"SP06*\" SIZE=4>  <INPUT
NAME=\"IID06*\" SIZE=6>                         <INPUT NAME=\"Qty06*\"
SIZE=1><BR>"
                        "  <INPUT NAME=\"SP07*\" SIZE=4>  <INPUT
NAME=\"IID07*\" SIZE=6>                         <INPUT NAME=\"Qty07*\"
SIZE=1><BR>"
                        "  <INPUT NAME=\"SP08*\" SIZE=4>  <INPUT
NAME=\"IID08*\" SIZE=6>                         <INPUT NAME=\"Qty08*\"
SIZE=1><BR>"
                        "  <INPUT NAME=\"SP09*\" SIZE=4>  <INPUT
NAME=\"IID09*\" SIZE=6>                         <INPUT NAME=\"Qty09*\"
SIZE=1><BR>"
                        "  <INPUT NAME=\"SP10*\" SIZE=4>  <INPUT
NAME=\"IID10*\" SIZE=6>                         <INPUT NAME=\"Qty10*\"
SIZE=1><BR>"
                        "  <INPUT NAME=\"SP11*\" SIZE=4>  <INPUT
NAME=\"IID11*\" SIZE=6>                         <INPUT NAME=\"Qty11*\"
SIZE=1><BR>"
                        "  <INPUT NAME=\"SP12*\" SIZE=4>  <INPUT
NAME=\"IID12*\" SIZE=6>                         <INPUT NAME=\"Qty12*\"
SIZE=1><BR>"
                        "  <INPUT NAME=\"SP13*\" SIZE=4>  <INPUT
NAME=\"IID13*\" SIZE=6>                         <INPUT NAME=\"Qty13*\"
SIZE=1><BR>"
                        "  <INPUT NAME=\"SP14*\" SIZE=4>  <INPUT
NAME=\"IID14*\" SIZE=6>                         <INPUT NAME=\"Qty14*\"
SIZE=1><BR>"

                        "Execution Status:
Total:<BR>"

                        "</font></PRE><HR>"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Process\">"

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

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

            }
            else
            {
                    c += wsprintf(szForm+c, "Warehouse: %4.4d
District: %2.2d                 Date: ",
                                    pNewOrderData->w_id,
                                    pNewOrderData->d_id);

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

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

                    if ( bValid )
                    {
                            c += sprintf(szForm+c,
            "%%Disc: %5.2f          <BR>"

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

            " Supp_W  Item_Id  Item Name                 Qty  Stock  B/G
Price    Amount<BR>",
                                    100.0*pNewOrderData-
>c_discount,
                                    pNewOrderData->o_id,
                                    pNewOrderData->o_ol_cnt,
                                    100.0 * pNewOrderData-
>w_tax,
                                    100.0 * pNewOrderData-
>d_tax);

                            for(i=0; i<pNewOrderData->o_ol_cnt; i++)
                            {
                                    c += sprintf(szForm+c,
"  %4.4d   %6.6d   %-24s  %2.2d    %3.3d    %1.1s   $%6.2f  $%7.2f  <BR>",
                                    pNewOrderData->OL[i].ol_supply_w_id,
                                    pNewOrderData->OL[i].ol_i_id,
                                    pNewOrderData->OL[i].ol_i_name,
                                    pNewOrderData->OL[i].ol_quantity,
                                    pNewOrderData->OL[i].ol_stock,
                                    pNewOrderData->OL[i].ol_brand_generic,
                                    pNewOrderData->OL[i].ol_i_price,
                                    pNewOrderData->OL[i].ol_amount );
                            }
                    }
                    else
                    {
                            c += wsprintf(szForm+c,
                                            "%Disc:<BR>"
                                            "Order Number: %8.8d
Number of Lines:              W_tax:          D_tax:<BR> <BR>"
                                            " Supp_W  Item_Id  Item
Name                 Qty  Stock  B/G  Price    Amount<BR>"
                                            , pNewOrderData->o_id);

                            i = 0;
                    }

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

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

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

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

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

            c = wsprintf(szForm,
                                    "<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD><BODY>"
                                    "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
                                    "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">"
                                    "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
                                    "<INPUT TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"
                                    "<INPUT TYPE=\"hidden\" NAME=\"TERMID\"
VALUE=\"%d\">"
                                    "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\"
VALUE=\"%d\">"
                                    "<PRE><font face=\"Courier\">
Payment<BR>"
                                    "Date: "
                                    , PAYMENT_FORM, iTermId,
Term.pClientData[iTermId].iSyncId);

            if ( !bInput )
            {
                    c += wsprintf(szForm+c, "%2.2d-%2.2d-
%4.4d %2.2d:%2.2d:%2.2d",
                                    pPaymentData->h_date.day,
                                    pPaymentData->h_date.month,
                                    pPaymentData->h_date.year,
                                    pPaymentData->h_date.hour,
                                    pPaymentData->h_date.minute,
                                    pPaymentData->h_date.second);
            }

            if ( bInput )
            {
                    c += wsprintf(szForm+c,
                                    "<BR> <BR>Warehouse: %4.4d"
                                    "                   District:
<INPUT NAME=\"DID*\" SIZE=1><BR> <BR> <BR> <BR>"
                                    "Customer: <INPUT NAME=\"CID*\" SIZE=4>"
                                    "Cust-Warehouse: <INPUT NAME=\"CWI*\"
SIZE=4>  "
                                    "Cust-District: <INPUT NAME=\"CDI*\"
SIZE=1><BR>"
                                    "Name:                          <INPUT
NAME=\"CLT*\" SIZE=16>                          Since:<BR>"
                                    "
Credit:<BR>"
                                    "
Disc:<BR>"
                                    "
Phone:<BR> <BR>"
                                    "Amount Paid:           $<INPUT
NAME=\"HAM*\" SIZE=7>       New Cust-Balance:<BR>"
                                    "Credit Limit:<BR> <BR>Cust-Data: <BR>
<BR> <BR> <BR> <BR></font></PRE><HR>"
                                    "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Process\"><INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
                                    "</BODY></FORM></HTML>"
                                    , Term.pClientData[iTermId].w_id);
            }
            else
            {
                    c += wsprintf(szForm+c,
                                    "<BR> <BR>Warehouse: %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->w_street_1,
pPaymentData->d_street_1
                                    , pPaymentData->w_street_2,
pPaymentData->d_street_2
                                    , pPaymentData->w_city, pPaymentData-
>w_state, pPaymentData->w_zip, pPaymentData->w_zip+5
                                    , pPaymentData->d_city, pPaymentData-
>d_state, pPaymentData->d_zip, pPaymentData->d_zip+5
                                    , pPaymentData->c_id, pPaymentData-
>c_w_id,      pPaymentData->c_d_id
                                    , pPaymentData->c_first, pPaymentData-
>c_middle, pPaymentData->c_last
                                    , pPaymentData->c_since.day,
pPaymentData->c_since.month,            pPaymentData->c_since.year
                                    , pPaymentData->c_street_1,
pPaymentData->c_credit
                                    );

                    c += sprintf(szForm+c,
                                    "       %-
20s                     %%Disc:  %5.2f<BR>",
                                    pPaymentData->c_street_2,
100.0*pPaymentData->c_discount);

                    c += wsprintf(szForm+c,
                                    "       %-20s %-2s %5.5s-%4.4s
Phone:  %6.6s-%3.3s-%3.3s-%4.4s<BR> <BR>",
                                    pPaymentData->c_city, pPaymentData-
>c_state, pPaymentData->c_zip, pPaymentData->c_zip+5,
                                    pPaymentData->c_phone, pPaymentData-
>c_phone+6, pPaymentData->c_phone+9, pPaymentData->c_phone+12 );

                    c += sprintf(szForm+c,
                                    "Amount Paid:           $%7.2f      New
Cust-Balance: $%14.2f<BR>"
                                    "Credit Limit:   $%13.2f<BR> <BR>"
                                    , pPaymentData->h_amount, pPaymentData-
>c_balance
                                    , pPaymentData->c_credit_lim
                                    );

                    if ( pPaymentData->c_credit[0] == 'B' &&
pPaymentData->c_credit[1] == 'C' )
                            c += wsprintf(szForm+c,
                                    "Cust-
Data: %-50.50s<BR>          %-50.50s<BR>            %-50.50s<BR>          %-
50.50s<BR>",
                                    pPaymentData-
>c_data, pPaymentData->c_data+50, pPaymentData->c_data+100, pPaymentData-
>c_data+150 );
                    else
                            strcpy(szForm+c, "Cust-Data: <BR> <BR>
<BR> <BR>");

                    strcat(szForm,          " <BR></font></PRE><HR>"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Payment..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Delivery..\">"
```

```
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Exit..\">"
                "</BODY></FORM></HTML>");
                }
}

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

void MakeOrderStatusForm(int iTermId, ORDER_STATUS_DATA *pOrderStatusData, BOOL
bInput, char *szForm)
{
                int                     i, c;
                static char szBR[] = "  <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR>";

                c = wsprintf(szForm,
                                "<HTML><HEAD><TITLE>TPC-C Order-
Status</TITLE></HEAD><BODY>"
                                "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"TERMID\"
VALUE=\"%d\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\"
VALUE=\"%d\">"
                                "<PRE><font face=\"Courier\">
Order-Status<BR>"
                                "Warehouse: %4.4d   ",
                                ORDER_STATUS_FORM, iTermId,
Term.pClientData[iTermId].iSyncId, Term.pClientData[iTermId].w_id);

                if ( bInput )
                {
                                strcpy(szForm+c,
                                                "District: <INPUT NAME=\"DID*\"
SIZE=1><BR>"
                                                "Customer: <INPUT NAME=\"CID*\" SIZE=4>
Name:                           <INPUT NAME=\"CLT*\" SIZE=23><BR>"
                                                "Cust-Balance:<BR> <BR>"
                                                "Order-Number:              Entry-Date:
Carrier-Number:<BR>"
                                                "Supply-W      Item-Id    Qty      Amount
Delivery-Date<BR> <BR> <BR> <BR> <BR>"
                                                " <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR></font></PRE>"
                                                "<HR><INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Process\"><INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
                                                "</BODY></FORM></HTML>" );
                }
                else
                {
                                c += wsprintf(szForm+c,
                                                "District: %2.2d<BR>"
                                                "Customer: %4.4d   Name: %-16s %-2s %-
16s<BR>",
                                                pOrderStatusData->d_id,
pOrderStatusData->c_id,
                                                pOrderStatusData->c_first,
pOrderStatusData->c_middle, pOrderStatusData->c_last);

                                c += sprintf(szForm+c, "Cust-Balance: $%9.2f<BR>
<BR>",
                                                pOrderStatusData->c_balance);

                                c += wsprintf(szForm+c,
                                                "Order-Number: %8.8d   Entry-
Date: %2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d   Carrier-Number: %2.2d<BR>"
                                                "Supply-W      Item-Id    Qty      Amount
Delivery-Date<BR>",
                                                pOrderStatusData->o_id,
                                                pOrderStatusData->o_entry_d.day,
                                                pOrderStatusData->o_entry_d.month,
                                                pOrderStatusData->o_entry_d.year,
                                                pOrderStatusData->o_entry_d.hour,
                                                pOrderStatusData->o_entry_d.minute,
                                                pOrderStatusData->o_entry_d.second,
                                                pOrderStatusData->o_carrier_id);

                                for(i=0; i< pOrderStatusData->o_ol_cnt; i++)
                                {
                                                c += sprintf(szForm+c,
"  %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);
                                }
                                strncpy( szForm+c, szBR, (15-i)*5 );
                                c += (15-i)*5;

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

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

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

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

                if ( bInput )
                {
                                strcpy( szForm+c,
                                                "Carrier Number: <INPUT NAME=\"OCD*\"
SIZE=1><BR> <BR>"
                                                "Execution Status: <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR>"
                                                " <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> </font></PRE><HR>"
                                                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Process\">"
                                                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Menu\">"
                                                "</BODY></FORM></HTML>" );
                }
                else
                {
                                wsprintf( szForm+c,
                                                "Carrier Number: %2.2d<BR> <BR>"
                                                "Execution Status: %s <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR>"
                                                " <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> </font></PRE>"
                                                "<HR><INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
                                                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"
                                                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"
                                                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Order-Status..\">"
                                                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Stock-Level..\">"
                                                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Exit..\">"
                                                "</BODY></FORM></HTML>"

                                                , pDeliveryData->o_carrier_id,
                                                (pDeliveryData->exec_status_code ==
eOK) ? "Delivery has been queued." : "Delivery Post Failed      "
                                                );
                }
}

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

                PDELIVERY_DATA              pDelivery;

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

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

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

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

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

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

		PSTOCK_LEVEL_DATA	pStockLevel;

		pStockLevel = Term.pClientData[iTermId].pTxn-
>BuffAddr_StockLevel();
		ZeroMemory( pStockLevel, sizeof(STOCK_LEVEL_DATA) );

		pStockLevel->w_id = Term.pClientData[iTermId].w_id;
		pStockLevel->d_id = Term.pClientData[iTermId].d_id;

		pStockLevel->threshold = GetIntKeyValue(&ptr, "TT*",
ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY, ERR_STOCKLEVEL_THRESHOLD_INVALID);
		if ( pStockLevel->threshold >= 100 || pStockLevel->threshold < 0 )
			throw new
CWEBCLNT_ERR( ERR_STOCKLEVEL_THRESHOLD_RANGE );

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

		pStockLevel = Term.pClientData[iTermId].pTxn-
>BuffAddr_StockLevel();
		MakeStockLevelForm(iTermId, pStockLevel, OUTPUT_FORM, szBuffer);
}

/* FUNCTION: GetNewOrderData
 *
 * PURPOSE:  This function extracts and validates the new order form data from
an http command string.
 *
 * ARGUMENTS: LPSTR                                  lpszQueryString
 *           client browser http command string
 *                                                   NEW_ORDER_DATA
 *           *pNewOrderData                          pointer to new order data
structure
 *
 */

void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA *pNewOrderData)
{
		char		szTmp[26];
		int			i;
		short		items;
		int			ol_i_id, ol_quantity;
		char		*ptr = lpszQueryString;

		static char szSP[MAX_OL_NEW_ORDER_ITEMS][6] =
			{ "SP00*", "SP01*", "SP02*", "SP03*", "SP04*",
			  "SP05*", "SP06*", "SP07*", "SP08*", "SP09*",
			  "SP10*", "SP11*", "SP12*", "SP13*", "SP14*" };
		static char szIID[MAX_OL_NEW_ORDER_ITEMS][7] =
			{ "IID00*", "IID01*", "IID02*", "IID03*", "IID04*",
			  "IID05*", "IID06*", "IID07*", "IID08*", "IID09*",
			  "IID10*", "IID11*", "IID12*", "IID13*", "IID14*" };
		static char szQty[MAX_OL_NEW_ORDER_ITEMS][7] =
			{ "Qty00*", "Qty01*", "Qty02*", "Qty03*", "Qty04*",
			  "Qty05*", "Qty06*", "Qty07*", "Qty08*", "Qty09*",
			  "Qty10*", "Qty11*", "Qty12*", "Qty13*", "Qty14*" };

		pNewOrderData->d_id = GetIntKeyValue(&ptr, "DID*",
ERR_NEWORDER_FORM_MISSING_DID, ERR_NEWORDER_DISTRICT_INVALID);
		pNewOrderData->c_id = GetIntKeyValue(&ptr, "CID*",
ERR_NEWORDER_CUSTOMER_KEY, ERR_NEWORDER_CUSTOMER_INVALID);

		for(i=0, items=0; i<MAX_OL_NEW_ORDER_ITEMS; i++)
		{
			GetKeyValue(&ptr, szSP[i], szTmp, sizeof(szTmp),
ERR_NEWORDER_MISSING_SUPPW_KEY);
			if ( szTmp[0] )
			{
				if ( !IsNumeric(szTmp) )
					throw new
CWEBCLNT_ERR( ERR_NEWORDER_SUPPW_INVALID );
				pNewOrderData->OL[items].ol_supply_w_id
= (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 *pPaymentData
 *           pointer to payment data structure
 */

void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA *pPaymentData)
{
		char		szTmp[26];
		char		*ptr = lpszQueryString;
		BOOL		bCustIdBlank;

		pPaymentData->d_id = GetIntKeyValue(&ptr, "DID*",
ERR_PAYMENT_MISSING_DID_KEY, ERR_PAYMENT_DISTRICT_INVALID);

		GetKeyValue(&ptr, "CID*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_CID_KEY);
		if ( szTmp[0] == 0 )
		{
			bCustIdBlank = TRUE;
			pPaymentData->c_id = 0;
		}
		else
		{	// parse customer id and verify that last name was
NOT entered
			bCustIdBlank = FALSE;
			if ( !IsNumeric(szTmp) )
				throw new
CWEBCLNT_ERR( ERR_PAYMENT_CUSTOMER_INVALID );
			pPaymentData->c_id = atoi(szTmp);
		}

		pPaymentData->c_w_id = GetIntKeyValue(&ptr, "CWI*",
ERR_PAYMENT_MISSING_CWI_KEY, ERR_PAYMENT_CWI_INVALID);
		pPaymentData->c_d_id = GetIntKeyValue(&ptr, "CDI*",
ERR_PAYMENT_MISSING_CDI_KEY, ERR_PAYMENT_CDI_INVALID);

		if ( bCustIdBlank )
		{	// customer id is blank, so last name must be entered
			GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_CLT_KEY);
			if ( szTmp[0] == 0 )
				throw new
CWEBCLNT_ERR( ERR_PAYMENT_MISSING_CID_CLT );

			_strupr( szTmp );
			if ( strlen(pPaymentData->c_last) > LAST_NAME_LEN )
				throw new
CWEBCLNT_ERR( ERR_PAYMENT_LAST_NAME_TO_LONG );
			strcpy(pPaymentData->c_last, szTmp);
		}
		else
		{	// parse customer id and verify that last name was
NOT entered
			GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_CLT_KEY);
			if ( szTmp[0] != 0 )
				throw new
CWEBCLNT_ERR( ERR_PAYMENT_CID_AND_CLT );
		}

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

/* FUNCTION: GetOrderStatusData
 *
 * PURPOSE:  This function extracts and validates the payment form data from an
http command string.
 *
 */
void GetOrderStatusData(LPSTR lpszQueryString, ORDER_STATUS_DATA
*pOrderStatusData)
{
		char		szTmp[26];
		char		*ptr = lpszQueryString;

		pOrderStatusData->d_id = GetIntKeyValue(&ptr, "DID*",
ERR_ORDERSTATUS_MISSING_DID_KEY, ERR_ORDERSTATUS_DID_INVALID);

		GetKeyValue(&ptr, "CID*", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CID_KEY);
		if ( szTmp[0] == 0 )
		{	// customer id is blank, so last name must be entered
			pOrderStatusData->c_id = 0;
			GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CLT_KEY);
			if ( szTmp[0] == 0 )
				throw new
CWEBCLNT_ERR( ERR_ORDERSTATUS_MISSING_CID_CLT );

			_strupr( szTmp );
			if ( strlen(pOrderStatusData->c_last) >
LAST_NAME_LEN )
				throw new
CWEBCLNT_ERR( ERR_ORDERSTATUS_CLT_RANGE );
			strcpy(pOrderStatusData->c_last, szTmp);
		}
		else
		{	// parse customer id and verify that last name was
NOT entered
			if ( !IsNumeric(szTmp) )
				throw new
CWEBCLNT_ERR( ERR_ORDERSTATUS_CID_INVALID );
			pOrderStatusData->c_id = atoi(szTmp);
			GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CLT_KEY);
			if ( szTmp[0] != 0 )
				throw new
CWEBCLNT_ERR( ERR_ORDERSTATUS_CID_AND_CLT );
		}

}

/* FUNCTION: BOOL IsNumeric(char *ptr)
 *
 * PURPOSE:  This function determines if a string is numeric. It fails if any
characters other
 *           than numeric and null terminator are
present.
 *
 * ARGUMENTS: char                                   *ptr          pointer to
string to check.
 *
 * RETURNS:  BOOL          FALSE         if string is not all
numeric
 *           TRUE          if string contains only numeric characters i.e. '0' -
'9'
 */

BOOL IsNumeric(char *ptr)
{
		if ( *ptr == 0 )
			return FALSE;

		while( *ptr && isdigit(*ptr) )
			ptr++;
		return ( !*ptr );
}

/* FUNCTION: BOOL IsDecimal(char *ptr)
 *
 * PURPOSE:  This function determines if a string is a non-negative decimal
value.
 *          It fails if any characters other than a series of numbers followed
by
 *           a decimal point, another series of
numbers, and a null terminator are present.
 *
 * ARGUMENTS: char                                   *ptr          pointer to
string to check.
 *
 * RETURNS:  BOOL          FALSE         if string is not a valid
non-negative decimal value
 *           TRUE          if string is OK
 */

BOOL IsDecimal(char *ptr)
{
		char *dotptr;
```

```
        BOOL  bValid;

        if ( *ptr == 0 )
                return FALSE;

        // find decimal point
        dotptr = strchr( ptr, '.' );
        if (dotptr == NULL)
                // no decimal point, so just check for numeric
                return IsNumeric(ptr);
        *dotptr = 0;  // temporarily replace decimal with a terminator

        if ( *ptr != 0 )
                bValid = IsNumeric(ptr);
        // string starts with decimal point
        else if (*(dotptr+1) == 0)
                return FALSE;  // nothing but a decimal point is bad
        else
                bValid = TRUE;

        if (*(dotptr+1) != 0)
                // check text after decimal point
                bValid &= IsNumeric(dotptr+1);

        *dotptr = '.';  // replace decimal point
        return bValid;
}
```

isapi_dll/src/tpcc.def

```
LIBRARY TPCC.DLL

EXPORTS

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

isapi_dll/src/tpcc.h

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

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

#define TP_MAX_RETRIES
                50


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

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


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

        CTPCC_BASE    *pTxn;

} CLIENTDATA, *PCLIENTDATA;


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

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


        enum WEBERROR
        {
                NO_ERR,
                ERR_COMMAND_UNDEFINED,
                ERR_D_ID_INVALID,
                ERR_DELIVERY_CARRIER_ID_RANGE,
                ERR_DELIVERY_CARRIER_INVALID,
                ERR_DELIVERY_MISSING_OCD_KEY,
                ERR_DELIVERY_THREAD_FAILED,
                ERR_GETPROCADDR_FAILED,
                ERR_HTML_ILL_FORMED,
                ERR_INVALID_SYNC_CONNECTION,
                ERR_INVALID_TERMID,
                ERR_LOADDLL_FAILED,
                ERR_MAX_CONNECTIONS_EXCEEDED,
                ERR_MEM_ALLOC_FAILED,
                ERR_MISSING_REGISTRY_ENTRIES,
                ERR_NEWORDER_CUSTOMER_INVALID,
                ERR_NEWORDER_CUSTOMER_KEY,
                ERR_NEWORDER_DISTRICT_INVALID,
                ERR_NEWORDER_FORM_MISSING_DID,
                ERR_NEWORDER_ITEMID_INVALID,
                ERR_NEWORDER_ITEMID_RANGE,
                ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
                ERR_NEWORDER_MISSING_IID_KEY,
                ERR_NEWORDER_MISSING_QTY_KEY,
                ERR_NEWORDER_MISSING_SUPPW_KEY,
                ERR_NEWORDER_NOITEMS_ENTERED,
                ERR_NEWORDER_QTY_INVALID,
                ERR_NEWORDER_QTY_RANGE,
                ERR_NEWORDER_QTY_WITHOUT_SUPPW,
                ERR_NEWORDER_SUPPW_INVALID,
                ERR_NO_SERVER_SPECIFIED,
                ERR_ORDERSTATUS_CID_AND_CLT,
                ERR_ORDERSTATUS_CID_INVALID,
                ERR_ORDERSTATUS_CLT_RANGE,
                ERR_ORDERSTATUS_DID_INVALID,
                ERR_ORDERSTATUS_MISSING_CID_CLT,
                ERR_ORDERSTATUS_MISSING_CID_KEY,
                ERR_ORDERSTATUS_MISSING_CLT_KEY,
                ERR_ORDERSTATUS_MISSING_DID_KEY,
                ERR_PAYMENT_CDI_INVALID,
                ERR_PAYMENT_CID_AND_CLT,
                ERR_PAYMENT_CUSTOMER_INVALID,
                ERR_PAYMENT_CWI_INVALID,
                ERR_PAYMENT_DISTRICT_INVALID,
                ERR_PAYMENT_HAM_INVALID,
                ERR_PAYMENT_HAM_RANGE,
                ERR_PAYMENT_LAST_NAME_TO_LONG,
                ERR_PAYMENT_MISSING_CDI_KEY,
                ERR_PAYMENT_MISSING_CID_CLT,
                ERR_PAYMENT_MISSING_CID_KEY,
                ERR_PAYMENT_MISSING_CLT,
                ERR_PAYMENT_MISSING_CLT_KEY,
                ERR_PAYMENT_MISSING_CWI_KEY,
                ERR_PAYMENT_MISSING_DID_KEY,
                ERR_PAYMENT_MISSING_HAM_KEY,
                ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
                ERR_STOCKLEVEL_THRESHOLD_INVALID,
                ERR_STOCKLEVEL_THRESHOLD_RANGE,
                ERR_VERSION_MISMATCH,
                ERR_W_ID_INVALID
        };

class CWEBCLNT_ERR : public CBaseErr
{
        public:

                CWEBCLNT_ERR(WEBERROR Err)
                {
                        m_Error = Err;
                        m_szTextDetail = NULL;
                        m_SystemErr = 0;
                        m_szErrorText = NULL;
                };

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

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

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

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

};

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

//function prototypes

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

isapi_dll/src/tpcc.rc

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

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

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

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

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
```

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

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

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

#endif    // !_MAC


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

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

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

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

#endif    // APSTUDIO_INVOKED


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

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


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

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
    IDD_DIALOG1, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 179
        TOPMARGIN, 7
        BOTTOMMARGIN, 88
    END
END
#endif    // APSTUDIO_INVOKED

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

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


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

tm_com_dll/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 **

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

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

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

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

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

odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib

odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
/out:".\bin/tpcc_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

odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386
/pdbtype:sept
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib

odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386
/out:".\bin/tpcc_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
```

tm_com_dll/src/tpcc_com.cpp

```
/*            FILE:                              TPCC_COM.CPP
 *                                                                  Microsoft TPC-C Kit Ver.
4.20.000
 *                                                                  Copyright Microsoft, 1999
 *               All Rights Reserved
 *
 *                                                                  not yet audited
 *
 *            PURPOSE:      Source file for TPC-C COM+ class implementation.
 *            Contact:      Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                          4.20.000 - first version
 */

// needed for CoinitializeEx
#define _WIN32_WINNT 0x0400

#include <windows.h>

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

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

#include "..\..\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "..\..\tpcc_com_all\src\tpcc_com_all_i.c"

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_COM* CTPCC_COM_new(BOOL bSinglePool)
{
            return new CTPCC_COM(bSinglePool);
}

CTPCC_COM::CTPCC_COM(BOOL bSinglePool)
{
            HRESULT hr = NULL;
            long lRet = 0;
            ULONG ulTmpSize = 0;

            m_pTxn                                  = NULL;
            m_pNewOrder                 = NULL;
            m_pPayment                  = NULL;
            m_pStockLevel = NULL;
            m_pOrderStatus              = NULL;

            m_bSinglePool = bSinglePool;

            ulTmpSize = (ULONG) sizeof(COM_DATA);
            VariantInit(&m_vTxn);
            m_vTxn.vt = VT_SAFEARRAY;

            m_vTxn.parray = SafeArrayCreateVector(VT_UI1, ulTmpSize,
ulTmpSize);
            if (!m_vTxn.parray)
                        throw new CCOMERR( E_FAIL );

            memset((void*)m_vTxn.parray->pvData,0,ulTmpSize);
            m_pTxn = (COM_DATA*)m_vTxn.parray->pvData;

            hr = CoInitializeEx(NULL, COINIT_MULTITHREADED);
            if (FAILED(hr))
            {
                        throw new CCOMERR( hr );
            }

            // create components
            if (m_bSinglePool)
            {
                        hr = CoCreateInstance(CLSID_TPCC, NULL, CLSCTX_SERVER,
IID_ITPCC, (void **)&m_pNewOrder);
                        if (FAILED(hr))
                                    throw new CCOMERR(hr);
```

```
                        // all txns will use same component
                        m_pPayment = m_pNewOrder;
                        m_pStockLevel = m_pNewOrder;
                        m_pOrderStatus = m_pNewOrder;
                }
                else
                {
                        // use different components for each txn

                        hr = CoCreateInstance(CLSID_NewOrder, NULL,
CLSCTX_SERVER, IID_ITPCC, (void **)&m_pNewOrder);
                        if (FAILED(hr))
                                throw new CCOMERR(hr);

                        hr = CoCreateInstance(CLSID_Payment, NULL,
CLSCTX_SERVER, IID_ITPCC, (void **)&m_pPayment);
                        if (FAILED(hr))
                                throw new CCOMERR(hr);

                        hr = CoCreateInstance(CLSID_StockLevel, NULL,
CLSCTX_SERVER, IID_ITPCC, (void **)&m_pStockLevel);
                        if (FAILED(hr))
                                throw new CCOMERR(hr);

                        hr = CoCreateInstance(CLSID_OrderStatus, NULL,
CLSCTX_SERVER, IID_ITPCC, (void **)&m_pOrderStatus);
                        if (FAILED(hr))
                                throw new CCOMERR(hr);
                }

                // call setcomplete to release each component back into pool
                hr = m_pNewOrder->CallSetComplete();
                if (FAILED(hr))
                        throw new CCOMERR(hr);

                if (!m_bSinglePool)
                {
                        hr = m_pPayment->CallSetComplete();
                        if (FAILED(hr))
                                throw new CCOMERR(hr);

                        hr = m_pStockLevel->CallSetComplete();
                        if (FAILED(hr))
                                throw new CCOMERR(hr);

                        hr = m_pOrderStatus->CallSetComplete();
                        if (FAILED(hr))
                                throw new CCOMERR(hr);
                }
}

CTPCC_COM::~CTPCC_COM()
{
                if (m_pTxn)
                        SafeArrayDestroy(m_vTxn.parray);

                ReleaseInterface(m_pNewOrder);
                if (!m_bSinglePool)
                {
                        ReleaseInterface(m_pPayment);
                        ReleaseInterface(m_pStockLevel);
                        ReleaseInterface(m_pOrderStatus);
                }
                CoUninitialize();
}

void CTPCC_COM::NewOrder()
{
                VARIANT   vTxn_out;

                HRESULT hr = m_pNewOrder->NewOrder(m_vTxn, &vTxn_out);
                if (FAILED(hr))
                        throw new CCOMERR( hr );
                memcpy(m_pTxn, (void *)vTxn_out.parray->pvData,vTxn_out.parray-
>rgsabound[0].cElements);
                SafeArrayDestroy(vTxn_out.parray);

                if ( m_pTxn->ErrorType != ERR_SUCCESS )
                        throw new CCOMERR( m_pTxn->ErrorType, m_pTxn-
>error );
}

void CTPCC_COM::Payment()
{
                VARIANT   vTxn_out;

                HRESULT hr = m_pPayment->Payment(m_vTxn, &vTxn_out);
                if (FAILED(hr))
                        throw new CCOMERR( hr );
                memcpy(m_pTxn, (void *)vTxn_out.parray->pvData,vTxn_out.parray-
>rgsabound[0].cElements);
                SafeArrayDestroy(vTxn_out.parray);

                if ( m_pTxn->ErrorType != ERR_SUCCESS )
                        throw new CCOMERR( m_pTxn->ErrorType, m_pTxn-
>error );
}

void CTPCC_COM::StockLevel()
{
                VARIANT   vTxn_out;

                HRESULT hr = m_pStockLevel->StockLevel(m_vTxn, &vTxn_out);
                if (FAILED(hr))
                        throw new CCOMERR( hr );
                memcpy(m_pTxn, (void *)vTxn_out.parray->pvData,vTxn_out.parray-
>rgsabound[0].cElements);
                SafeArrayDestroy(vTxn_out.parray);

                if ( m_pTxn->ErrorType != ERR_SUCCESS )
                        throw new CCOMERR( m_pTxn->ErrorType, m_pTxn-
>error );
}

void CTPCC_COM::OrderStatus()
{
                VARIANT   vTxn_out;

                HRESULT hr = m_pOrderStatus->OrderStatus(m_vTxn, &vTxn_out);
                if (FAILED(hr))
                        throw new CCOMERR( hr );
                memcpy(m_pTxn, (void *)vTxn_out.parray->pvData,vTxn_out.parray-
>rgsabound[0].cElements);
                SafeArrayDestroy(vTxn_out.parray);

                if ( m_pTxn->ErrorType != ERR_SUCCESS )
                        throw new CCOMERR( m_pTxn->ErrorType, m_pTxn-
>error );
}
```

tm_com_dll/src/tpcc_com.h

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

#pragma once

#include <stdio.h>
#include "..\..\tpcc_com_ps\src\tpcc_com_ps.h"

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

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

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

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

                        int                     m_hr;
                        int                     m_iErrorType;
                        int                     m_iError;

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

                        int ErrorNum() {return m_hr;}

                        char *ErrorText()
                        {
                                if (m_hr == S_OK)
                                        sprintf( m_szErrorText,
"Error: Class %d, error # %d", m_iErrorType, m_iError );
                                else
                                        sprintf( m_szErrorText,
"Error: COM HRESULT %x", m_hr );
                                return m_szErrorText;
                        }
};

class DllDecl CTPCC_COM : public CTPCC_BASE
{
                private:
                        BOOL m_bSinglePool;

                        // COM Interface pointers
                        ITPCC*                                  m_pNewOrder;
                        ITPCC*                                  m_pPayment;
                        ITPCC*
                m_pStockLevel;
                        ITPCC*
                m_pOrderStatus;

                        struct COM_DATA
                        {
                                int ErrorType;
                                int error;
                                union
                                {
                                        NEW_ORDER_DATA
                NewOrder;
                                        PAYMENT_DATA
                Payment;
                                        DELIVERY_DATA
                Delivery;
                                        STOCK_LEVEL_DATA
                StockLevel;
                                        ORDER_STATUS_DATA
                OrderStatus;
                                } u;
                        } *m_pTxn;

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

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

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


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


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

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

tpcc_com_all/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 **

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

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

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

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
```

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

odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ..\db_dblib_dll\bin\tpcc_dblib.lib ..\db_odbc_dll\bin\tpcc_odbc.lib
kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib

advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386

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

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

odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386
/pdbtype:sept
# ADD LINK32 ..\db_dblib_dll\bin\tpcc_dblib.lib ..\db_odbc_dll\bin\tpcc_odbc.lib
kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib

advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386

/pdbtype:sept

!ENDIF

# Begin Target

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

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

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

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

SOURCE=.\src\tpcc_com_all.idl

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

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

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

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

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

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

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

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

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

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

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

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

!ENDIF

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

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

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

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

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

tpcc_com_all/src/Methods.h

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


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

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

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

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

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

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

static void WriteMessageToEventLog(LPTSTR lpszMsg);

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

            CTPCC_Common();
            ~CTPCC_Common();

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

            HRESULT __stdcall CallSetComplete();

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

// IObjectConstruct
            STDMETHODIMP Construct(IDispatch * pUnk);

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

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

};

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

BEGIN_COM_MAP(CTPCC)
            COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
            COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

};


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

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

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

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

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

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


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

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

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


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

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

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

tpcc_com_all/src/resource.h

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

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

tpcc_com_all/src/tpcc_com_all.cpp

```
/*          FILE:                       TPCC_COM_ALL.CPP
 *                                                  Microsoft TPC-C Kit Ver.
4.20.000
 *                                                  Copyright Microsoft, 1999
 *             All Rights Reserved
 *
 *                                                  Version 4.10.000 audited
by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *          PURPOSE:    Implementation for TPC-C Tuxedo class.
 *          Contact:    Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                      4.20.000 - updated rev number to match kit
 */

#define STRICT
#define _WIN32_WINNT 0x0400
#define _ATL_APARTMENT_THREADED

#include <stdio.h>
#include <atlbase.h>
//You may derive a class from CComModule and use it if you want to override
//something, but do not change the name of _Module
extern CComModule _Module;

#include <atlcom.h>
#include <initguid.h>
#include <transact.h>
#include <atlimpl.cpp>
#include <comsvcs.h>

#include <sqltypes.h>
#include <sql.h>
#include <sqlext.h>

#include "tpcc_com_ps.h"
#include "..\..\common\src\trans.h"
            //tpckit transaction header contains definations of structures
specific

to TPC-C
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\error.h"
#include "..\..\common\src\ReadRegistry.h"
#include "..\..\db_dblib_dll\src\tpcc_dblib.h"                  // DBLIB
implementation of TPC-C txns
#include "..\..\db_odbc_dll\src\tpcc_odbc.h"                    // ODBC
implementation of TPC-C txns

#include "resource.h"
#include "tpcc_com_all.h"
#include "tpcc_com_all_i.c"
#include "Methods.h"
#include "..\..\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "..\..\common\src\ReadRegistry.cpp"


CComModule _Module;

BEGIN_OBJECT_MAP(ObjectMap)
            OBJECT_ENTRY(CLSID_TPCC, CTPCC)
            OBJECT_ENTRY(CLSID_NewOrder, CNewOrder)
            OBJECT_ENTRY(CLSID_OrderStatus, COrderStatus)
            OBJECT_ENTRY(CLSID_Payment, CPayment)
            OBJECT_ENTRY(CLSID_StockLevel, CStockLevel)
END_OBJECT_MAP()


// configuration settings from registry
TPCCREGISTRYDATA            Reg;
char
            szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

static  HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB            *pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC                         *pCTPCC_ODBC_new;


////////////////////////////////////////////////////////////////////////////
// DLL Entry Point

extern "C"
BOOL WINAPI DllMain(HINSTANCE hInstance, DWORD dwReason, LPVOID /*lpReserved*/)
{
            char szDllName[128];

            try
            {
                        if (dwReason == DLL_PROCESS_ATTACH)
                        {
                                    _Module.Init(ObjectMap, hInstance);
                                    DisableThreadLibraryCalls(hInstance);


                                    DWORD dwSize =
MAX_COMPUTERNAME_LENGTH+1;
                                    GetComputerName(szMyComputerName,
&dwSize);
                                    szMyComputerName[dwSize] = 0;

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

                                    if (Reg.eDB_Protocol == DBLIB)
                                    {
                                                strcpy( szDllName,
Reg.szPath );
                                                strcat( szDllName,
"tpcc_dblib.dll");
                                                hLibInstanceDb =
LoadLibrary( szDllName );
                                                if (hLibInstanceDb ==
NULL)
                                                            throw new
CCOMPONENT_ERR( ERR_LOADDLL_FAILED, szDllName, GetLastError() );
                                                // get function pointer to
wrapper for class constructor
                                                pCTPCC_DBLIB_new =
(TYPE_CTPCC_DBLIB*) GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_new");
                                                if (pCTPCC_DBLIB_new ==
NULL)
                                                            throw new
CCOMPONENT_ERR( ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                    }
                                    else if (Reg.eDB_Protocol == ODBC)
                                    {
                                                strcpy( szDllName,
Reg.szPath );
                                                strcat( szDllName,
"tpcc_odbc.dll");
                                                hLibInstanceDb =
LoadLibrary( szDllName );
                                                if (hLibInstanceDb ==
NULL)
                                                            throw new
CCOMPONENT_ERR( ERR_LOADDLL_FAILED, szDllName, GetLastError() );
                                                // get function pointer to
wrapper for class constructor
                                                pCTPCC_ODBC_new =
(TYPE_CTPCC_ODBC*) GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
                                                if (pCTPCC_ODBC_new ==
NULL)
                                                            throw new
CCOMPONENT_ERR( ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                    }
                                    else
                                                throw new
CCOMPONENT_ERR( ERR_UNKNOWN_DB_PROTOCOL );
                        }
                        else if (dwReason == DLL_PROCESS_DETACH)
                                    _Module.Term();

            }
            catch (CBaseErr *e)
            {
                        WriteMessageToEventLog(e->ErrorText());
                        delete e;
                        return FALSE;
            }
            catch (...)
            {
                        WriteMessageToEventLog(TEXT("Unhandled exception in
object DllMain"));
                        return FALSE;
            }

            return TRUE; // OK
}

////////////////////////////////////////////////////////////////////////////
// Used to determine whether the DLL can be unloaded by OLE

STDAPI DllCanUnloadNow(void)
{
            return (_Module.GetLockCount()==0) ? S_OK : S_FALSE;
}

////////////////////////////////////////////////////////////////////////////
// Returns a class factory to create an object of the requested type

STDAPI DllGetClassObject(REFCLSID rclsid, REFIID riid, LPVOID* ppv)
{
            return _Module.GetClassObject(rclsid, riid, ppv);
}

////////////////////////////////////////////////////////////////////////////
// DllRegisterServer - Adds entries to the system registry

STDAPI DllRegisterServer(void)
{
            // registers object, typelib and all interfaces in typelib
            return _Module.RegisterServer(TRUE);
}

////////////////////////////////////////////////////////////////////////////
// DllUnregisterServer - Removes entries from the system registry

STDAPI DllUnregisterServer(void)
{
            _Module.UnregisterServer();
            return S_OK;
}


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

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

    _stprintf(szMsg, TEXT("Error in COM+ TPC-C Component: "));
```

```
    lpszStrings[0] = szMsg;
    lpszStrings[1] = lpszMsg;

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

        (VOID) DeregisterEventSource(hEventSource);
    }
}

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

/* FUNCTION: CCOMPONENT_ERR::ErrorText
 *
 */

char* CCOMPONENT_ERR::ErrorText(void)
{
            static SERRORMSG errorMsgs[] =
            {
                        { ERR_MISSING_REGISTRY_ENTRIES,         "Required
entries missing from registry."
            },
                        { ERR_LOADDLL_FAILED,
            "Load of DLL failed.  DLL="


},
                        { ERR_GETPROCADDR_FAILED,               "Could not
map proc in DLL.  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(VARIANT txn_in, VARIANT* txn_out)
{
            PNEW_ORDER_DATA             pNewOrder;
            COM_DATA                    *pData;
            try
            {
                        pData = (COM_DATA*)txn_in.parray->pvData;
                        pNewOrder = m_pTxn->BuffAddr_NewOrder();

                        memcpy(pNewOrder, &pData->u.NewOrder,
sizeof(NEW_ORDER_DATA));
                        m_pTxn->NewOrder();                         // do the
actual txn

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

            txn_in.parray->rgsabound->cElements,

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

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

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

                        pData->retval = e->ErrorType();
                        pData->error = e->ErrorNum();
                        delete e;
                        return E_FAIL;
            }
            catch (...)
            {
                        WriteMessageToEventLog(TEXT("Unhandled exception."));
                        pData->retval = ERR_TYPE_LOGIC;
                        pData->error = 0;
                        m_bCanBePooled = FALSE;
                        return E_FAIL;
            }
}

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

                        memcpy(pPayment, &pData->u.Payment,
sizeof(PAYMENT_DATA));
                        m_pTxn->Payment();                          // do the
actual txn

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

            txn_in.parray->rgsabound->cElements,
```

```
                        pData = (COM_DATA*) txn_out->parray->pvData;

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

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

                        pData->retval = e->ErrorType();
                        pData->error = e->ErrorNum();
                        delete e;
                        return E_FAIL;
            }
            catch (...)
            {
                        WriteMessageToEventLog(TEXT("Unhandled exception."));
                        pData->retval = ERR_TYPE_LOGIC;
                        pData->error = 0;
                        m_bCanBePooled = FALSE;
                        return E_FAIL;
            }
}

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

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

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

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

            txn_in.parray->rgsabound->cElements,

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

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

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

                        pData->retval = e->ErrorType();
                        pData->error = e->ErrorNum();
                        delete e;
                        return E_FAIL;
            }
            catch (...)
            {
                        WriteMessageToEventLog(TEXT("Unhandled exception."));
                        pData->retval = ERR_TYPE_LOGIC;
                        pData->error = 0;
                        m_bCanBePooled = FALSE;
                        return E_FAIL;
            }
}

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

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

                        m_pTxn->OrderStatus();

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

            txn_in.parray->rgsabound->cElements,
```

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

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

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

                        pData->retval = e->ErrorType();
                        pData->error = e->ErrorNum();
                        delete e;
                        return E_FAIL;
            }
            catch (...)
            {
                        WriteMessageToEventLog(TEXT("Unhandled exception."));
                        pData->retval = ERR_TYPE_LOGIC;
                        pData->error = 0;
                        m_bCanBePooled = FALSE;
                        return E_FAIL;
            }
}
```

tpcc_com_all/src/tpcc_com_all.def

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

LIBRARY      "tpcc_com_all.dll"

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

tpcc_com_all/src/tpcc_com_all.h

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

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


 /* File created by MIDL compiler version 5.03.0280 */
/* at Thu Dec 13 23:13:14 2001
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


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

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

#ifndef __tpcc_com_all_h__
#define __tpcc_com_all_h__

/* Forward Declarations */

#ifndef __TPCC_FWD_DEFINED__
#define __TPCC_FWD_DEFINED__

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

#endif       /* __TPCC_FWD_DEFINED__ */


#ifndef __NewOrder_FWD_DEFINED__
#define __NewOrder_FWD_DEFINED__

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

#endif       /* __NewOrder_FWD_DEFINED__ */


#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;


#ifndef __TPCCLib_LIBRARY_DEFINED__
#define __TPCCLib_LIBRARY_DEFINED__

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


EXTERN_C const IID LIBID_TPCCLib;

EXTERN_C const CLSID CLSID_TPCC;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_NewOrder;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_OrderStatus;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_Payment;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_StockLevel;

#ifdef __cplusplus

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

/* Additional Prototypes for ALL interfaces */

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

tpcc_com_all/src/tpcc_com_all.idl

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

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


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

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

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


            [
                        uuid(975BAABF-84A7-11D2-BA47-00C04FBFE08B),
                        helpstring("NewOrder Class")
            ]
            coclass NewOrder
            {
                        [default] interface ITPCC;
            };


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


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


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

};
```

tpcc_com_all/src/tpcc_com_all.rc

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

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

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

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

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

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

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

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

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

#endif    // APSTUDIO_INVOKED


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

VS_VERSION_INFO VERSIONINFO
 FILEVERSION 1,0,0,1
 PRODUCTVERSION 1,0,0,1
 FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
 FILEFLAGS 0x1L
#else
 FILEFLAGS 0x0L
#endif
 FILEOS 0x4L
 FILETYPE 0x2L
 FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904B0"
        BEGIN
            VALUE "CompanyName", "\0"
            VALUE "FileDescription", "tpcc_com_all Module\0"
            VALUE "FileVersion", "1, 0, 0, 1\0"
            VALUE "InternalName", "TPCCNEWORDER\0"
            VALUE "LegalCopyright", "Copyright 1997\0"
            VALUE "OriginalFilename", "tpcc_com_all.DLL\0"
            VALUE "ProductName", "tpcc_com_all Module\0"
            VALUE "ProductVersion", "1, 0, 0, 1\0"
            VALUE "OLESelfRegister", "\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END

#endif    // !_MAC


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

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

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

STRINGTABLE DISCARDABLE
BEGIN
    IDS_PROJNAME            "tpcc_com_all"
END

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


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

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

tpcc_com_all/src/tpcc_com_all.rgs

```
HKCR
{
            TPCC.AllTxns.1 = s 'All Txns Class'
            {
                        CLSID = s '{122A3128-2520-11D3-BA71-00C04FBFE08B}'
            }
            TPCC.AllTxns = s 'TPCC Class'
            {
                        CurVer = s 'TPCC.AllTxns.1'
            }
            NoRemove CLSID
            {
                        ForceRemove {122A3128-2520-11D3-BA71-00C04FBFE08B} =
s 'TPCC Class'
                        {
                                    ProgID = s 'TPCC.AllTxns.1'
                                    VersionIndependentProgID = s
'TPCC.AllTxns'
                                    InprocServer32 = s '%MODULE%'
                                    {
                                                val ThreadingModel = s
'Both'
                                    }
                        }
            }
}
```

tpcc_com_all/src/tpcc_com_all_i.c

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

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

/* link this file in with the server and any clients */


 /* File created by MIDL compiler version 5.03.0280 */
/* at Thu Dec 13 23:13:14 2001
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

#if !defined(_M_IA64) && !defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif


#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char  c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x252O,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC,0x122A3128,0x252O,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,0xE0,0x
8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_Payment,0xCDO2F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x0O,0xC0,0x4F,0xBF,0xE0,0x8
B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif


#endif /* !defined(_M_IA64) && !defined(_M_AXP64)*/


#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */


 /* File created by MIDL compiler version 5.03.0280 */
/* at Thu Dec 13 23:13:14 2001
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext,
robust
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

#if defined(_M_IA64) || defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif


#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char  c[8];
} IID;
```

```
#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B)
;


MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x
8B);


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,0xE0,0x8
B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif


#endif /* defined(_M_IA64) || defined(_M_AXP64)*/
```

tpcc_com_all/src/tpcc_com_no.rgs

```
HKCR
{
            TPCC.NewOrder.1 = s 'NewOrder Class'
            {
                        CLSID = s '{975BAABF-84A7-11D2-BA47-00C04FBFE08B}'
            }
            TPCC.NewOrder = s 'NewOrder Class'
            {
                        CurVer = s 'TPCC.NewOrder.1'
            }
            NoRemove CLSID
            {
                        ForceRemove {975BAABF-84A7-11D2-BA47-00C04FBFE08B} =
s 'NewOrder Class'
                        {
                                    ProgID = s 'TPCC.NewOrder.1'
                                    VersionIndependentProgID = s
'TPCC.NewOrder'
                                    InprocServer32 = s '%MODULE%'
                                    {
                                                val ThreadingModel = s
'Both'
                                    }
                        }
            }
}
```

tpcc_com_all/src/tpcc_com_os.rgs

```
HKCR
{
            TPCC.OrderStatus.1 = s 'OrderStatus Class'
            {
                        CLSID = s '{266836AD-A50D-11D2-BA4E-00C04FBFE08B}'
            }
            TPCC.OrderStatus = s 'OrderStatus Class'
            {
                        CurVer = s 'TPCC.OrderStatus.1'
            }
            NoRemove CLSID
            {
                        ForceRemove {266836AD-A50D-11D2-BA4E-00C04FBFE08B} =
s 'OrderStatus Class'
                        {
                                    ProgID = s 'TPCC.OrderStatus.1'
                                    VersionIndependentProgID = s
'TPCC.OrderStatus'
                                    InprocServer32 = s '%MODULE%'
                                    {
                                                val ThreadingModel = s
'Both'
                                    }
                        }
            }
}
```

tpcc_com_all/src/tpcc_com_pay.rgs

```
HKCR
{
            TPCC.Payment.1 = s 'Payment Class'
            {
                        CLSID = s '{CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B}'
            }
            TPCC.Payment = s 'Payment Class'
            {
                        CurVer = s 'TPCC.Payment.1'
            }
            NoRemove CLSID
            {
                        ForceRemove {CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B} =
s 'Payment Class'
                        {
                                    ProgID = s 'TPCC.Payment.1'
                                    VersionIndependentProgID = s
'TPCC.Payment'
                                    InprocServer32 = s '%MODULE%'
                                    {
                                                val ThreadingModel = s
'Both'
                                    }
                        }
            }
}
```

tpcc_com_all/src/tpcc_com_ps.h

```
#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the definitions for the interfaces */


 /* File created by MIDL compiler version 5.03.0280 */
/* at Thu Dec 13 23:13:08 2001
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


/* verify that the <rpcndr.h> version is high enough to compile this file*/
#ifndef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440
#endif

#include "rpc.h"
#include "rpcndr.h"

#ifndef __RPCNDR_H_VERSION__
#error this stub requires an updated version of <rpcndr.h>
#endif // __RPCNDR_H_VERSION__

#ifndef COM_NO_WINDOWS_H
#include "windows.h"
#include "ole2.h"
#endif /*COM_NO_WINDOWS_H*/

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

/* Forward Declarations */

#ifndef __ITPCC_FWD_DEFINED__
#define __ITPCC_FWD_DEFINED__
typedef interface ITPCC ITPCC;
#endif       /* __ITPCC_FWD_DEFINED__ */


/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"

#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_ps_0000 */
/* [local] */



extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

/* interface ITPCC */
/* [unique][helpstring][uuid][oleautomation][object] */


EXTERN_C const IID IID_ITPCC;

#if defined(__cplusplus) && !defined(CINTERFACE)

    MIDL_INTERFACE("FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
    ITPCC : public IUnknown
    {
    public:
        virtual HRESULT __stdcall NewOrder(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

        virtual HRESULT __stdcall Payment(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

        virtual HRESULT __stdcall Delivery(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

        virtual HRESULT __stdcall StockLevel(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

        virtual HRESULT __stdcall OrderStatus(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

        virtual HRESULT __stdcall CallSetComplete( void) = 0;

    };

#else        /* C style interface */

    typedef struct ITPCCVtbl
    {
        BEGIN_INTERFACE

        HRESULT ( STDMETHODCALLTYPE __RPC_FAR *QueryInterface )(
            ITPCC __RPC_FAR * This,
            /* [in] */ REFIID riid,
            /* [iid_is][out] */ void __RPC_FAR *__RPC_FAR *ppvObject);

        ULONG ( STDMETHODCALLTYPE __RPC_FAR *AddRef )(
            ITPCC __RPC_FAR * This);

        ULONG ( STDMETHODCALLTYPE __RPC_FAR *Release )(
            ITPCC __RPC_FAR * This);

        HRESULT ( __stdcall __RPC_FAR *NewOrder )(
            ITPCC __RPC_FAR * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out);

        HRESULT ( __stdcall __RPC_FAR *Payment )(
            ITPCC __RPC_FAR * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out);

        HRESULT ( __stdcall __RPC_FAR *Delivery )(
            ITPCC __RPC_FAR * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out);

        HRESULT ( __stdcall __RPC_FAR *StockLevel )(
            ITPCC __RPC_FAR * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out);

        HRESULT ( __stdcall __RPC_FAR *OrderStatus )(
            ITPCC __RPC_FAR * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out);

        HRESULT ( __stdcall __RPC_FAR *CallSetComplete )(
            ITPCC __RPC_FAR * This);

        END_INTERFACE
    } ITPCCVtbl;

    interface ITPCC
    {
        CONST_VTBL struct ITPCCVtbl __RPC_FAR *lpVtbl;
    };


#ifdef COBJMACROS


#define ITPCC_QueryInterface(This,riid,ppvObject)     \
    (This)->lpVtbl -> QueryInterface(This,riid,ppvObject)

#define ITPCC_AddRef(This) \
    (This)->lpVtbl -> AddRef(This)

#define ITPCC_Release(This)              \
    (This)->lpVtbl -> Release(This)


#define ITPCC_NewOrder(This,txn_in,txn_out)           \
    (This)->lpVtbl -> NewOrder(This,txn_in,txn_out)

#define ITPCC_Payment(This,txn_in,txn_out)            \
    (This)->lpVtbl -> Payment(This,txn_in,txn_out)
```

```
#define ITPCC_Delivery(This,txn_in,txn_out)          \
    (This)->lpVtbl -> Delivery(This,txn_in,txn_out)

#define ITPCC_StockLevel(This,txn_in,txn_out)        \
    (This)->lpVtbl -> StockLevel(This,txn_in,txn_out)

#define ITPCC_OrderStatus(This,txn_in,txn_out)       \
    (This)->lpVtbl -> OrderStatus(This,txn_in,txn_out)

#define ITPCC_CallSetComplete(This)     \
    (This)->lpVtbl -> CallSetComplete(This)

#endif /* COBJMACROS */


#endif       /* C style interface */


HRESULT __stdcall ITPCC_NewOrder_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_Payment_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_Delivery_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_StockLevel_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_OrderStatus_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC __RPC_FAR * This);


void __RPC_STUB ITPCC_CallSetComplete_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);



#endif       /* __ITPCC_INTERFACE_DEFINED__ */


/* Additional Prototypes for ALL interfaces */

unsigned long             __RPC_USER  VARIANT_UserSize(     unsigned long
__RPC_FAR *, unsigned long            , VARIANT __RPC_FAR * );
unsigned char __RPC_FAR * __RPC_USER  VARIANT_UserMarshal(  unsigned long
__RPC_FAR *, unsigned char __RPC_FAR *, VARIANT __RPC_FAR * );
unsigned char __RPC_FAR * __RPC_USER  VARIANT_UserUnmarshal(unsigned long
__RPC_FAR *, unsigned char __RPC_FAR *, VARIANT __RPC_FAR * );
void                      __RPC_USER  VARIANT_UserFree(     unsigned long
__RPC_FAR *, VARIANT __RPC_FAR * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

tpcc_com_all/src/tpcc_com_sl.rgs

```
HKCR
{
            TPCC.StockLevel.1 = s 'StockLevel Class'
            {
                        CLSID = s '{2668369E-A50D-11D2-BA4E-00C04FBFE08B}'
            }
            TPCC.StockLevel = s 'StockLevel Class'
            {
                        CurVer = s 'TPCC.StockLevel.1'
            }
            NoRemove CLSID
            {
                        ForceRemove {2668369E-A50D-11D2-BA4E-00C04FBFE08B} =
s 'StockLevel Class'
                        {
                                    ProgID = s 'TPCC.StockLevel.1'
                                    VersionIndependentProgID = s
'TPCC.StockLevel'
                                    InprocServer32 = s '%MODULE%'
                                    {
                                                val ThreadingModel = s
'Both'
                                    }
                        }
            }
}
```

tpcc_com_ps/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 **

# TARGTYPE "Win32 (x86) Application" 0x0101

CFG=tpcc_com_ps - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_ps.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_ps.mak" CFG="tpcc_com_ps - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpcc_com_ps - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "tpcc_com_ps - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "tpcc_com_ps - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D _WIN32_WINNT=0x0400 /D
"REGISTER_PROXY_DLL" /FD /c
# SUBTRACT CPP /YX
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib

odbc32.lib odbccp32.lib /nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpcrt4.lib oleaut32.lib uuid.lib
/nologo /entry:"DllMain" /subsystem:windows /dll /pdb:none

/machine:I386 /def:".\src\tpcc_com_ps.def"
# Begin Custom Build - Copying tpcc_com_ps.h
InputPath=.\bin\tpcc_com_ps.dll
SOURCE="$(InputPath)"

"..\tpcc_com_all\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
            copy    .\src\tpcc_com_ps.h    ..\tpcc_com_all\src\

# End Custom Build

!ELSEIF  "$(CFG)" == "tpcc_com_ps - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /ZI /Od /D "WIN32" /D "_DEBUG" /D _WIN32_WINNT=0x0400 /D
"REGISTER_PROXY_DLL" /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib

odbc32.lib odbccp32.lib /nologo /subsystem:windows /debug /machine:I386
/pdbtype:sept
# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpcrt4.lib oleaut32.lib uuid.lib
/nologo /entry:"DllMain" /dll /debug /machine:IX86

/def:".\src\tpcc_com_ps.def" /pdbtype:sept
# SUBTRACT LINK32 /pdb:none
# Begin Custom Build - Copying tpcc_com_ps.h
InputPath=.\bin\tpcc_com_ps.dll
SOURCE="$(InputPath)"

"..\tpcc_com_all\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
            copy    .\src\tpcc_com_ps.h    ..\tpcc_com_all\src\

# End Custom Build

!ENDIF

# Begin Target

# Name "tpcc_com_ps - Win32 Release"
# Name "tpcc_com_ps - Win32 Debug"
# Begin Group "Source"

# PROP Default_Filter ""
# Begin Source File

SOURCE=.\src\dlldata.c
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps.def
# PROP Exclude_From_Build 1
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps.idl

!IF  "$(CFG)" == "tpcc_com_ps - Win32 Release"

# PROP Ignore_Default_Tool 1
# Begin Custom Build
InputPath=.\src\tpcc_com_ps.idl

BuildCmds= \
            midl /Oicf /h "tpcc_com_ps.h" /iid "tpcc_com_ps_i.c"
".\src\tpcc_com_ps.idl"    /out ".\src"

".\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_ps_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\dlldata.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_ps_p.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)
# End Custom Build

!ELSEIF  "$(CFG)" == "tpcc_com_ps - Win32 Debug"

# PROP Ignore_Default_Tool 1
# Begin Custom Build
InputPath=.\src\tpcc_com_ps.idl

BuildCmds= \
            midl /Oicf /h "tpcc_com_ps.h" /iid "tpcc_com_ps_i.c"
".\src\tpcc_com_ps.idl"    /out ".\src"
```

```
".\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_ps_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\dlldata.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_ps_p.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)
# End Custom Build

!ENDIF

# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps_i.c
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps_p.c
# End Source File
# End Group
# End Target
# End Project
```

tpcc_com_ps/src/dlldata.c

```
/*********************************************************
   DllData file -- generated by MIDL compiler

        DO NOT ALTER THIS FILE

   This file is regenerated by MIDL on every IDL file compile.

   To completely reconstruct this file, delete it and rerun MIDL
   on all the IDL files in this DLL, specifying this file for the
   /dlldata command line option

*********************************************************/


#include <rpcproxy.h>

#ifdef __cplusplus
extern "C"   {
#endif

EXTERN_PROXY_FILE( tpcc_com_ps )


PROXYFILE_LIST_START
/* Start of list */
  REFERENCE_PROXY_FILE( tpcc_com_ps ),
/* End of list */
PROXYFILE_LIST_END


DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )

#ifdef __cplusplus
}  /*extern "C" */
#endif

/* end of generated dlldata file */
```

tpcc_com_ps/src/tpcc_com_ps.def

```
LIBRARY      "tpcc_com_ps"

DESCRIPTION  'Proxy/Stub DLL'

EXPORTS
              DllGetClassObject       @1 PRIVATE
              DllCanUnloadNow         @2 PRIVATE
              GetProxyDllInfo         @3 PRIVATE
              DllRegisterServer                       @4            PRIVATE
              DllUnregisterServer                     @5            PRIVATE
```

tpcc_com_ps/src/tpcc_com_ps.h

```
#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the definitions for the interfaces */


 /* File created by MIDL compiler version 5.03.0280 */
/* at Thu Dec 13 23:13:08 2001
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


/* verify that the <rpcndr.h> version is high enough to compile this file*/
#ifndef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440
#endif

#include "rpc.h"
#include "rpcndr.h"

#ifndef __RPCNDR_H_VERSION__
#error this stub requires an updated version of <rpcndr.h>
#endif // __RPCNDR_H_VERSION__

#ifndef COM_NO_WINDOWS_H
#include "windows.h"
#include "ole2.h"
#endif /*COM_NO_WINDOWS_H*/

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

/* Forward Declarations */

#ifndef __ITPCC_FWD_DEFINED__
#define __ITPCC_FWD_DEFINED__
typedef interface ITPCC ITPCC;
#endif       /* __ITPCC_FWD_DEFINED__ */


/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"

#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_ps_0000 */
/* [local] */



extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

/* interface ITPCC */
/* [unique][helpstring][uuid][oleautomation][object] */


EXTERN_C const IID IID_ITPCC;

#if defined(__cplusplus) && !defined(CINTERFACE)

    MIDL_INTERFACE("FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
    ITPCC : public IUnknown
    {
    public:
        virtual HRESULT __stdcall NewOrder(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

        virtual HRESULT __stdcall Payment(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

        virtual HRESULT __stdcall Delivery(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

        virtual HRESULT __stdcall StockLevel(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

        virtual HRESULT __stdcall OrderStatus(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

        virtual HRESULT __stdcall CallSetComplete( void) = 0;

    };

#else        /* C style interface */

    typedef struct ITPCCVtbl
    {
        BEGIN_INTERFACE

        HRESULT ( STDMETHODCALLTYPE __RPC_FAR *QueryInterface )(
            ITPCC __RPC_FAR * This,
            /* [in] */ REFIID riid,
            /* [iid_is][out] */ void __RPC_FAR *__RPC_FAR *ppvObject);

        ULONG ( STDMETHODCALLTYPE __RPC_FAR *AddRef )(
            ITPCC __RPC_FAR * This);

        ULONG ( STDMETHODCALLTYPE __RPC_FAR *Release )(
            ITPCC __RPC_FAR * This);

        HRESULT ( __stdcall __RPC_FAR *NewOrder )(
            ITPCC __RPC_FAR * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out);

        HRESULT ( __stdcall __RPC_FAR *Payment )(
            ITPCC __RPC_FAR * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out);

        HRESULT ( __stdcall __RPC_FAR *Delivery )(
            ITPCC __RPC_FAR * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out);

        HRESULT ( __stdcall __RPC_FAR *StockLevel )(
            ITPCC __RPC_FAR * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out);

        HRESULT ( __stdcall __RPC_FAR *OrderStatus )(
            ITPCC __RPC_FAR * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out);

        HRESULT ( __stdcall __RPC_FAR *CallSetComplete )(
            ITPCC __RPC_FAR * This);

        END_INTERFACE
    } ITPCCVtbl;

    interface ITPCC
    {
        CONST_VTBL struct ITPCCVtbl __RPC_FAR *lpVtbl;
    };


#ifdef COBJMACROS


#define ITPCC_QueryInterface(This,riid,ppvObject)     \
    (This)->lpVtbl -> QueryInterface(This,riid,ppvObject)

#define ITPCC_AddRef(This) \
    (This)->lpVtbl -> AddRef(This)

#define ITPCC_Release(This)              \
    (This)->lpVtbl -> Release(This)


#define ITPCC_NewOrder(This,txn_in,txn_out)           \
    (This)->lpVtbl -> NewOrder(This,txn_in,txn_out)

#define ITPCC_Payment(This,txn_in,txn_out)            \
    (This)->lpVtbl -> Payment(This,txn_in,txn_out)

#define ITPCC_Delivery(This,txn_in,txn_out)           \
    (This)->lpVtbl -> Delivery(This,txn_in,txn_out)

#define ITPCC_StockLevel(This,txn_in,txn_out)         \
    (This)->lpVtbl -> StockLevel(This,txn_in,txn_out)

#define ITPCC_OrderStatus(This,txn_in,txn_out)        \
    (This)->lpVtbl -> OrderStatus(This,txn_in,txn_out)

#define ITPCC_CallSetComplete(This)     \
    (This)->lpVtbl -> CallSetComplete(This)

#endif /* COBJMACROS */


#endif       /* C style interface */


HRESULT __stdcall ITPCC_NewOrder_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_Payment_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_Delivery_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);
```

```
HRESULT __stdcall ITPCC_StockLevel_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_OrderStatus_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC __RPC_FAR * This);


void __RPC_STUB ITPCC_CallSetComplete_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


#endif      /* __ITPCC_INTERFACE_DEFINED__ */


/* Additional Prototypes for ALL interfaces */

unsigned long             __RPC_USER  VARIANT_UserSize(     unsigned long
__RPC_FAR *, unsigned long            , VARIANT __RPC_FAR * );
unsigned char __RPC_FAR * __RPC_USER  VARIANT_UserMarshal(  unsigned long
__RPC_FAR *, unsigned char __RPC_FAR *, VARIANT __RPC_FAR * );
unsigned char __RPC_FAR * __RPC_USER  VARIANT_UserUnmarshal(unsigned long
__RPC_FAR *, unsigned char __RPC_FAR *, VARIANT __RPC_FAR * );
void                      __RPC_USER  VARIANT_UserFree(     unsigned long
__RPC_FAR *, VARIANT __RPC_FAR * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

tpcc_com_ps/src/tpcc_com_ps.idl

```
/*          FILE:                           ITPCC.IDL
 *                                                      Microsoft TPC-C Kit Ver.
4.20.000
 *                                                      Copyright Microsoft, 1999
 *             All Rights Reserved
 *
 *                                                      not yet audited
 *
 *          PURPOSE:     Defines the interface used by TPCC. This interface
can be implemented by C++ components.
 *
 *  Change history:
 *                       4.20.000 - first version
 */

// Forward declare all types defined
interface ITPCC;
import "oaidl.idl";
import "ocidl.idl";

            [
                        object,
                        oleautomation,
                        uuid(FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B),
                        helpstring("ITPCC Interface"),
                        pointer_default(unique)
            ]
            interface ITPCC : IUnknown
            {

        HRESULT _stdcall NewOrder

              (

              [in] VARIANT  txn_in,

              [out] VARIANT *txn_out

              );

        HRESULT _stdcall Payment

              (

              [in] VARIANT txn_in,

              [out] VARIANT *txn_out

              );

        HRESULT _stdcall Delivery

              (

              [in] VARIANT txn_in,

              [out] VARIANT *txn_out

              );

        HRESULT _stdcall StockLevel

              (

              [in] VARIANT txn_in,

              [out] VARIANT *txn_out

              );

        HRESULT _stdcall OrderStatus

              (

              [in] VARIANT txn_in,

              [out] VARIANT *txn_out

              );

        HRESULT _stdcall CallSetComplete

              (

              );

    };      // interface ITPCC
```

tpcc_com_ps/src/tpcc_com_ps_i.c

```
#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */


 /* File created by MIDL compiler version 5.03.0280 */
/* at Thu Dec 13 23:13:08 2001
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

#if !defined(_M_IA64) && !defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif


#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char  c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



#endif /* !defined(_M_IA64) && !defined(_M_AXP64)*/


#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */


 /* File created by MIDL compiler version 5.03.0280 */
/* at Thu Dec 13 23:13:08 2001
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext,
robust
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

#if defined(_M_IA64) || defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif


#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char  c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



#endif /* defined(_M_IA64) || defined(_M_AXP64)*/
```

tpcc_com_ps/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 Thu Dec 13 23:13:08 2001
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

#if !defined(_M_IA64) && !defined(_M_AXP64)
#define USE_STUBLESS_PROXY


/* verify that the <rpcproxy.h> version is high enough to compile this file*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 440
#endif


#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of <rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__


#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE   997
#define PROC_FORMAT_STRING_SIZE   193
#define TRANSMIT_AS_TABLE_SIZE    0
#define WIRE_MARSHAL_TABLE_SIZE   1

typedef struct _MIDL_TYPE_FORMAT_STRING
    {
    short          Pad;
    unsigned char  Format[ TYPE_FORMAT_STRING_SIZE ];
    } MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
    {
    short          Pad;
    unsigned char  Format[ PROC_FORMAT_STRING_SIZE ];
    } MIDL_PROC_FORMAT_STRING;


extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;


/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,
   GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */


/* Object interface: IUnknown, ver. 0.0,
   GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */


/* Object interface: ITPCC, ver. 0.0,
   GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */


extern const MIDL_STUB_DESC Object_StubDesc;


extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[] =
    {
    0,
    34,
    68,
    102,
    136,
    170
    };

static const MIDL_SERVER_INFO ITPCC_ServerInfo =
    {
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0
    };

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo =
    {
    &Object_StubDesc,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0
    };


CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    (void *)-1 /* ITPCC::NewOrder */ ,
    (void *)-1 /* ITPCC::Payment */ ,
    (void *)-1 /* ITPCC::Delivery */ ,
    (void *)-1 /* ITPCC::StockLevel */ ,
    (void *)-1 /* ITPCC::OrderStatus */ ,
    (void *)-1 /* ITPCC::CallSetComplete */
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,
    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

extern const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

static const MIDL_STUB_DESC Object_StubDesc =
    {
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.3.280 */
    0,
    UserMarshalRoutines,
    0,  /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0,  /* Reserved3 */
    0,  /* Reserved4 */
    0   /* Reserved5 */
    };

#pragma data_seg(".rdata")

static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
        {

            {
            VARIANT_UserSize
            ,VARIANT_UserMarshal
            ,VARIANT_UserUnmarshal
            ,VARIANT_UserFree
            }

        };


#if !defined(__RPC_WIN32__)
#error  Invalid build platform for this stub.
#endif

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this stub because it uses these
features:
#error    -Oif or -Oicf, [wire_marshal] or [user_marshal] attribute.
#error However, your C/C++ compilation flags indicate you intend to run this app
on earlier systems.
#error This app will die there with the RPC_X_WRONG_STUB_VERSION error.
#endif


static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
    {
        0,
        {

            /* Procedure NewOrder */

                                        0x33,                   /*
FC_AUTO_HANDLE */
                                        0x6c,                   /* Old Flags:
object, Oi2 */
/*  2 */    NdrFcLong( 0x0 ),           /* 0 */
/*  6 */    NdrFcShort( 0x3 ),          /* 3 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/*  8 */    NdrFcShort( 0x1c ),         /* x86 Stack size/offset = 28 */
#else
                                        NdrFcShort( 0x20 ),         /*  MIPS
Stack size/offset = 32 */
#endif
#else
                                        NdrFcShort( 0x20 ),         /* PPC Stack
size/offset = 32 */
#endif
#else
                                        NdrFcShort( 0x28 ),         /* Alpha
Stack size/offset = 40 */
#endif
/* 10 */    NdrFcShort( 0x0 ),          /* 0 */
/* 12 */    NdrFcShort( 0x8 ),          /* 8 */
/* 14 */    0x7,                        /* Oi2 Flags:  srv must size, clt must
size, has return, */
                                        0x3,                        /* 3 */

            /* Parameter txn_in */

/* 16 */    NdrFcShort( 0x8b ),         /* Flags:  must size, must free, in, by
val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 18 */    NdrFcShort( 0x4 ),          /* x86 Stack size/offset = 4 */
#else
                                        NdrFcShort( 0x8 ),          /*  MIPS
Stack size/offset = 8 */
#endif
#else
                                        NdrFcShort( 0x8 ),          /* PPC Stack
size/offset = 8 */
#endif
#else
                                        NdrFcShort( 0x8 ),          /* Alpha
Stack size/offset = 8 */
#endif
/* 20 */    NdrFcShort( 0x3c8 ),        /* Type Offset=968 */

            /* Parameter txn_out */

/* 22 */    NdrFcShort( 0x4113 ),       /* Flags:  must size, must free, out,
simple ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 24 */    NdrFcShort( 0x14 ),         /* x86 Stack size/offset = 20 */
#else
                                        NdrFcShort( 0x18 ),         /*  MIPS
Stack size/offset = 24 */
#endif
#else
                                        NdrFcShort( 0x18 ),         /* PPC Stack
size/offset = 24 */
#endif
#else
                                        NdrFcShort( 0x18 ),         /* Alpha
Stack size/offset = 24 */
#endif
/* 26 */    NdrFcShort( 0x3da ),        /* Type Offset=986 */

            /* Return value */

/* 28 */    NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 30 */    NdrFcShort( 0x18 ),         /* x86 Stack size/offset = 24 */
#else
                                        NdrFcShort( 0x1c ),         /*  MIPS
Stack size/offset = 28 */
#endif
#else
                                        NdrFcShort( 0x1c ),         /* PPC Stack
size/offset = 28 */
#endif
#else
                                        NdrFcShort( 0x20 ),         /* Alpha
Stack size/offset = 32 */
#endif
/* 32 */    0x8,                        /* FC_LONG */
                                        0x0,                        /* 0 */

            /* Procedure Payment */

/* 34 */    0x33,                       /* FC_AUTO_HANDLE */
                                        0x6c,                       /* Old Flags:
object, Oi2 */
/* 36 */    NdrFcLong( 0x0 ),           /* 0 */
/* 40 */    NdrFcShort( 0x4 ),          /* 4 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 42 */    NdrFcShort( 0x1c ),         /* x86 Stack size/offset = 28 */
#else
                                        NdrFcShort( 0x20 ),         /*  MIPS
Stack size/offset = 32 */
#endif
#else
                                        NdrFcShort( 0x20 ),         /* PPC Stack
size/offset = 32 */
#endif
#else
                                        NdrFcShort( 0x28 ),         /* Alpha
Stack size/offset = 40 */
#endif
/* 44 */    NdrFcShort( 0x0 ),          /* 0 */
/* 46 */    NdrFcShort( 0x8 ),          /* 8 */
/* 48 */    0x7,                        /* Oi2 Flags:  srv must size, clt must
size, has return, */
                                        0x3,                        /* 3 */

            /* Parameter txn_in */

/* 50 */    NdrFcShort( 0x8b ),         /* Flags:  must size, must free, in, by
val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 52 */    NdrFcShort( 0x4 ),          /* x86 Stack size/offset = 4 */
#else
                                        NdrFcShort( 0x8 ),          /*  MIPS
Stack size/offset = 8 */
#endif
```

```
#else                                   NdrFcShort( 0x8 ),      /* PPC Stack
size/offset = 8 */
#endif
#else                                   NdrFcShort( 0x8 ),      /* Alpha
Stack size/offset = 8 */
#endif
/* 54 */    NdrFcShort( 0x3c8 ),        /* Type Offset=968 */

            /* Parameter txn_out */

/* 56 */    NdrFcShort( 0x4113 ),       /* Flags:  must size, must free, out,
simple ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 58 */    NdrFcShort( 0x14 ),         /* x86 Stack size/offset = 20 */
#else                                   NdrFcShort( 0x18 ),     /*  MIPS
Stack size/offset = 24 */
#endif
#else                                   NdrFcShort( 0x18 ),     /* PPC Stack
size/offset = 24 */
#endif
#else                                   NdrFcShort( 0x18 ),     /* Alpha
Stack size/offset = 24 */
#endif
/* 60 */    NdrFcShort( 0x3da ),        /* Type Offset=986 */

            /* Return value */

/* 62 */    NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 64 */    NdrFcShort( 0x18 ),         /* x86 Stack size/offset = 24 */
#else                                   NdrFcShort( 0x1c ),     /*  MIPS
Stack size/offset = 28 */
#endif
#else                                   NdrFcShort( 0x1c ),     /* PPC Stack
size/offset = 28 */
#endif
#else                                   NdrFcShort( 0x20 ),     /* Alpha
Stack size/offset = 32 */
#endif
/* 66 */    0x8,                        /* FC_LONG */
                                        0x0,                    /* 0 */

            /* Procedure Delivery */

/* 68 */    0x33,                       /* FC_AUTO_HANDLE */
                                        0x6c,                   /* Old Flags:
object, Oi2 */
/* 70 */    NdrFcLong( 0x0 ),           /* 0 */
/* 74 */    NdrFcShort( 0x5 ),          /* 5 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 76 */    NdrFcShort( 0x1c ),         /* x86 Stack size/offset = 28 */
#else                                   NdrFcShort( 0x20 ),     /*  MIPS
Stack size/offset = 32 */
#endif
#else                                   NdrFcShort( 0x20 ),     /* PPC Stack
size/offset = 32 */
#endif
#else                                   NdrFcShort( 0x28 ),     /* Alpha
Stack size/offset = 40 */
#endif
/* 78 */    NdrFcShort( 0x0 ),          /* 0 */
/* 80 */    NdrFcShort( 0x8 ),          /* 8 */
/* 82 */    0x7,                        /* Oi2 Flags:  srv must size, clt must
size, has return, */
                                        0x3,                    /* 3 */

            /* Parameter txn_in */

/* 84 */    NdrFcShort( 0x8b ),         /* Flags:  must size, must free, in, by
val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 86 */    NdrFcShort( 0x4 ),          /* x86 Stack size/offset = 4 */
#else                                   NdrFcShort( 0x8 ),      /*  MIPS
Stack size/offset = 8 */
#endif
#else                                   NdrFcShort( 0x8 ),      /* PPC Stack
size/offset = 8 */
#endif
#else                                   NdrFcShort( 0x8 ),      /* Alpha
Stack size/offset = 8 */
#endif
/* 88 */    NdrFcShort( 0x3c8 ),        /* Type Offset=968 */

            /* Parameter txn_out */

/* 90 */    NdrFcShort( 0x4113 ),       /* Flags:  must size, must free, out,
simple ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 92 */    NdrFcShort( 0x14 ),         /* x86 Stack size/offset = 20 */
#else                                   NdrFcShort( 0x18 ),     /*  MIPS
Stack size/offset = 24 */
#endif
#else                                   NdrFcShort( 0x18 ),     /* PPC Stack
size/offset = 24 */
#endif
#else                                   NdrFcShort( 0x18 ),     /* Alpha
Stack size/offset = 24 */
#endif
/* 94 */    NdrFcShort( 0x3da ),        /* Type Offset=986 */

            /* Return value */

/* 96 */    NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 98 */    NdrFcShort( 0x18 ),         /* x86 Stack size/offset = 24 */
#else                                   NdrFcShort( 0x1c ),     /*  MIPS
Stack size/offset = 28 */
#endif
#else                                   NdrFcShort( 0x1c ),     /* PPC Stack
size/offset = 28 */
#endif
#else                                   NdrFcShort( 0x20 ),     /* Alpha
Stack size/offset = 32 */
#endif
/* 100 */   0x8,                        /* FC_LONG */
                                        0x0,                    /* 0 */

            /* Procedure StockLevel */

/* 102 */   0x33,                       /* FC_AUTO_HANDLE */
                                        0x6c,                   /* Old Flags:
object, Oi2 */
/* 104 */   NdrFcLong( 0x0 ),           /* 0 */
/* 108 */   NdrFcShort( 0x6 ),          /* 6 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 110 */   NdrFcShort( 0x1c ),         /* x86 Stack size/offset = 28 */
#else                                   NdrFcShort( 0x20 ),     /*  MIPS
Stack size/offset = 32 */
#endif
#else                                   NdrFcShort( 0x20 ),     /* PPC Stack
size/offset = 32 */
#endif
#else                                   NdrFcShort( 0x28 ),     /* Alpha
Stack size/offset = 40 */
#endif
/* 112 */   NdrFcShort( 0x0 ),          /* 0 */
/* 114 */   NdrFcShort( 0x8 ),          /* 8 */
/* 116 */   0x7,                        /* Oi2 Flags:  srv must size, clt must
size, has return, */
                                        0x3,                    /* 3 */

            /* Parameter txn_in */

/* 118 */   NdrFcShort( 0x8b ),         /* Flags:  must size, must free, in, by
val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 120 */   NdrFcShort( 0x4 ),          /* x86 Stack size/offset = 4 */
#else                                   NdrFcShort( 0x8 ),      /*  MIPS
Stack size/offset = 8 */
#endif
#else                                   NdrFcShort( 0x8 ),      /* PPC Stack
size/offset = 8 */
#endif
#else                                   NdrFcShort( 0x8 ),      /* Alpha
Stack size/offset = 8 */
#endif
/* 122 */   NdrFcShort( 0x3c8 ),        /* Type Offset=968 */

            /* Parameter txn_out */

/* 124 */   NdrFcShort( 0x4113 ),       /* Flags:  must size, must free, out,
simple ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 126 */   NdrFcShort( 0x14 ),         /* x86 Stack size/offset = 20 */
#else                                   NdrFcShort( 0x18 ),     /*  MIPS
Stack size/offset = 24 */
#endif
#else                                   NdrFcShort( 0x18 ),     /* PPC Stack
size/offset = 24 */
#endif
#else                                   NdrFcShort( 0x18 ),     /* Alpha
Stack size/offset = 24 */
#endif
/* 128 */   NdrFcShort( 0x3da ),        /* Type Offset=986 */

            /* Return value */

/* 130 */   NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 132 */   NdrFcShort( 0x18 ),         /* x86 Stack size/offset = 24 */
#else                                   NdrFcShort( 0x1c ),     /*  MIPS
Stack size/offset = 28 */
#endif
#else                                   NdrFcShort( 0x1c ),     /* PPC Stack
size/offset = 28 */
#endif
#else                                   NdrFcShort( 0x20 ),     /* Alpha
Stack size/offset = 32 */
#endif
/* 134 */   0x8,                        /* FC_LONG */
                                        0x0,                    /* 0 */

            /* Procedure OrderStatus */

/* 136 */   0x33,                       /* FC_AUTO_HANDLE */
                                        0x6c,                   /* Old Flags:
object, Oi2 */
/* 138 */   NdrFcLong( 0x0 ),           /* 0 */
/* 142 */   NdrFcShort( 0x7 ),          /* 7 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 144 */   NdrFcShort( 0x1c ),         /* x86 Stack size/offset = 28 */
#else                                   NdrFcShort( 0x20 ),     /*  MIPS
Stack size/offset = 32 */
#endif
#else                                   NdrFcShort( 0x20 ),     /* PPC Stack
size/offset = 32 */
#endif
#else                                   NdrFcShort( 0x28 ),     /* Alpha
Stack size/offset = 40 */
#endif
/* 146 */   NdrFcShort( 0x0 ),          /* 0 */
/* 148 */   NdrFcShort( 0x8 ),          /* 8 */
/* 150 */   0x7,                        /* Oi2 Flags:  srv must size, clt must
size, has return, */
                                        0x3,                    /* 3 */

            /* Parameter txn_in */

/* 152 */   NdrFcShort( 0x8b ),         /* Flags:  must size, must free, in, by
val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 154 */   NdrFcShort( 0x4 ),          /* x86 Stack size/offset = 4 */
#else                                   NdrFcShort( 0x8 ),      /*  MIPS
Stack size/offset = 8 */
#endif
#else                                   NdrFcShort( 0x8 ),      /* PPC Stack
size/offset = 8 */
#endif
#else                                   NdrFcShort( 0x8 ),      /* Alpha
Stack size/offset = 8 */
#endif
/* 156 */   NdrFcShort( 0x3c8 ),        /* Type Offset=968 */

            /* Parameter txn_out */

/* 158 */   NdrFcShort( 0x4113 ),       /* Flags:  must size, must free, out,
simple ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 160 */   NdrFcShort( 0x14 ),         /* x86 Stack size/offset = 20 */
#else                                   NdrFcShort( 0x18 ),     /*  MIPS
Stack size/offset = 24 */
#endif
#else                                   NdrFcShort( 0x18 ),     /* PPC Stack
size/offset = 24 */
#endif
#else                                   NdrFcShort( 0x18 ),     /* Alpha
Stack size/offset = 24 */
#endif
/* 162 */   NdrFcShort( 0x3da ),        /* Type Offset=986 */

            /* Return value */

/* 164 */   NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 166 */   NdrFcShort( 0x18 ),         /* x86 Stack size/offset = 24 */
#else                                   NdrFcShort( 0x1c ),     /*  MIPS
Stack size/offset = 28 */
#endif
#else                                   NdrFcShort( 0x1c ),     /* PPC Stack
size/offset = 28 */
#endif
#else                                   NdrFcShort( 0x20 ),     /* Alpha
Stack size/offset = 32 */
#endif
```

```
/* 168 */    0x8,                        /* FC_LONG */
             0x0,                        /* 0 */

             /* Procedure CallSetComplete */

/* 170 */    0x33,                       /* FC_AUTO_HANDLE */
             0x6c,                       /* Old Flags:
object, Oi2 */
/* 172 */    NdrFcLong( 0x0 ),           /* 0 */
/* 176 */    NdrFcShort( 0x8 ),          /* 8 */
#ifndef _ALPHA_
/* 178 */    NdrFcShort( 0x8 ),          /* x86, MIPS, PPC Stack size/offset = 8
*/
#else
                                         NdrFcShort( 0x10 ),         /* Alpha
Stack size/offset = 16 */
#endif
/* 180 */    NdrFcShort( 0x0 ),          /* 0 */
/* 182 */    NdrFcShort( 0x8 ),          /* 8 */
/* 184 */    0x4,                        /* Oi2 Flags:  has return, */
             0x1,                        /* 1 */

             /* Return value */

/* 186 */    NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 188 */    NdrFcShort( 0x4 ),          /* x86, MIPS, PPC Stack size/offset = 4
*/
#else
                                         NdrFcShort( 0x8 ),          /* Alpha
Stack size/offset = 8 */
#endif
/* 190 */    0x8,                        /* FC_LONG */
             0x0,                        /* 0 */

             0x0
        }
    };

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
    {
        0,
        {
                                         NdrFcShort( 0x0 ),          /* 0 */
/*  2 */
                                         0x12, 0x0,    /* FC_UP */
/*  4 */     NdrFcShort( 0x3b0 ),        /* Offset= 944 (948) */
/*  6 */
                                         0x2b,                       /*
FC_NON_ENCAPSULATED_UNION */
                                         0x9,                        /* FC_ULONG
*/
/*  8 */     0x7,                        /* Corr desc: FC_USHORT */
             0x0,                                                    /*  */
/* 10 */     NdrFcShort( 0xfff8 ),       /* -8 */
/* 12 */     NdrFcShort( 0x2 ),          /* Offset= 2 (14) */
/* 14 */     NdrFcShort( 0x10 ),         /* 16 */
/* 16 */     NdrFcShort( 0x2b ),         /* 43 */
/* 18 */     NdrFcLong( 0x3 ),           /* 3 */
/* 22 */     NdrFcShort( 0x8008 ),       /* Simple arm type: FC_LONG */
/* 24 */     NdrFcLong( 0x11 ),          /* 17 */
/* 28 */     NdrFcShort( 0x8001 ),       /* Simple arm type: FC_BYTE */
/* 30 */     NdrFcLong( 0x2 ),           /* 2 */
/* 34 */     NdrFcShort( 0x8006 ),       /* Simple arm type: FC_SHORT */
/* 36 */     NdrFcLong( 0x4 ),           /* 4 */
/* 40 */     NdrFcShort( 0x800a ),       /* Simple arm type: FC_FLOAT */
/* 42 */     NdrFcLong( 0x5 ),           /* 5 */
/* 46 */     NdrFcShort( 0x800c ),       /* Simple arm type: FC_DOUBLE */
/* 48 */     NdrFcLong( 0xb ),           /* 11 */
/* 52 */     NdrFcShort( 0x8006 ),       /* Simple arm type: FC_SHORT */
/* 54 */     NdrFcLong( 0xa ),           /* 10 */
/* 58 */     NdrFcShort( 0x8008 ),       /* Simple arm type: FC_LONG */
/* 60 */     NdrFcLong( 0x6 ),           /* 6 */
/* 64 */     NdrFcShort( 0xd6 ),         /* Offset= 214 (278) */
/* 66 */     NdrFcLong( 0x7 ),           /* 7 */
/* 70 */     NdrFcShort( 0x800c ),       /* Simple arm type: FC_DOUBLE */
/* 72 */     NdrFcLong( 0x8 ),           /* 8 */
/* 76 */     NdrFcShort( 0xd0 ),         /* Offset= 208 (284) */
/* 78 */     NdrFcLong( 0xd ),           /* 13 */
/* 82 */     NdrFcShort( 0xe2 ),         /* Offset= 226 (308) */
/* 84 */     NdrFcLong( 0x9 ),           /* 9 */
/* 88 */     NdrFcShort( 0xee ),         /* Offset= 238 (326) */
/* 90 */     NdrFcLong( 0x2000 ),        /* 8192 */
/* 94 */     NdrFcShort( 0xfa ),         /* Offset= 250 (344) */
/* 96 */     NdrFcLong( 0x24 ),          /* 36 */
/* 100 */    NdrFcShort( 0x308 ),        /* Offset= 776 (876) */
/* 102 */    NdrFcLong( 0x4024 ),        /* 16420 */
/* 106 */    NdrFcShort( 0x302 ),        /* Offset= 770 (876) */
/* 108 */    NdrFcLong( 0x4011 ),        /* 16401 */
/* 112 */    NdrFcShort( 0x300 ),        /* Offset= 768 (880) */
/* 114 */    NdrFcLong( 0x4002 ),        /* 16386 */
/* 118 */    NdrFcShort( 0x2fe ),        /* Offset= 766 (884) */
/* 120 */    NdrFcLong( 0x4003 ),        /* 16387 */
/* 124 */    NdrFcShort( 0x2fc ),        /* Offset= 764 (888) */
/* 126 */    NdrFcLong( 0x4004 ),        /* 16388 */
/* 130 */    NdrFcShort( 0x2fa ),        /* Offset= 762 (892) */
/* 132 */    NdrFcLong( 0x4005 ),        /* 16389 */
/* 136 */    NdrFcShort( 0x2f8 ),        /* Offset= 760 (896) */
/* 138 */    NdrFcLong( 0x400b ),        /* 16395 */
/* 142 */    NdrFcShort( 0x2e6 ),        /* Offset= 742 (884) */
/* 144 */    NdrFcLong( 0x400a ),        /* 16394 */
/* 148 */    NdrFcShort( 0x2e4 ),        /* Offset= 740 (888) */
/* 150 */    NdrFcLong( 0x4006 ),        /* 16390 */
/* 154 */    NdrFcShort( 0x2ea ),        /* Offset= 746 (900) */
/* 156 */    NdrFcLong( 0x4007 ),        /* 16391 */
/* 160 */    NdrFcShort( 0x2e0 ),        /* Offset= 736 (896) */
/* 162 */    NdrFcLong( 0x4008 ),        /* 16392 */
/* 166 */    NdrFcShort( 0x2e2 ),        /* Offset= 738 (904) */
/* 168 */    NdrFcLong( 0x400d ),        /* 16397 */
/* 172 */    NdrFcShort( 0x2e0 ),        /* Offset= 736 (908) */
/* 174 */    NdrFcLong( 0x4009 ),        /* 16393 */
/* 178 */    NdrFcShort( 0x2de ),        /* Offset= 734 (912) */
/* 180 */    NdrFcLong( 0x6000 ),        /* 24576 */
/* 184 */    NdrFcShort( 0x2dc ),        /* Offset= 732 (916) */
/* 186 */    NdrFcLong( 0x400c ),        /* 16396 */
/* 190 */    NdrFcShort( 0x2da ),        /* Offset= 730 (920) */
/* 192 */    NdrFcLong( 0x10 ),          /* 16 */
/* 196 */    NdrFcShort( 0x8002 ),       /* Simple arm type: FC_CHAR */
/* 198 */    NdrFcLong( 0x12 ),          /* 18 */
/* 202 */    NdrFcShort( 0x8006 ),       /* Simple arm type: FC_SHORT */
/* 204 */    NdrFcLong( 0x13 ),          /* 19 */
/* 208 */    NdrFcShort( 0x8008 ),       /* Simple arm type: FC_LONG */
/* 210 */    NdrFcLong( 0x16 ),          /* 22 */
/* 214 */    NdrFcShort( 0x8008 ),       /* Simple arm type: FC_LONG */
/* 216 */    NdrFcLong( 0x17 ),          /* 23 */
/* 220 */    NdrFcShort( 0x8008 ),       /* Simple arm type: FC_LONG */
/* 222 */    NdrFcLong( 0xe ),           /* 14 */
/* 226 */    NdrFcShort( 0x2be ),        /* Offset= 702 (928) */
/* 228 */    NdrFcLong( 0x400e ),        /* 16398 */
/* 232 */    NdrFcShort( 0x2c4 ),        /* Offset= 708 (940) */
/* 234 */    NdrFcLong( 0x4010 ),        /* 16400 */
/* 238 */    NdrFcShort( 0x2c2 ),        /* Offset= 706 (944) */
/* 240 */    NdrFcLong( 0x4012 ),        /* 16402 */
/* 244 */    NdrFcShort( 0x280 ),        /* Offset= 640 (884) */
/* 246 */    NdrFcLong( 0x4013 ),        /* 16403 */
/* 250 */    NdrFcShort( 0x27e ),        /* Offset= 638 (888) */
/* 252 */    NdrFcLong( 0x4016 ),        /* 16406 */
/* 256 */    NdrFcShort( 0x278 ),        /* Offset= 632 (888) */
/* 258 */    NdrFcLong( 0x4017 ),        /* 16407 */
/* 262 */    NdrFcShort( 0x272 ),        /* Offset= 626 (888) */
/* 264 */    NdrFcLong( 0x0 ),           /* 0 */
/* 268 */    NdrFcShort( 0x0 ),          /* Offset= 0 (268) */
/* 270 */    NdrFcLong( 0x1 ),           /* 1 */
/* 274 */    NdrFcShort( 0x0 ),          /* Offset= 0 (274) */
/* 276 */    NdrFcShort( 0xffffffff ),   /* Offset= -1 (275) */
/* 278 */
                                         0x15,                       /* FC_STRUCT
*/
                                         0x7,                        /* 7 */
/* 280 */    NdrFcShort( 0x8 ),          /* 8 */
/* 282 */    0xb,                        /* FC_HYPER */
                                         0x5b,                       /* FC_END */
/* 284 */
                                         0x12, 0x0,    /* FC_UP */
/* 286 */    NdrFcShort( 0xc ),          /* Offset= 12 (298) */
/* 288 */
                                         0x1b,                       /* FC_CARRAY
*/
                                         0x1,                        /* 1 */
/* 290 */    NdrFcShort( 0x2 ),          /* 2 */
/* 292 */    0x9,                        /* Corr desc: FC_ULONG */
                                         0x0,                        /*  */
/* 294 */    NdrFcShort( 0xfffc ),       /* -4 */
/* 296 */    0x6,                        /* FC_SHORT */
                                         0x5b,                       /* FC_END */
/* 298 */
                                         0x17,                       /* FC_CSTRUCT
*/
                                         0x3,                        /* 3 */
/* 300 */    NdrFcShort( 0x8 ),          /* 8 */
/* 302 */    NdrFcShort( 0xfffffff2 ),   /* Offset= -14 (288) */
/* 304 */    0x8,                        /* FC_LONG */
                                         0x8,                        /* FC_LONG */
/* 306 */    0x5c,                       /* FC_PAD */
                                         0x5b,                       /* FC_END */
/* 308 */
                                         0x2f,                       /* FC_IP */
                                         0x5a,                       /*
FC_CONSTANT_IID */
/* 310 */    NdrFcLong( 0x0 ),           /* 0 */
/* 314 */    NdrFcShort( 0x0 ),          /* 0 */
/* 316 */    NdrFcShort( 0x0 ),          /* 0 */
/* 318 */    0xc0,                       /* 192 */
                                         0x0,                        /* 0 */
/* 320 */    0x0,                        /* 0 */
                                         0x0,                        /* 0 */
/* 322 */    0x0,                        /* 0 */
                                         0x0,                        /* 0 */
/* 324 */    0x0,                        /* 0 */
                                         0x46,                       /* 70 */
/* 326 */
                                         0x2f,                       /* FC_IP */
                                         0x5a,                       /*
FC_CONSTANT_IID */
/* 328 */    NdrFcLong( 0x20400 ),       /* 132096 */
/* 332 */    NdrFcShort( 0x0 ),          /* 0 */
/* 334 */    NdrFcShort( 0x0 ),          /* 0 */
/* 336 */    0xc0,                       /* 192 */
                                         0x0,                        /* 0 */
/* 338 */    0x0,                        /* 0 */
                                         0x0,                        /* 0 */
/* 340 */    0x0,                        /* 0 */
                                         0x0,                        /* 0 */
/* 342 */    0x0,                        /* 0 */
                                         0x46,                       /* 70 */
/* 344 */
                                         0x12, 0x10,  /* FC_UP [pointer_deref]
*/
/* 346 */    NdrFcShort( 0x2 ),          /* Offset= 2 (348) */
/* 348 */
                                         0x12, 0x0,    /* FC_UP */
/* 350 */    NdrFcShort( 0x1fc ),        /* Offset= 508 (858) */
/* 352 */
                                         0x2a,                       /*
FC_ENCAPSULATED_UNION */
                                         0x49,                       /* 73 */
/* 354 */    NdrFcShort( 0x18 ),         /* 24 */
/* 356 */    NdrFcShort( 0xa ),          /* 10 */
/* 358 */    NdrFcLong( 0x8 ),           /* 8 */
/* 362 */    NdrFcShort( 0x58 ),         /* Offset= 88 (450) */
/* 364 */    NdrFcLong( 0xd ),           /* 13 */
/* 368 */    NdrFcShort( 0x78 ),         /* Offset= 120 (488) */
/* 370 */    NdrFcLong( 0x9 ),           /* 9 */
/* 374 */    NdrFcShort( 0x94 ),         /* Offset= 148 (522) */
/* 376 */    NdrFcLong( 0xc ),           /* 12 */
/* 380 */    NdrFcShort( 0xbc ),         /* Offset= 188 (568) */
/* 382 */    NdrFcLong( 0x24 ),          /* 36 */
/* 386 */    NdrFcShort( 0x114 ),        /* Offset= 276 (662) */
/* 388 */    NdrFcLong( 0x800d ),        /* 32781 */
/* 392 */    NdrFcShort( 0x130 ),        /* Offset= 304 (696) */
/* 394 */    NdrFcLong( 0x10 ),          /* 16 */
/* 398 */    NdrFcShort( 0x148 ),        /* Offset= 328 (726) */
/* 400 */    NdrFcLong( 0x2 ),           /* 2 */
/* 404 */    NdrFcShort( 0x160 ),        /* Offset= 352 (756) */
/* 406 */    NdrFcLong( 0x3 ),           /* 3 */
/* 410 */    NdrFcShort( 0x178 ),        /* Offset= 376 (786) */
/* 412 */    NdrFcLong( 0x14 ),          /* 20 */
/* 416 */    NdrFcShort( 0x190 ),        /* Offset= 400 (816) */
/* 418 */    NdrFcShort( 0xffffffff ),   /* Offset= -1 (417) */
/* 420 */
                                         0x1b,                       /* FC_CARRAY
*/
                                         0x3,                        /* 3 */
/* 422 */    NdrFcShort( 0x4 ),          /* 4 */
/* 424 */    0x19,                       /* Corr desc:  field pointer, FC_ULONG
*/
                                         0x0,                        /*  */
/* 426 */    NdrFcShort( 0x0 ),          /* 0 */
/* 428 */
                                         0x4b,                       /* FC_PP */
                                         0x5c,                       /* FC_PAD */
/* 430 */
                                         0x48,                       /*
FC_VARIABLE_REPEAT */
                                         0x49,                       /*
FC_FIXED_OFFSET */
/* 432 */    NdrFcShort( 0x4 ),          /* 4 */
/* 434 */    NdrFcShort( 0x0 ),          /* 0 */
/* 436 */    NdrFcShort( 0x1 ),          /* 1 */
/* 438 */    NdrFcShort( 0x0 ),          /* 0 */
/* 440 */    NdrFcShort( 0x0 ),          /* 0 */
/* 442 */    0x12, 0x0,    /* FC_UP */
/* 444 */    NdrFcShort( 0xffffff6e ),   /* Offset= -146 (298) */
/* 446 */
                                         0x5b,                       /* FC_END */
                                         0x8,                        /* FC_LONG */
/* 448 */    0x5c,                       /* FC_PAD */
                                         0x5b,                       /* FC_END */
/* 450 */
                                         0x16,                       /* FC_PSTRUCT
*/
                                         0x3,                        /* 3 */
/* 452 */    NdrFcShort( 0x8 ),          /* 8 */
/* 454 */
                                         0x4b,                       /* FC_PP */
                                         0x5c,                       /* FC_PAD */
/* 456 */
                                         0x46,                       /*
FC_NO_REPEAT */
                                         0x5c,                       /* FC_PAD */
/* 458 */    NdrFcShort( 0x4 ),          /* 4 */
/* 460 */    NdrFcShort( 0x4 ),          /* 4 */
/* 462 */    0x11, 0x0,    /* FC_RP */
/* 464 */    NdrFcShort( 0xffffffd4 ),   /* Offset= -44 (420) */
/* 466 */
                                         0x5b,                       /* FC_END */
                                         0x8,                        /* FC_LONG */
/* 468 */    0x8,                        /* FC_LONG */
                                         0x5b,                       /* FC_END */
/* 470 */
                                         0x21,                       /*
FC_BOGUS_ARRAY */
                                         0x3,                        /* 3 */
/* 472 */    NdrFcShort( 0x0 ),          /* 0 */
/* 474 */    0x19,                       /* Corr desc:  field pointer, FC_ULONG
*/
                                         0x0,                        /*  */
/* 476 */    NdrFcShort( 0x0 ),          /* 0 */
/* 478 */    NdrFcLong( 0xffffffff ),    /* -1 */
/* 482 */    0x4c,                       /* FC_EMBEDDED_COMPLEX */
                                         0x0,                        /* 0 */
/* 484 */    NdrFcShort( 0xffffff50 ),   /* Offset= -176 (308) */
/* 486 */    0x5c,                       /* FC_PAD */
                                         0x5b,                       /* FC_END */
/* 488 */
                                         0x1a,                       /*
FC_BOGUS_STRUCT */
                                         0x3,                        /* 3 */
/* 490 */    NdrFcShort( 0x8 ),          /* 8 */
/* 492 */    NdrFcShort( 0x0 ),          /* 0 */
/* 494 */    NdrFcShort( 0x6 ),          /* Offset= 6 (500) */
/* 496 */    0x8,                        /* FC_LONG */
                                         0x36,                       /* FC_POINTER
*/
/* 498 */    0x5c,                       /* FC_PAD */
                                         0x5b,                       /* FC_END */
/* 500 */
                                         0x11, 0x0,    /* FC_RP */
/* 502 */    NdrFcShort( 0xfffffe0 ),    /* Offset= -32 (470) */
/* 504 */
                                         0x21,                       /*
FC_BOGUS_ARRAY */
                                         0x3,                        /* 3 */
/* 506 */    NdrFcShort( 0x0 ),          /* 0 */
/* 508 */    0x19,                       /* Corr desc:  field pointer, FC_ULONG
*/
                                         0x0,                        /*  */
/* 510 */    NdrFcShort( 0x0 ),          /* 0 */
/* 512 */    NdrFcLong( 0xffffffff ),    /* -1 */
/* 516 */    0x4c,                       /* FC_EMBEDDED_COMPLEX */
                                         0x0,                        /* 0 */
/* 518 */    NdrFcShort( 0xffffff40 ),   /* Offset= -192 (326) */
/* 520 */    0x5c,                       /* FC_PAD */
                                         0x5b,                       /* FC_END */
```

```
/* 522 */                                   0x1a,                   /*
FC_BOGUS_STRUCT */
                                            0x3,                    /* 3 */
/* 524 */     NdrFcShort( 0x8 ),            /* 8 */
/* 526 */     NdrFcShort( 0x0 ),            /* 0 */
/* 528 */     NdrFcShort( 0x6 ),            /* Offset= 6 (534) */
/* 530 */     0x8,                          /* FC_LONG */
                                            0x36,                   /* FC_POINTER
*/
/* 532 */     0x5c,                         /* FC_PAD */
                                            0x5b,                   /* FC_END */
/* 534 */
                                            0x11, 0x0,    /* FC_RP */
/* 536 */     NdrFcShort( 0xffffffe0 ),     /* Offset= -32 (504) */
/* 538 */
                                            0x1b,                   /* FC_CARRAY
*/
                                            0x3,                    /* 3 */
/* 540 */     NdrFcShort( 0x4 ),            /* 4 */
/* 542 */     0x19,                         /* Corr desc:  field pointer, FC_ULONG
*/
                                            0x0,                    /*  */
/* 544 */     NdrFcShort( 0x0 ),            /* 0 */
/* 546 */
                                            0x4b,                   /* FC_PP */
                                            0x5c,                   /* FC_PAD */
/* 548 */
                                            0x48,                   /*
FC_VARIABLE_REPEAT */
                                            0x49,                   /*
FC_FIXED_OFFSET */
/* 550 */     NdrFcShort( 0x4 ),            /* 4 */
/* 552 */     NdrFcShort( 0x0 ),            /* 0 */
/* 554 */     NdrFcShort( 0x1 ),            /* 1 */
/* 556 */     NdrFcShort( 0x0 ),            /* 0 */
/* 558 */     NdrFcShort( 0x0 ),            /* 0 */
/* 560 */     0x12, 0x0,    /* FC_UP */
/* 562 */     NdrFcShort( 0x182 ),          /* Offset= 386 (948) */
/* 564 */
                                            0x5b,                   /* FC_END */
                                            0x8,                    /* FC_LONG */
/* 566 */     0x5c,                         /* FC_PAD */
                                            0x5b,                   /* FC_END */
/* 568 */
                                            0x1a,                   /*
FC_BOGUS_STRUCT */
                                            0x3,                    /* 3 */
/* 570 */     NdrFcShort( 0x8 ),            /* 8 */
/* 572 */     NdrFcShort( 0x0 ),            /* 0 */
/* 574 */     NdrFcShort( 0x6 ),            /* Offset= 6 (580) */
/* 576 */     0x8,                          /* FC_LONG */
                                            0x36,                   /* FC_POINTER
*/
/* 578 */     0x5c,                         /* FC_PAD */
                                            0x5b,                   /* FC_END */
/* 580 */
                                            0x11, 0x0,    /* FC_RP */
/* 582 */     NdrFcShort( 0xfffffffd ),     /* Offset= -44 (538) */
/* 584 */
                                            0x2f,                   /* FC_IP */
                                            0x5a,                   /*
FC_CONSTANT_IID */
/* 586 */     NdrFcLong( 0x2f ),            /* 47 */
/* 590 */     NdrFcShort( 0x0 ),            /* 0 */
/* 592 */     NdrFcShort( 0x0 ),            /* 0 */
/* 594 */     0xc0,                         /* 192 */
                                            0x0,                    /* 0 */
/* 596 */     0x0,                          /* 0 */
                                            0x0,                    /* 0 */
/* 598 */     0x0,                          /* 0 */
                                            0x0,                    /* 0 */
/* 600 */     0x0,                          /* 0 */
                                            0x46,                   /* 70 */
/* 602 */
                                            0x1b,                   /* FC_CARRAY
*/
                                            0x0,                    /* 0 */
/* 604 */     NdrFcShort( 0x1 ),            /* 1 */
/* 606 */     0x19,                         /* Corr desc:  field pointer, FC_ULONG
*/
                                            0x0,                    /*  */
/* 608 */     NdrFcShort( 0x4 ),            /* 4 */
/* 610 */     0x1,                          /* FC_BYTE */
                                            0x5b,                   /* FC_END */
/* 612 */
                                            0x1a,                   /*
FC_BOGUS_STRUCT */
                                            0x3,                    /* 3 */
/* 614 */     NdrFcShort( 0x10 ),           /* 16 */
/* 616 */     NdrFcShort( 0x0 ),            /* 0 */
/* 618 */     NdrFcShort( 0xa ),            /* Offset= 10 (628) */
/* 620 */     0x8,                          /* FC_LONG */
                                            0x8,                    /* FC_LONG */
/* 622 */     0x4c,                         /* FC_EMBEDDED_COMPLEX */
                                            0x0,                    /* 0 */
/* 624 */     NdrFcShort( 0xfffffffd8 ),    /* Offset= -40 (584) */
/* 626 */     0x36,                         /* FC_POINTER */
                                            0x5b,                   /* FC_END */
/* 628 */
                                            0x12, 0x0,    /* FC_UP */
/* 630 */     NdrFcShort( 0xffffffe4 ),     /* Offset= -28 (602) */
/* 632 */
                                            0x1b,                   /* FC_CARRAY
*/
                                            0x3,                    /* 3 */
/* 634 */     NdrFcShort( 0x4 ),            /* 4 */
/* 636 */     0x19,                         /* Corr desc:  field pointer, FC_ULONG
*/
                                            0x0,                    /*  */
/* 638 */     NdrFcShort( 0x0 ),            /* 0 */
/* 640 */
                                            0x4b,                   /* FC_PP */
                                            0x5c,                   /* FC_PAD */
/* 642 */
                                            0x48,                   /*
FC_VARIABLE_REPEAT */
                                            0x49,                   /*
FC_FIXED_OFFSET */
/* 644 */     NdrFcShort( 0x4 ),            /* 4 */
/* 646 */     NdrFcShort( 0x0 ),            /* 0 */
/* 648 */     NdrFcShort( 0x1 ),            /* 1 */
/* 650 */     NdrFcShort( 0x0 ),            /* 0 */
/* 652 */     NdrFcShort( 0x0 ),            /* 0 */
/* 654 */     0x12, 0x0,    /* FC_UP */
/* 656 */     NdrFcShort( 0xfffffffd4 ),    /* Offset= -44 (612) */
/* 658 */
                                            0x5b,                   /* FC_END */
                                            0x8,                    /* FC_LONG */
/* 660 */     0x5c,                         /* FC_PAD */
                                            0x5b,                   /* FC_END */
/* 662 */
                                            0x1a,                   /*
FC_BOGUS_STRUCT */
                                            0x3,                    /* 3 */
/* 664 */     NdrFcShort( 0x8 ),            /* 8 */
/* 666 */     NdrFcShort( 0x0 ),            /* 0 */
/* 668 */     NdrFcShort( 0x6 ),            /* Offset= 6 (674) */
/* 670 */     0x8,                          /* FC_LONG */
                                            0x36,                   /* FC_POINTER
*/
/* 672 */     0x5c,                         /* FC_PAD */
                                            0x5b,                   /* FC_END */
/* 674 */
                                            0x11, 0x0,    /* FC_RP */
/* 676 */     NdrFcShort( 0xfffffffd4 ),    /* Offset= -44 (632) */
/* 678 */
                                            0x1d,                   /*
FC_SMFARRAY */
                                            0x0,                    /* 0 */
/* 680 */     NdrFcShort( 0x8 ),            /* 8 */
/* 682 */     0x1,                          /* FC_BYTE */
                                            0x5b,                   /* FC_END */
/* 684 */
                                            0x15,                   /* FC_STRUCT
*/
                                            0x3,                    /* 3 */
/* 686 */     NdrFcShort( 0x10 ),           /* 16 */
/* 688 */     0x8,                          /* FC_LONG */
                                            0x6,                    /* FC_SHORT
*/
/* 690 */     0x6,                          /* FC_SHORT */
                                            0x4c,                   /*
FC_EMBEDDED_COMPLEX */
/* 692 */     0x0,                          /* 0 */
                                            NdrFcShort( 0xfffffff1 ),     /* Offset= -
15 (678) */
                                            0x5b,                   /* FC_END */
/* 696 */
                                            0x1a,                   /*
FC_BOGUS_STRUCT */
                                            0x3,                    /* 3 */
/* 698 */     NdrFcShort( 0x18 ),           /* 24 */
/* 700 */     NdrFcShort( 0x0 ),            /* 0 */
/* 702 */     NdrFcShort( 0xa ),            /* Offset= 10 (712) */
/* 704 */     0x8,                          /* FC_LONG */
                                            0x36,                   /* FC_POINTER
*/
/* 706 */     0x4c,                         /* FC_EMBEDDED_COMPLEX */
                                            0x0,                    /* 0 */
/* 708 */     NdrFcShort( 0xffffffe8 ),     /* Offset= -24 (684) */
/* 710 */     0x5c,                         /* FC_PAD */
                                            0x5b,                   /* FC_END */
/* 712 */
                                            0x11, 0x0,    /* FC_RP */
/* 714 */     NdrFcShort( 0xffffff0c ),     /* Offset= -244 (470) */
/* 716 */
                                            0x1b,                   /* FC_CARRAY
*/
                                            0x0,                    /* 0 */
/* 718 */     NdrFcShort( 0x1 ),            /* 1 */
/* 720 */     0x19,                         /* Corr desc:  field pointer, FC_ULONG
*/
                                            0x0,                    /*  */
/* 722 */     NdrFcShort( 0x0 ),            /* 0 */
/* 724 */     0x1,                          /* FC_BYTE */
                                            0x5b,                   /* FC_END */
/* 726 */
                                            0x16,                   /* FC_PSTRUCT
*/
                                            0x3,                    /* 3 */
/* 728 */     NdrFcShort( 0x8 ),            /* 8 */
/* 730 */
                                            0x4b,                   /* FC_PP */
                                            0x5c,                   /* FC_PAD */
/* 732 */
                                            0x46,                   /*
FC_NO_REPEAT */
                                            0x5c,                   /* FC_PAD */
/* 734 */     NdrFcShort( 0x4 ),            /* 4 */
/* 736 */     NdrFcShort( 0x4 ),            /* 4 */
/* 738 */     0x12, 0x0,    /* FC_UP */
/* 740 */     NdrFcShort( 0xffffffe8 ),     /* Offset= -24 (716) */
/* 742 */
                                            0x5b,                   /* FC_END */
                                            0x8,                    /* FC_LONG */
/* 744 */     0x8,                          /* FC_LONG */
                                            0x5b,                   /* FC_END */
/* 746 */
                                            0x1b,                   /* FC_CARRAY
*/
                                            0x1,                    /* 1 */
/* 748 */     NdrFcShort( 0x2 ),            /* 2 */
/* 750 */     0x19,                         /* Corr desc:  field pointer, FC_ULONG
*/
                                            0x0,                    /*  */
/* 752 */     NdrFcShort( 0x0 ),            /* 0 */
/* 754 */     0x6,                          /* FC_SHORT */
                                            0x5b,                   /* FC_END */
/* 756 */
                                            0x16,                   /* FC_PSTRUCT
*/
                                            0x3,                    /* 3 */
/* 758 */     NdrFcShort( 0x8 ),            /* 8 */
/* 760 */
                                            0x4b,                   /* FC_PP */
                                            0x5c,                   /* FC_PAD */
/* 762 */
                                            0x46,                   /*
FC_NO_REPEAT */
                                            0x5c,                   /* FC_PAD */
/* 764 */     NdrFcShort( 0x4 ),            /* 4 */
/* 766 */     NdrFcShort( 0x4 ),            /* 4 */
/* 768 */     0x12, 0x0,    /* FC_UP */
/* 770 */     NdrFcShort( 0xffffffe8 ),     /* Offset= -24 (746) */
/* 772 */
                                            0x5b,                   /* FC_END */
                                            0x8,                    /* FC_LONG */
/* 774 */     0x8,                          /* FC_LONG */
                                            0x5b,                   /* FC_END */
/* 776 */
                                            0x1b,                   /* FC_CARRAY
*/
                                            0x3,                    /* 3 */
/* 778 */     NdrFcShort( 0x4 ),            /* 4 */
/* 780 */     0x19,                         /* Corr desc:  field pointer, FC_ULONG
*/
                                            0x0,                    /*  */
/* 782 */     NdrFcShort( 0x0 ),            /* 0 */
/* 784 */     0x8,                          /* FC_LONG */
                                            0x5b,                   /* FC_END */
/* 786 */
                                            0x16,                   /* FC_PSTRUCT
*/
                                            0x3,                    /* 3 */
/* 788 */     NdrFcShort( 0x8 ),            /* 8 */
/* 790 */
                                            0x4b,                   /* FC_PP */
                                            0x5c,                   /* FC_PAD */
/* 792 */
                                            0x46,                   /*
FC_NO_REPEAT */
                                            0x5c,                   /* FC_PAD */
/* 794 */     NdrFcShort( 0x4 ),            /* 4 */
/* 796 */     NdrFcShort( 0x4 ),            /* 4 */
/* 798 */     0x12, 0x0,    /* FC_UP */
/* 800 */     NdrFcShort( 0xffffffe8 ),     /* Offset= -24 (776) */
/* 802 */
                                            0x5b,                   /* FC_END */
                                            0x8,                    /* FC_LONG */
/* 804 */     0x8,                          /* FC_LONG */
                                            0x5b,                   /* FC_END */
/* 806 */
                                            0x1b,                   /* FC_CARRAY
*/
                                            0x7,                    /* 7 */
/* 808 */     NdrFcShort( 0x8 ),            /* 8 */
/* 810 */     0x19,                         /* Corr desc:  field pointer, FC_ULONG
*/
                                            0x0,                    /*  */
/* 812 */     NdrFcShort( 0x0 ),            /* 0 */
/* 814 */     0xb,                          /* FC_HYPER */
                                            0x5b,                   /* FC_END */
/* 816 */
                                            0x16,                   /* FC_PSTRUCT
*/
                                            0x3,                    /* 3 */
/* 818 */     NdrFcShort( 0x8 ),            /* 8 */
/* 820 */
                                            0x4b,                   /* FC_PP */
                                            0x5c,                   /* FC_PAD */
/* 822 */
                                            0x46,                   /*
FC_NO_REPEAT */
                                            0x5c,                   /* FC_PAD */
/* 824 */     NdrFcShort( 0x4 ),            /* 4 */
/* 826 */     NdrFcShort( 0x4 ),            /* 4 */
/* 828 */     0x12, 0x0,    /* FC_UP */
/* 830 */     NdrFcShort( 0xffffffe8 ),     /* Offset= -24 (806) */
/* 832 */
                                            0x5b,                   /* FC_END */
                                            0x8,                    /* FC_LONG */
/* 834 */     0x8,                          /* FC_LONG */
                                            0x5b,                   /* FC_END */
/* 836 */
                                            0x15,                   /* FC_STRUCT
*/
                                            0x3,                    /* 3 */
/* 838 */     NdrFcShort( 0x8 ),            /* 8 */
/* 840 */     0x8,                          /* FC_LONG */
                                            0x8,                    /* FC_LONG */
/* 842 */     0x5c,                         /* FC_PAD */
                                            0x5b,                   /* FC_END */
/* 844 */
                                            0x1b,                   /* FC_CARRAY
*/
                                            0x3,                    /* 3 */
/* 846 */     NdrFcShort( 0x8 ),            /* 8 */
/* 848 */     0x7,                          /* Corr desc: FC_USHORT */
                                            0x0,                    /*  */
```

```
/* 850 */     NdrFcShort( 0xffd8 ),      /* -40 */
/* 852 */     0x4c,                      /* FC_EMBEDDED_COMPLEX */
              0x0,                                /* 0 */
/* 854 */     NdrFcShort( 0xffffffee ),  /* Offset= -18 (836) */
/* 856 */     0x5c,                      /* FC_PAD */
              0x5b,                               /* FC_END */
/* 858 */
              0x1a,                               /*
FC_BOGUS_STRUCT */
              0x3,                                /* 3 */
/* 860 */     NdrFcShort( 0x28 ),        /* 40 */
/* 862 */     NdrFcShort( 0xffffffee ),  /* Offset= -18 (844) */
/* 864 */     NdrFcShort( 0x0 ),         /* Offset= 0 (864) */
/* 866 */     0x6,                       /* FC_SHORT */
              0x6,                                /* FC_SHORT
*/
/* 868 */     0x38,                      /* FC_ALIGNM4 */
              0x8,                                /* FC_LONG */
/* 870 */     0x8,                       /* FC_LONG */
              0x4c,                               /*
FC_EMBEDDED_COMPLEX */
/* 872 */     0x0,                       /* 0 */
              NdrFcShort( 0xfffffdf7 ),  /* Offset= -
521 (352) */
              0x5b,                               /* FC_END */
/* 876 */
              0x12, 0x0,    /* FC_UP */
/* 878 */     NdrFcShort( 0xfffffef6 ),  /* Offset= -266 (612) */
/* 880 */
              0x12, 0x8,    /* FC_UP [simple_pointer]
*/
/* 882 */     0x1,                       /* FC_BYTE */
              0x5c,                               /* FC_PAD */
/* 884 */
              0x12, 0x8,    /* FC_UP [simple_pointer]
*/
/* 886 */     0x6,                       /* FC_SHORT */
              0x5c,                               /* FC_PAD */
/* 888 */
              0x12, 0x8,    /* FC_UP [simple_pointer]
*/
/* 890 */     0x8,                       /* FC_LONG */
              0x5c,                               /* FC_PAD */
/* 892 */
              0x12, 0x8,    /* FC_UP [simple_pointer]
*/
/* 894 */     0xa,                       /* FC_FLOAT */
              0x5c,                               /* FC_PAD */
/* 896 */
              0x12, 0x8,    /* FC_UP [simple_pointer]
*/
/* 898 */     0xc,                       /* FC_DOUBLE */
              0x5c,                               /* FC_PAD */
/* 900 */
              0x12, 0x0,    /* FC_UP */
/* 902 */     NdrFcShort( 0xfffffd90 ),  /* Offset= -624 (278) */
/* 904 */
              0x12, 0x10,   /* FC_UP [pointer_deref]
*/
/* 906 */     NdrFcShort( 0xfffffd92 ),  /* Offset= -622 (284) */
/* 908 */
              0x12, 0x10,   /* FC_UP [pointer_deref]
*/
/* 910 */     NdrFcShort( 0xfffffda6 ),  /* Offset= -602 (308) */
/* 912 */
              0x12, 0x10,   /* FC_UP [pointer_deref]
*/
/* 914 */     NdrFcShort( 0xfffffdb4 ),  /* Offset= -588 (326) */
/* 916 */
              0x12, 0x10,   /* FC_UP [pointer_deref]
*/
/* 918 */     NdrFcShort( 0xfffffdc2 ),  /* Offset= -574 (344) */
/* 920 */
              0x12, 0x10,   /* FC_UP [pointer_deref]
*/
/* 922 */     NdrFcShort( 0x2 ),         /* Offset= 2 (924) */
/* 924 */
              0x12, 0x0,    /* FC_UP */
/* 926 */     NdrFcShort( 0x16 ),        /* Offset= 22 (948) */
/* 928 */
              0x15,                               /* FC_STRUCT
*/
              0x7,                                /* 7 */
/* 930 */     NdrFcShort( 0x10 ),        /* 16 */
/* 932 */     0x6,                       /* FC_SHORT */
              0x1,                                /* FC_BYTE */
/* 934 */     0x1,                       /* FC_BYTE */
              0x38,                               /* FC_ALIGNM4
*/
/* 936 */     0x8,                       /* FC_LONG */
              0x39,                               /* FC_ALIGNM8
*/
/* 938 */     0xb,                       /* FC_HYPER */
              0x5b,                               /* FC_END */
/* 940 */
              0x12, 0x0,    /* FC_UP */
/* 942 */     NdrFcShort( 0xfffffff2 ),  /* Offset= -14 (928) */
/* 944 */
              0x12, 0x8,    /* FC_UP [simple_pointer]
*/
/* 946 */     0x2,                       /* FC_CHAR */
              0x5c,                               /* FC_PAD */
/* 948 */
              0x1a,                               /*
FC_BOGUS_STRUCT */
              0x7,                                /* 7 */
/* 950 */     NdrFcShort( 0x20 ),        /* 32 */
/* 952 */     NdrFcShort( 0x0 ),         /* 0 */
/* 954 */     NdrFcShort( 0x0 ),         /* Offset= 0 (954) */
/* 956 */     0x8,                       /* FC_LONG */
              0x8,                                /* FC_LONG */
/* 958 */     0x6,                       /* FC_SHORT */
              0x6,                                /* FC_SHORT
*/
/* 960 */     0x6,                       /* FC_SHORT */
              0x6,                                /* FC_SHORT
*/
/* 962 */     0x4c,                      /* FC_EMBEDDED_COMPLEX */
              0x0,                                /* 0 */
/* 964 */     NdrFcShort( 0xfffffc42 ),  /* Offset= -958 (6) */
/* 966 */     0x5c,                      /* FC_PAD */
              0x5b,                               /* FC_END */
/* 968 */     0xb4,                      /* FC_USER_MARSHAL */
              0x83,                               /* 131 */
/* 970 */     NdrFcShort( 0x0 ),         /* 0 */
/* 972 */     NdrFcShort( 0x10 ),        /* 16 */
/* 974 */     NdrFcShort( 0x0 ),         /* 0 */
/* 976 */     NdrFcShort( 0xfffffc32 ),  /* Offset= -974 (2) */
/* 978 */
              0x11, 0x4,    /* FC_RP
[alloced_on_stack] */
              NdrFcShort( 0x6 ),         /* Offset= 6 (986) */
/* 982 */
              0x13, 0x0,    /* FC_OP */
/* 984 */     NdrFcShort( 0xffffffdc ),  /* Offset= -36 (948) */
/* 986 */     0xb4,                      /* FC_USER_MARSHAL */
              0x83,                               /* 131 */
/* 988 */     NdrFcShort( 0x0 ),         /* 0 */
/* 990 */     NdrFcShort( 0x10 ),        /* 16 */
/* 992 */     NdrFcShort( 0x0 ),         /* 0 */
/* 994 */     NdrFcShort( 0xfffffff4 ),  /* Offset= -12 (982) */
              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 Thu Dec 13 23:13:08 2001
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext,
robust
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

#if defined(_M_IA64) || defined(_M_AXP64)
#define USE_STUBLESS_PROXY


/* verify that the <rpcproxy.h> version is high enough to compile this file*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 475
#endif


#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of <rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__


#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE   979
#define PROC_FORMAT_STRING_SIZE   253
#define TRANSMIT_AS_TABLE_SIZE    0
#define WIRE_MARSHAL_TABLE_SIZE   1

typedef struct _MIDL_TYPE_FORMAT_STRING
    {
    short          Pad;
    unsigned char  Format[ TYPE_FORMAT_STRING_SIZE ];
    } MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
    {
    short          Pad;
    unsigned char  Format[ PROC_FORMAT_STRING_SIZE ];
    } MIDL_PROC_FORMAT_STRING;


extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;


/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,
   GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */


/* Object interface: IUnknown, ver. 0.0,
   GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */


/* Object interface: ITPCC, ver. 0.0,
   GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */


extern const MIDL_STUB_DESC Object_StubDesc;


extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[] =
    {
    0,
    44,
    88,
    132,
    176,
    220
    };

static const MIDL_SERVER_INFO ITPCC_ServerInfo =
    {
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0
    };

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo =
    {
    &Object_StubDesc,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0
    };

CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    (void *)-1 /* ITPCC::NewOrder */ ,
    (void *)-1 /* ITPCC::Payment */ ,
    (void *)-1 /* ITPCC::Delivery */ ,
    (void *)-1 /* ITPCC::StockLevel */ ,
    (void *)-1 /* ITPCC::OrderStatus */ ,
    (void *)-1 /* ITPCC::CallSetComplete */
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,
    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

extern const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

static const MIDL_STUB_DESC Object_StubDesc =
    {
```

```
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x50002, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.3.280 */
    0,
    UserMarshalRoutines,
    0,  /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0,  /* Reserved3 */
    0,  /* Reserved4 */
    0   /* Reserved5 */
    };

#pragma data_seg(".rdata")

static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
        {

            {
            VARIANT_UserSize
            ,VARIANT_UserMarshal
            ,VARIANT_UserUnmarshal
            ,VARIANT_UserFree
            }

        };


#if !defined(__RPC_WIN64__)
#error  Invalid build platform for this stub.
#endif

static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
    {
        0,
        {

            /* Procedure NewOrder */

                                        0x33,                       /*
FC_AUTO_HANDLE */
                                        0x6c,                       /* Old Flags:
object, Oi2 */
/*  2 */    NdrFcLong( 0x0 ),           /* 0 */
/*  6 */    NdrFcShort( 0x3 ),          /* 3 */
#ifndef _ALPHA_
/*  8 */    NdrFcShort( 0x38 ),         /* ia64 Stack size/offset = 56 */
#else
                                        NdrFcShort( 0x30 ),         /* axp64
Stack size/offset = 48 */
#endif
/* 10 */    NdrFcShort( 0x0 ),          /* 0 */
/* 12 */    NdrFcShort( 0x8 ),          /* 8 */
/* 14 */    0x47,                       /* Oi2 Flags:  srv must size, clt must
size, has return, has ext, */
                                        0x3,                        /* 3 */
/* 16 */    0xa,                        /* 10 */
                                        0x7,                        /* Ext Flags:
new corr desc, clt corr check, srv corr check, */
/* 18 */    NdrFcShort( 0x20 ),         /* 32 */
/* 20 */    NdrFcShort( 0x20 ),         /* 32 */
/* 22 */    NdrFcShort( 0x0 ),          /* 0 */
/* 24 */    NdrFcShort( 0x0 ),          /* 0 */

            /* Parameter txn_in */

/* 26 */    NdrFcShort( 0x8b ),         /* Flags:  must size, must free, in, by
val, */
#ifndef _ALPHA_
/* 28 */    NdrFcShort( 0x10 ),         /* ia64 Stack size/offset = 16 */
#else
                                        NdrFcShort( 0x8 ),          /* axp64
Stack size/offset = 8 */
#endif
/* 30 */    NdrFcShort( 0x3b6 ),        /* Type Offset=950 */

            /* Parameter txn_out */

/* 32 */    NdrFcShort( 0x6113 ),       /* Flags:  must size, must free, out,
simple ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 34 */    NdrFcShort( 0x28 ),         /* ia64 Stack size/offset = 40 */
#else
                                        NdrFcShort( 0x20 ),         /* axp64
Stack size/offset = 32 */
#endif
/* 36 */    NdrFcShort( 0x3c8 ),        /* Type Offset=968 */

            /* Return value */

/* 38 */    NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 40 */    NdrFcShort( 0x30 ),         /* ia64 Stack size/offset = 48 */
#else
                                        NdrFcShort( 0x28 ),         /* axp64
Stack size/offset = 40 */
#endif
/* 42 */    0x8,                        /* FC_LONG */
                                        0x0,                        /* 0 */

            /* Procedure Payment */

/* 44 */    0x33,                       /* FC_AUTO_HANDLE */
                                        0x6c,                       /* Old Flags:
object, Oi2 */
/* 46 */    NdrFcLong( 0x0 ),           /* 0 */
/* 50 */    NdrFcShort( 0x4 ),          /* 4 */
#ifndef _ALPHA_
/* 52 */    NdrFcShort( 0x38 ),         /* ia64 Stack size/offset = 56 */
#else
                                        NdrFcShort( 0x30 ),         /* axp64
Stack size/offset = 48 */
#endif
/* 54 */    NdrFcShort( 0x0 ),          /* 0 */
/* 56 */    NdrFcShort( 0x8 ),          /* 8 */
/* 58 */    0x47,                       /* Oi2 Flags:  srv must size, clt must
size, has return, has ext, */
                                        0x3,                        /* 3 */
/* 60 */    0xa,                        /* 10 */
                                        0x7,                        /* Ext Flags:
new corr desc, clt corr check, srv corr check, */
/* 62 */    NdrFcShort( 0x20 ),         /* 32 */
/* 64 */    NdrFcShort( 0x20 ),         /* 32 */
/* 66 */    NdrFcShort( 0x0 ),          /* 0 */
/* 68 */    NdrFcShort( 0x0 ),          /* 0 */

            /* Parameter txn_in */

/* 70 */    NdrFcShort( 0x8b ),         /* Flags:  must size, must free, in, by
val, */
#ifndef _ALPHA_
/* 72 */    NdrFcShort( 0x10 ),         /* ia64 Stack size/offset = 16 */
#else
                                        NdrFcShort( 0x8 ),          /* axp64
Stack size/offset = 8 */
#endif
/* 74 */    NdrFcShort( 0x3b6 ),        /* Type Offset=950 */

            /* Parameter txn_out */

/* 76 */    NdrFcShort( 0x6113 ),       /* Flags:  must size, must free, out,
simple ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 78 */    NdrFcShort( 0x28 ),         /* ia64 Stack size/offset = 40 */
#else
                                        NdrFcShort( 0x20 ),         /* axp64
Stack size/offset = 32 */
#endif
/* 80 */    NdrFcShort( 0x3c8 ),        /* Type Offset=968 */

            /* Return value */

/* 82 */    NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 84 */    NdrFcShort( 0x30 ),         /* ia64 Stack size/offset = 48 */
#else
                                        NdrFcShort( 0x28 ),         /* axp64
Stack size/offset = 40 */
#endif
/* 86 */    0x8,                        /* FC_LONG */
                                        0x0,                        /* 0 */

            /* Procedure Delivery */

/* 88 */    0x33,                       /* FC_AUTO_HANDLE */
                                        0x6c,                       /* Old Flags:
object, Oi2 */
/* 90 */    NdrFcLong( 0x0 ),           /* 0 */
/* 94 */    NdrFcShort( 0x5 ),          /* 5 */
#ifndef _ALPHA_
/* 96 */    NdrFcShort( 0x38 ),         /* ia64 Stack size/offset = 56 */
#else
                                        NdrFcShort( 0x30 ),         /* axp64
Stack size/offset = 48 */
#endif
/* 98 */    NdrFcShort( 0x0 ),          /* 0 */
/* 100 */   NdrFcShort( 0x8 ),          /* 8 */
/* 102 */   0x47,                       /* Oi2 Flags:  srv must size, clt must
size, has return, has ext, */
                                        0x3,                        /* 3 */
/* 104 */   0xa,                        /* 10 */
                                        0x7,                        /* Ext Flags:
new corr desc, clt corr check, srv corr check, */
/* 106 */   NdrFcShort( 0x20 ),         /* 32 */
/* 108 */   NdrFcShort( 0x20 ),         /* 32 */
/* 110 */   NdrFcShort( 0x0 ),          /* 0 */
/* 112 */   NdrFcShort( 0x0 ),          /* 0 */

            /* Parameter txn_in */

/* 114 */   NdrFcShort( 0x8b ),         /* Flags:  must size, must free, in, by
val, */
#ifndef _ALPHA_
/* 116 */   NdrFcShort( 0x10 ),         /* ia64 Stack size/offset = 16 */
#else
                                        NdrFcShort( 0x8 ),          /* axp64
Stack size/offset = 8 */
#endif
/* 118 */   NdrFcShort( 0x3b6 ),        /* Type Offset=950 */

            /* Parameter txn_out */

/* 120 */   NdrFcShort( 0x6113 ),       /* Flags:  must size, must free, out,
simple ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 122 */   NdrFcShort( 0x28 ),         /* ia64 Stack size/offset = 40 */
#else
                                        NdrFcShort( 0x20 ),         /* axp64
Stack size/offset = 32 */
#endif
/* 124 */   NdrFcShort( 0x3c8 ),        /* Type Offset=968 */

            /* Return value */

/* 126 */   NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 128 */   NdrFcShort( 0x30 ),         /* ia64 Stack size/offset = 48 */
#else
                                        NdrFcShort( 0x28 ),         /* axp64
Stack size/offset = 40 */
#endif
/* 130 */   0x8,                        /* FC_LONG */
                                        0x0,                        /* 0 */

            /* Procedure StockLevel */

/* 132 */   0x33,                       /* FC_AUTO_HANDLE */
                                        0x6c,                       /* Old Flags:
object, Oi2 */
/* 134 */   NdrFcLong( 0x0 ),           /* 0 */
/* 138 */   NdrFcShort( 0x6 ),          /* 6 */
#ifndef _ALPHA_
/* 140 */   NdrFcShort( 0x38 ),         /* ia64 Stack size/offset = 56 */
#else
                                        NdrFcShort( 0x30 ),         /* axp64
Stack size/offset = 48 */
#endif
/* 142 */   NdrFcShort( 0x0 ),          /* 0 */
/* 144 */   NdrFcShort( 0x8 ),          /* 8 */
/* 146 */   0x47,                       /* 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( 0x20 ),         /* 32 */
/* 152 */   NdrFcShort( 0x20 ),         /* 32 */
/* 154 */   NdrFcShort( 0x0 ),          /* 0 */
/* 156 */   NdrFcShort( 0x0 ),          /* 0 */

            /* Parameter txn_in */

/* 158 */   NdrFcShort( 0x8b ),         /* Flags:  must size, must free, in, by
val, */
#ifndef _ALPHA_
/* 160 */   NdrFcShort( 0x10 ),         /* ia64 Stack size/offset = 16 */
#else
                                        NdrFcShort( 0x8 ),          /* axp64
Stack size/offset = 8 */
#endif
/* 162 */   NdrFcShort( 0x3b6 ),        /* Type Offset=950 */

            /* Parameter txn_out */

/* 164 */   NdrFcShort( 0x6113 ),       /* Flags:  must size, must free, out,
simple ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 166 */   NdrFcShort( 0x28 ),         /* ia64 Stack size/offset = 40 */
#else
                                        NdrFcShort( 0x20 ),         /* axp64
Stack size/offset = 32 */
#endif
/* 168 */   NdrFcShort( 0x3c8 ),        /* Type Offset=968 */

            /* Return value */

/* 170 */   NdrFcShort( 0x70 ),         /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 172 */   NdrFcShort( 0x30 ),         /* ia64 Stack size/offset = 48 */
#else
                                        NdrFcShort( 0x28 ),         /* axp64
Stack size/offset = 40 */
#endif
/* 174 */   0x8,                        /* FC_LONG */
                                        0x0,                        /* 0 */

            /* Procedure OrderStatus */

/* 176 */   0x33,                       /* FC_AUTO_HANDLE */
                                        0x6c,                       /* Old Flags:
object, Oi2 */
/* 178 */   NdrFcLong( 0x0 ),           /* 0 */
/* 182 */   NdrFcShort( 0x7 ),          /* 7 */
#ifndef _ALPHA_
/* 184 */   NdrFcShort( 0x38 ),         /* ia64 Stack size/offset = 56 */
#else
                                        NdrFcShort( 0x30 ),         /* axp64
Stack size/offset = 48 */
#endif
/* 186 */   NdrFcShort( 0x0 ),          /* 0 */
/* 188 */   NdrFcShort( 0x8 ),          /* 8 */
/* 190 */   0x47,                       /* Oi2 Flags:  srv must size, clt must
size, has return, has ext, */
                                        0x3,                        /* 3 */
/* 192 */   0xa,                        /* 10 */
                                        0x7,                        /* Ext Flags:
new corr desc, clt corr check, srv corr check, */
/* 194 */   NdrFcShort( 0x20 ),         /* 32 */
/* 196 */   NdrFcShort( 0x20 ),         /* 32 */
/* 198 */   NdrFcShort( 0x0 ),          /* 0 */
/* 200 */   NdrFcShort( 0x0 ),          /* 0 */

            /* Parameter txn_in */

/* 202 */   NdrFcShort( 0x8b ),         /* Flags:  must size, must free, in, by
val, */
#ifndef _ALPHA_
/* 204 */   NdrFcShort( 0x10 ),         /* ia64 Stack size/offset = 16 */
#else
                                        NdrFcShort( 0x8 ),          /* axp64
Stack size/offset = 8 */
#endif
/* 206 */   NdrFcShort( 0x3b6 ),        /* Type Offset=950 */

            /* Parameter txn_out */
```

```
/* 208 */    NdrFcShort( 0x6113 ),      /* Flags:  must size, must free, out,
simple_ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 210 */    NdrFcShort( 0x28 ),        /* ia64 Stack size/offset = 40 */
#else
                                        NdrFcShort( 0x20 ),        /* axp64
Stack size/offset = 32 */
#endif
/* 212 */    NdrFcShort( 0x3c8 ),       /* Type Offset=968 */

            /* Return value */

/* 214 */    NdrFcShort( 0x70 ),        /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 216 */    NdrFcShort( 0x30 ),        /* ia64 Stack size/offset = 48 */
#else
                                        NdrFcShort( 0x28 ),        /* axp64
Stack size/offset = 40 */
#endif
/* 218 */    0x8,                       /* FC_LONG */
                                        0x0,                       /* 0 */

            /* Procedure CallSetComplete */

/* 220 */    0x33,                      /* FC_AUTO_HANDLE */
                                        0x6c,                      /* Old Flags:
object, 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 */
/* 236 */    0xa,                       /* 10 */
                                        0x1,                       /* Ext Flags:
new corr desc, */
/* 238 */    NdrFcShort( 0x0 ),         /* 0 */
/* 240 */    NdrFcShort( 0x0 ),         /* 0 */
/* 242 */    NdrFcShort( 0x0 ),         /* 0 */
/* 244 */    NdrFcShort( 0x0 ),         /* 0 */

            /* Return value */

/* 246 */    NdrFcShort( 0x70 ),        /* Flags:  out, return, base type, */
/* 248 */    NdrFcShort( 0x8 ),         /* ia64, axp64 Stack size/offset = 8 */
/* 250 */    0x8,                       /* FC_LONG */
                                        0x0,                       /* 0 */

                                        0x0

        }
    };

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
    {
        0,
        {
                                        NdrFcShort( 0x0 ),         /* 0 */
/*  2 */
                                        0x12, 0x0,   /* FC_UP */
/*  4 */    NdrFcShort( 0x39e ),       /* Offset= 926 (930) */
/*  6 */
                                        0x2b,                      /*
FC_NON_ENCAPSULATED_UNION */
                                        0x9,                       /* FC_ULONG
*/
/*  8 */    0x7,                       /* Corr desc: FC_USHORT */
                                        0x0,                       /*  */
/* 10 */    NdrFcShort( 0xfff8 ),      /* -8 */
/* 12 */    NdrFcShort( 0x1 ),         /* Corr flags:  early, */
/* 14 */    NdrFcShort( 0x2 ),         /* Offset= 2 (16) */
/* 16 */    NdrFcShort( 0x10 ),        /* 16 */
/* 18 */    NdrFcShort( 0x2b ),        /* 43 */
/* 20 */    NdrFcLong( 0x3 ),          /* 3 */
/* 24 */    NdrFcShort( 0x8008 ),      /* Simple arm type: FC_LONG */
/* 26 */    NdrFcLong( 0x11 ),         /* 17 */
/* 30 */    NdrFcShort( 0x8001 ),      /* Simple arm type: FC_BYTE */
/* 32 */    NdrFcLong( 0x2 ),          /* 2 */
/* 36 */    NdrFcShort( 0x8006 ),      /* Simple arm type: FC_SHORT */
/* 38 */    NdrFcLong( 0x4 ),          /* 4 */
/* 42 */    NdrFcShort( 0x800a ),      /* Simple arm type: FC_FLOAT */
/* 44 */    NdrFcLong( 0x5 ),          /* 5 */
/* 48 */    NdrFcShort( 0x800c ),      /* Simple arm type: FC_DOUBLE */
/* 50 */    NdrFcLong( 0xb ),          /* 11 */
/* 54 */    NdrFcShort( 0x8006 ),      /* Simple arm type: FC_SHORT */
/* 56 */    NdrFcLong( 0xa ),          /* 10 */
/* 60 */    NdrFcShort( 0x8008 ),      /* Simple arm type: FC_LONG */
/* 62 */    NdrFcLong( 0x6 ),          /* 6 */
/* 66 */    NdrFcShort( 0xd6 ),        /* Offset= 214 (280) */
/* 68 */    NdrFcLong( 0x7 ),          /* 7 */
/* 72 */    NdrFcShort( 0x800c ),      /* Simple arm type: FC_DOUBLE */
/* 74 */    NdrFcLong( 0x8 ),          /* 8 */
/* 78 */    NdrFcShort( 0xd0 ),        /* Offset= 208 (286) */
/* 80 */    NdrFcLong( 0xd ),          /* 13 */
/* 84 */    NdrFcShort( 0xe4 ),        /* Offset= 228 (312) */
/* 86 */    NdrFcLong( 0x9 ),          /* 9 */
/* 90 */    NdrFcShort( 0xf0 ),        /* Offset= 240 (330) */
/* 92 */    NdrFcLong( 0x2000 ),       /* 8192 */
/* 96 */    NdrFcShort( 0xfc ),        /* Offset= 252 (348) */
/* 98 */    NdrFcLong( 0x24 ),         /* 36 */
/* 102 */   NdrFcShort( 0x2f4 ),       /* Offset= 756 (858) */
/* 104 */   NdrFcLong( 0x4024 ),       /* 16420 */
/* 108 */   NdrFcShort( 0x2ee ),       /* Offset= 750 (858) */
/* 110 */   NdrFcLong( 0x4011 ),       /* 16401 */
/* 114 */   NdrFcShort( 0x2ec ),       /* Offset= 748 (862) */
/* 116 */   NdrFcLong( 0x4002 ),       /* 16386 */
/* 120 */   NdrFcShort( 0x2ea ),       /* Offset= 746 (866) */
/* 122 */   NdrFcLong( 0x4003 ),       /* 16387 */
/* 126 */   NdrFcShort( 0x2e8 ),       /* Offset= 744 (870) */
/* 128 */   NdrFcLong( 0x4004 ),       /* 16388 */
/* 132 */   NdrFcShort( 0x2e6 ),       /* Offset= 742 (874) */
/* 134 */   NdrFcLong( 0x4005 ),       /* 16389 */
/* 138 */   NdrFcShort( 0x2e4 ),       /* Offset= 740 (878) */
/* 140 */   NdrFcLong( 0x400b ),       /* 16395 */
/* 144 */   NdrFcShort( 0x2d2 ),       /* Offset= 722 (866) */
/* 146 */   NdrFcLong( 0x400a ),       /* 16394 */
/* 150 */   NdrFcShort( 0x2d0 ),       /* Offset= 720 (870) */
/* 152 */   NdrFcLong( 0x4006 ),       /* 16390 */
/* 156 */   NdrFcShort( 0x2d6 ),       /* Offset= 726 (882) */
/* 158 */   NdrFcLong( 0x4007 ),       /* 16391 */
/* 162 */   NdrFcShort( 0x2cc ),       /* Offset= 716 (878) */
/* 164 */   NdrFcLong( 0x4008 ),       /* 16392 */
/* 168 */   NdrFcShort( 0x2ce ),       /* Offset= 718 (886) */
/* 170 */   NdrFcLong( 0x400d ),       /* 16397 */
/* 174 */   NdrFcShort( 0x2cc ),       /* Offset= 716 (890) */
/* 176 */   NdrFcLong( 0x4009 ),       /* 16393 */
/* 180 */   NdrFcShort( 0x2ca ),       /* Offset= 714 (894) */
/* 182 */   NdrFcLong( 0x6000 ),       /* 24576 */
/* 186 */   NdrFcShort( 0x2c8 ),       /* Offset= 712 (898) */
/* 188 */   NdrFcLong( 0x400c ),       /* 16396 */
/* 192 */   NdrFcShort( 0x2c6 ),       /* Offset= 710 (902) */
/* 194 */   NdrFcLong( 0x10 ),         /* 16 */
/* 198 */   NdrFcShort( 0x8002 ),      /* Simple arm type: FC_CHAR */
/* 200 */   NdrFcLong( 0x12 ),         /* 18 */
/* 204 */   NdrFcShort( 0x8006 ),      /* Simple arm type: FC_SHORT */
/* 206 */   NdrFcLong( 0x13 ),         /* 19 */
/* 210 */   NdrFcShort( 0x8008 ),      /* Simple arm type: FC_LONG */
/* 212 */   NdrFcLong( 0x16 ),         /* 22 */
/* 216 */   NdrFcShort( 0x8008 ),      /* Simple arm type: FC_LONG */
/* 218 */   NdrFcLong( 0x17 ),         /* 23 */
/* 222 */   NdrFcShort( 0x8008 ),      /* Simple arm type: FC_LONG */
/* 224 */   NdrFcLong( 0xe ),          /* 14 */
/* 228 */   NdrFcShort( 0x2aa ),       /* Offset= 682 (910) */
/* 230 */   NdrFcLong( 0x400e ),       /* 16398 */
/* 234 */   NdrFcShort( 0x2b0 ),       /* Offset= 688 (922) */
/* 236 */   NdrFcLong( 0x4010 ),       /* 16400 */
/* 240 */   NdrFcShort( 0x2ae ),       /* Offset= 686 (926) */
/* 242 */   NdrFcLong( 0x4012 ),       /* 16402 */
/* 246 */   NdrFcShort( 0x26c ),       /* Offset= 620 (866) */
/* 248 */   NdrFcLong( 0x4013 ),       /* 16403 */
/* 252 */   NdrFcShort( 0x26a ),       /* Offset= 618 (870) */
/* 254 */   NdrFcLong( 0x4016 ),       /* 16406 */
/* 258 */   NdrFcShort( 0x264 ),       /* Offset= 612 (870) */
/* 260 */   NdrFcLong( 0x4017 ),       /* 16407 */
/* 264 */   NdrFcShort( 0x25e ),       /* Offset= 606 (870) */
/* 266 */   NdrFcLong( 0x0 ),          /* 0 */
/* 270 */   NdrFcShort( 0x0 ),         /* Offset= 0 (270) */
/* 272 */   NdrFcLong( 0x1 ),          /* 1 */
/* 276 */   NdrFcShort( 0x0 ),         /* Offset= 0 (276) */
/* 278 */   NdrFcShort( 0xffffffff ),  /* Offset= -1 (277) */
/* 280 */
                                        0x15,                      /* FC_STRUCT
*/
                                        0x7,                       /* 7 */
/* 282 */   NdrFcShort( 0x8 ),         /* 8 */
/* 284 */   0xb,                       /* FC_HYPER */
                                        0x5b,                      /* FC_END */
/* 286 */
                                        0x12, 0x0,   /* FC_UP */
/* 288 */   NdrFcShort( 0xe ),         /* Offset= 14 (302) */
/* 290 */
                                        0x1b,                      /* FC_CARRAY
*/
                                        0x1,                       /* 1 */
/* 292 */   NdrFcShort( 0x2 ),         /* 2 */
/* 294 */   0x9,                       /* Corr desc: FC_ULONG */
                                        0x0,                       /*  */
/* 296 */   NdrFcShort( 0xfffc ),      /* -4 */
/* 298 */   NdrFcShort( 0x1 ),         /* Corr flags:  early, */
/* 300 */   0x6,                       /* FC_SHORT */
                                        0x5b,                      /* FC_END */
/* 302 */
                                        0x17,                      /* FC_CSTRUCT
*/
                                        0x3,                       /* 3 */
/* 304 */   NdrFcShort( 0x8 ),         /* 8 */
/* 306 */   NdrFcShort( 0xfffffff0 ),  /* Offset= -16 (290) */
/* 308 */   0x8,                       /* FC_LONG */
                                        0x8,                       /* FC_LONG */
/* 310 */   0x5c,                      /* FC_PAD */
                                        0x5b,                      /* FC_END */
/* 312 */
                                        0x2f,                      /* FC_IP */
                                        0x5a,                      /*
FC_CONSTANT_IID */
/* 314 */   NdrFcLong( 0x0 ),          /* 0 */
/* 318 */   NdrFcShort( 0x0 ),         /* 0 */
/* 320 */   NdrFcShort( 0x0 ),         /* 0 */
/* 322 */   0xc0,                      /* 192 */
                                        0x0,                       /* 0 */
/* 324 */   0x0,                       /* 0 */
                                        0x0,                       /* 0 */
/* 326 */   0x0,                       /* 0 */
                                        0x0,                       /* 0 */
/* 328 */   0x0,                       /* 0 */
                                        0x46,                      /* 70 */
/* 330 */
                                        0x2f,                      /* FC_IP */
                                        0x5a,                      /*
FC_CONSTANT_IID */
/* 332 */   NdrFcLong( 0x20400 ),      /* 132096 */
/* 336 */   NdrFcShort( 0x0 ),         /* 0 */
/* 338 */   NdrFcShort( 0x0 ),         /* 0 */
/* 340 */   0xc0,                      /* 192 */
                                        0x0,                       /* 0 */
/* 342 */   0x0,                       /* 0 */
                                        0x0,                       /* 0 */
/* 344 */   0x0,                       /* 0 */
                                        0x0,                       /* 0 */
/* 346 */   0x0,                       /* 0 */
                                        0x46,                      /* 70 */
/* 348 */
                                        0x12, 0x10,  /* FC_UP [pointer_deref]
*/
/* 350 */    NdrFcShort( 0x2 ),        /* Offset= 2 (352) */
/* 352 */
                                        0x12, 0x0,   /* FC_UP */
/* 354 */    NdrFcShort( 0x1e6 ),      /* Offset= 486 (840) */
/* 356 */
                                        0x2a,                      /*
FC_ENCAPSULATED_UNION */
                                        0x89,                      /* 137 */
/* 358 */    NdrFcShort( 0x20 ),       /* 32 */
/* 360 */    NdrFcShort( 0xa ),        /* 10 */
/* 362 */    NdrFcLong( 0x8 ),         /* 8 */
/* 366 */    NdrFcShort( 0x50 ),       /* Offset= 80 (446) */
/* 368 */    NdrFcLong( 0xd ),         /* 13 */
/* 372 */    NdrFcShort( 0x70 ),       /* Offset= 112 (484) */
/* 374 */    NdrFcLong( 0x9 ),         /* 9 */
/* 378 */    NdrFcShort( 0x90 ),       /* Offset= 144 (522) */
/* 380 */    NdrFcLong( 0xc ),         /* 12 */
/* 384 */    NdrFcShort( 0xb0 ),       /* Offset= 176 (560) */
/* 386 */    NdrFcLong( 0x24 ),        /* 36 */
/* 390 */    NdrFcShort( 0x104 ),      /* Offset= 260 (650) */
/* 392 */    NdrFcLong( 0x800d ),      /* 32781 */
/* 396 */    NdrFcShort( 0x120 ),      /* Offset= 288 (684) */
/* 398 */    NdrFcLong( 0x10 ),        /* 16 */
/* 402 */    NdrFcShort( 0x13a ),      /* Offset= 314 (716) */
/* 404 */    NdrFcLong( 0x2 ),         /* 2 */
/* 408 */    NdrFcShort( 0x150 ),      /* Offset= 336 (744) */
/* 410 */    NdrFcLong( 0x3 ),         /* 3 */
/* 414 */    NdrFcShort( 0x166 ),      /* Offset= 358 (772) */
/* 416 */    NdrFcLong( 0x14 ),        /* 20 */
/* 420 */    NdrFcShort( 0x17c ),      /* Offset= 380 (800) */
/* 422 */    NdrFcShort( 0xffffffff ), /* Offset= -1 (421) */
/* 424 */
                                        0x21,                      /*
FC_BOGUS_ARRAY */
                                        0x3,                       /* 3 */
/* 426 */    NdrFcShort( 0x0 ),        /* 0 */
/* 428 */    0x19,                     /* Corr desc:  field pointer, FC_ULONG
*/
                                        0x0,                       /*  */
/* 430 */    NdrFcShort( 0x0 ),        /* 0 */
/* 432 */    NdrFcShort( 0x1 ),        /* Corr flags:  early, */
/* 434 */    NdrFcLong( 0xffffffff ),  /* -1 */
/* 438 */    NdrFcShort( 0x0 ),        /* Corr flags:  */
/* 440 */
                                        0x12, 0x0,   /* FC_UP */
/* 442 */    NdrFcShort( 0xffffff74 ), /* Offset= -140 (302) */
/* 444 */    0x5c,                     /* FC_PAD */
                                        0x5b,                      /* FC_END */
/* 446 */
                                        0x1a,                      /*
FC_BOGUS_STRUCT */
                                        0x3,                       /* 3 */
/* 448 */    NdrFcShort( 0x10 ),       /* 16 */
/* 450 */    NdrFcShort( 0x0 ),        /* 0 */
/* 452 */    NdrFcShort( 0x6 ),        /* Offset= 6 (458) */
/* 454 */    0x8,                      /* FC_LONG */
                                        0x39,                      /* FC_ALIGNM8
*/
/* 456 */    0x36,                     /* FC_POINTER */
                                        0x5b,                      /* FC_END */
/* 458 */
                                        0x11, 0x0,   /* FC_RP */
/* 460 */    NdrFcShort( 0xffffffdc ), /* Offset= -36 (424) */
/* 462 */
                                        0x21,                      /*
FC_BOGUS_ARRAY */
                                        0x3,                       /* 3 */
/* 464 */    NdrFcShort( 0x0 ),        /* 0 */
/* 466 */    0x19,                     /* Corr desc:  field pointer, FC_ULONG
*/
                                        0x0,                       /*  */
/* 468 */    NdrFcShort( 0x0 ),        /* 0 */
/* 470 */    NdrFcShort( 0x1 ),        /* Corr flags:  early, */
/* 472 */    NdrFcLong( 0xffffffff ),  /* -1 */
/* 476 */    NdrFcShort( 0x0 ),        /* Corr flags:  */
/* 478 */    0x4c,                     /* FC_EMBEDDED_COMPLEX */
                                        0x0,                       /* 0 */
/* 480 */    NdrFcShort( 0xffffff58 ), /* Offset= -168 (312) */
/* 482 */    0x5c,                     /* FC_PAD */
                                        0x5b,                      /* FC_END */
/* 484 */
                                        0x1a,                      /*
FC_BOGUS_STRUCT */
                                        0x3,                       /* 3 */
/* 486 */    NdrFcShort( 0x10 ),       /* 16 */
/* 488 */    NdrFcShort( 0x0 ),        /* 0 */
/* 490 */    NdrFcShort( 0x6 ),        /* Offset= 6 (496) */
/* 492 */    0x8,                      /* FC_LONG */
                                        0x39,                      /* FC_ALIGNM8
*/
/* 494 */    0x36,                     /* FC_POINTER */
                                        0x5b,                      /* FC_END */
/* 496 */
                                        0x11, 0x0,   /* FC_RP */
/* 498 */    NdrFcShort( 0xffffffdc ), /* Offset= -36 (462) */
/* 500 */
                                        0x21,                      /*
FC_BOGUS_ARRAY */
                                        0x3,                       /* 3 */
/* 502 */    NdrFcShort( 0x0 ),        /* 0 */
/* 504 */    0x19,                     /* Corr desc:  field pointer, FC_ULONG
*/
                                        0x0,                       /*  */
/* 506 */    NdrFcShort( 0x0 ),        /* 0 */
/* 508 */    NdrFcShort( 0x1 ),        /* Corr flags:  early, */
/* 510 */    NdrFcLong( 0xffffffff ),  /* -1 */
/* 514 */    NdrFcShort( 0x0 ),        /* Corr flags:  */
/* 516 */    0x4c,                     /* FC_EMBEDDED_COMPLEX */
                                        0x0,                       /* 0 */
/* 518 */    NdrFcShort( 0xffffff44 ), /* Offset= -188 (330) */
/* 520 */    0x5c,                     /* FC_PAD */
                                        0x5b,                      /* FC_END */
```

```
/* 522 */                                   0x1a,                   /*
FC_BOGUS_STRUCT */
                                            0x3,                    /* 3 */
/* 524 */    NdrFcShort( 0x10 ),            /* 16 */
/* 526 */    NdrFcShort( 0x0 ),             /* 0 */
/* 528 */    NdrFcShort( 0x6 ),             /* Offset= 6 (534) */
/* 530 */    0x8,                           /* FC_LONG */
                                            0x39,                   /* FC_ALIGNM8
*/
/* 532 */    0x36,                          /* FC_POINTER */
                                            0x5b,                   /* FC_END */
/* 534 */
                                            0x11, 0x0,   /* FC_RP */
/* 536 */    NdrFcShort( 0xfffffffdc ),     /* Offset= -36 (500) */
/* 538 */
                                            0x21,                   /*
FC_BOGUS_ARRAY */
                                            0x3,                    /* 3 */
/* 540 */    NdrFcShort( 0x0 ),             /* 0 */
/* 542 */    0x19,                          /* Corr desc:  field pointer, FC_ULONG
*/
                                            0x0,                    /*  */
/* 544 */    NdrFcShort( 0x0 ),             /* 0 */
/* 546 */    NdrFcShort( 0x1 ),             /* Corr flags:  early, */
/* 548 */    NdrFcLong( 0xffffffff ),       /* -1 */
/* 552 */    NdrFcShort( 0x0 ),             /* Corr flags:  */
/* 554 */
                                            0x12, 0x0,   /* FC_UP */
/* 556 */    NdrFcShort( 0x176 ),           /* Offset= 374 (930) */
/* 558 */    0x5c,                          /* FC_PAD */
                                            0x5b,                   /* FC_END */
/* 560 */
                                            0x1a,                   /*
FC_BOGUS_STRUCT */
                                            0x3,                    /* 3 */
/* 562 */    NdrFcShort( 0x10 ),            /* 16 */
/* 564 */    NdrFcShort( 0x0 ),             /* 0 */
/* 566 */    NdrFcShort( 0x6 ),             /* Offset= 6 (572) */
/* 568 */    0x8,                           /* FC_LONG */
                                            0x39,                   /* FC_ALIGNM8
*/
/* 570 */    0x36,                          /* FC_POINTER */
                                            0x5b,                   /* FC_END */
/* 572 */
                                            0x11, 0x0,   /* FC_RP */
/* 574 */    NdrFcShort( 0xfffffffdc ),     /* Offset= -36 (538) */
/* 576 */
                                            0x2f,                   /* FC_IP */
                                            0x5a,                   /*
FC_CONSTANT_IID */
/* 578 */    NdrFcLong( 0x2f ),             /* 47 */
/* 582 */    NdrFcShort( 0x0 ),             /* 0 */
/* 584 */    NdrFcShort( 0x0 ),             /* 0 */
/* 586 */    0xc0,                          /* 192 */
                                            0x0,                    /* 0 */
/* 588 */    0x0,                           /* 0 */
                                            0x0,                    /* 0 */
/* 590 */    0x0,                           /* 0 */
                                            0x0,                    /* 0 */
/* 592 */    0x0,                           /* 0 */
                                            0x46,                   /* 70 */
/* 594 */
                                            0x1b,                   /* FC_CARRAY
*/
                                            0x0,                    /* 0 */
/* 596 */    NdrFcShort( 0x1 ),             /* 1 */
/* 598 */    0x19,                          /* Corr desc:  field pointer, FC_ULONG
*/
                                            0x0,                    /*  */
/* 600 */    NdrFcShort( 0x4 ),             /* 4 */
/* 602 */    NdrFcShort( 0x1 ),             /* Corr flags:  early, */
/* 604 */    0x1,                           /* FC_BYTE */
                                            0x5b,                   /* FC_END */
/* 606 */
                                            0x1a,                   /*
FC_BOGUS_STRUCT */
                                            0x3,                    /* 3 */
/* 608 */    NdrFcShort( 0x18 ),            /* 24 */
/* 610 */    NdrFcShort( 0x0 ),             /* 0 */
/* 612 */    NdrFcShort( 0xc ),             /* Offset= 12 (624) */
/* 614 */    0x8,                           /* FC_LONG */
                                            0x8,                    /* FC_LONG */
/* 616 */    0x4c,                          /* FC_EMBEDDED_COMPLEX */
                                            0x0,                    /* 0 */
/* 618 */    NdrFcShort( 0xffffffd6 ),      /* Offset= -42 (576) */
/* 620 */    0x39,                          /* FC_ALIGNM8 */
                                            0x36,                   /* FC_POINTER
*/
/* 622 */    0x5c,                          /* FC_PAD */
                                            0x5b,                   /* FC_END */
/* 624 */
                                            0x12, 0x0,   /* FC_UP */
/* 626 */    NdrFcShort( 0xfffffffe0 ),     /* Offset= -32 (594) */
/* 628 */
                                            0x21,                   /*
FC_BOGUS_ARRAY */
                                            0x3,                    /* 3 */
/* 630 */    NdrFcShort( 0x0 ),             /* 0 */
/* 632 */    0x19,                          /* Corr desc:  field pointer, FC_ULONG
*/
                                            0x0,                    /*  */
/* 634 */    NdrFcShort( 0x0 ),             /* 0 */
/* 636 */    NdrFcShort( 0x1 ),             /* Corr flags:  early, */
/* 638 */    NdrFcLong( 0xffffffff ),       /* -1 */
/* 642 */    NdrFcShort( 0x0 ),             /* Corr flags:  */
/* 644 */
                                            0x12, 0x0,   /* FC_UP */
/* 646 */    NdrFcShort( 0xffffffd8 ),      /* Offset= -40 (606) */
/* 648 */    0x5c,                          /* FC_PAD */
                                            0x5b,                   /* FC_END */
/* 650 */
                                            0x1a,                   /*
FC_BOGUS_STRUCT */
                                            0x3,                    /* 3 */
/* 652 */    NdrFcShort( 0x10 ),            /* 16 */
/* 654 */    NdrFcShort( 0x0 ),             /* 0 */
/* 656 */    NdrFcShort( 0x6 ),             /* Offset= 6 (662) */
/* 658 */    0x8,                           /* FC_LONG */
                                            0x39,                   /* FC_ALIGNM8
*/
/* 660 */    0x36,                          /* FC_POINTER */
                                            0x5b,                   /* FC_END */
/* 662 */
                                            0x11, 0x0,   /* FC_RP */
/* 664 */    NdrFcShort( 0xfffffffdc ),     /* Offset= -36 (628) */
/* 666 */
                                            0x1d,                   /*
FC_SMFARRAY */
                                            0x0,                    /* 0 */
/* 668 */    NdrFcShort( 0x8 ),             /* 8 */
/* 670 */    0x1,                           /* FC_BYTE */
                                            0x5b,                   /* FC_END */
/* 672 */
                                            0x15,                   /* FC_STRUCT
*/
                                            0x3,                    /* 3 */
/* 674 */    NdrFcShort( 0x10 ),            /* 16 */
/* 676 */    0x8,                           /* FC_LONG */
                                            0x6,                    /* FC_SHORT
*/
/* 678 */    0x6,                           /* FC_SHORT */
                                            0x4c,                   /*
FC_EMBEDDED_COMPLEX */
/* 680 */    0x0,                           /* 0 */
                                            NdrFcShort( 0xfffffff1 ),  /* Offset= -
15 (666) */
                                            0x5b,                   /* FC_END */
/* 684 */
                                            0x1a,                   /*
FC_BOGUS_STRUCT */
                                            0x3,                    /* 3 */
/* 686 */    NdrFcShort( 0x20 ),            /* 32 */
/* 688 */    NdrFcShort( 0x0 ),             /* 0 */
/* 690 */    NdrFcShort( 0xa ),             /* Offset= 10 (700) */
/* 692 */    0x8,                           /* FC_LONG */
                                            0x39,                   /* FC_ALIGNM8
*/
/* 694 */    0x36,                          /* FC_POINTER */
                                            0x4c,                   /*
FC_EMBEDDED_COMPLEX */
/* 696 */    0x0,                           /* 0 */
                                            NdrFcShort( 0xfffffffe7 ),  /* Offset= -
25 (672) */
                                            0x5b,                   /* FC_END */
/* 700 */
                                            0x11, 0x0,   /* FC_RP */
/* 702 */    NdrFcShort( 0xffffff10 ),      /* Offset= -240 (462) */
/* 704 */
                                            0x1b,                   /* FC_CARRAY
*/
                                            0x0,                    /* 0 */
/* 706 */    NdrFcShort( 0x1 ),             /* 1 */
/* 708 */    0x19,                          /* Corr desc:  field pointer, FC_ULONG
*/
                                            0x0,                    /*  */
/* 710 */    NdrFcShort( 0x0 ),             /* 0 */
/* 712 */    NdrFcShort( 0x1 ),             /* Corr flags:  early, */
/* 714 */    0x1,                           /* FC_BYTE */
                                            0x5b,                   /* FC_END */
/* 716 */
                                            0x1a,                   /*
FC_BOGUS_STRUCT */
                                            0x3,                    /* 3 */
/* 718 */    NdrFcShort( 0x10 ),            /* 16 */
/* 720 */    NdrFcShort( 0x0 ),             /* 0 */
/* 722 */    NdrFcShort( 0x6 ),             /* Offset= 6 (728) */
/* 724 */    0x8,                           /* FC_LONG */
                                            0x39,                   /* FC_ALIGNM8
*/
/* 726 */    0x36,                          /* FC_POINTER */
                                            0x5b,                   /* FC_END */
/* 728 */
                                            0x12, 0x0,   /* FC_UP */
/* 730 */    NdrFcShort( 0xffffffe6 ),      /* Offset= -26 (704) */
/* 732 */
                                            0x1b,                   /* FC_CARRAY
*/
                                            0x1,                    /* 1 */
/* 734 */    NdrFcShort( 0x2 ),             /* 2 */
/* 736 */    0x19,                          /* Corr desc:  field pointer, FC_ULONG
*/
                                            0x0,                    /*  */
/* 738 */    NdrFcShort( 0x0 ),             /* 0 */
/* 740 */    NdrFcShort( 0x1 ),             /* Corr flags:  early, */
/* 742 */    0x6,                           /* FC_SHORT */
                                            0x5b,                   /* FC_END */
/* 744 */
                                            0x1a,                   /*
FC_BOGUS_STRUCT */
                                            0x3,                    /* 3 */
/* 746 */    NdrFcShort( 0x10 ),            /* 16 */
/* 748 */    NdrFcShort( 0x0 ),             /* 0 */
/* 750 */    NdrFcShort( 0x6 ),             /* Offset= 6 (756) */
/* 752 */    0x8,                           /* FC_LONG */
                                            0x39,                   /* FC_ALIGNM8
*/
/* 754 */    0x36,                          /* FC_POINTER */
                                            0x5b,                   /* FC_END */
/* 756 */
                                            0x12, 0x0,   /* FC_UP */
/* 758 */    NdrFcShort( 0xffffffe6 ),      /* Offset= -26 (732) */
/* 760 */
                                            0x1b,                   /* FC_CARRAY
*/
                                            0x3,                    /* 3 */
/* 762 */    NdrFcShort( 0x4 ),             /* 4 */
/* 764 */    0x19,                          /* Corr desc:  field pointer, FC_ULONG
*/
                                            0x0,                    /*  */
/* 766 */    NdrFcShort( 0x0 ),             /* 0 */
/* 768 */    NdrFcShort( 0x1 ),             /* Corr flags:  early, */
/* 770 */    0x8,                           /* FC_LONG */
                                            0x5b,                   /* FC_END */
/* 772 */
                                            0x1a,                   /*
FC_BOGUS_STRUCT */
                                            0x3,                    /* 3 */
/* 774 */    NdrFcShort( 0x10 ),            /* 16 */
/* 776 */    NdrFcShort( 0x0 ),             /* 0 */
/* 778 */    NdrFcShort( 0x6 ),             /* Offset= 6 (784) */
/* 780 */    0x8,                           /* FC_LONG */
                                            0x39,                   /* FC_ALIGNM8
*/
/* 782 */    0x36,                          /* FC_POINTER */
                                            0x5b,                   /* FC_END */
/* 784 */
                                            0x12, 0x0,   /* FC_UP */
/* 786 */    NdrFcShort( 0xffffffe6 ),      /* Offset= -26 (760) */
/* 788 */
                                            0x1b,                   /* FC_CARRAY
*/
                                            0x7,                    /* 7 */
/* 790 */    NdrFcShort( 0x8 ),             /* 8 */
/* 792 */    0x19,                          /* Corr desc:  field pointer, FC_ULONG
*/
                                            0x0,                    /*  */
/* 794 */    NdrFcShort( 0x0 ),             /* 0 */
/* 796 */    NdrFcShort( 0x1 ),             /* Corr flags:  early, */
/* 798 */    0xb,                           /* FC_HYPER */
                                            0x5b,                   /* FC_END */
/* 800 */
                                            0x1a,                   /*
FC_BOGUS_STRUCT */
                                            0x3,                    /* 3 */
/* 802 */    NdrFcShort( 0x10 ),            /* 16 */
/* 804 */    NdrFcShort( 0x0 ),             /* 0 */
/* 806 */    NdrFcShort( 0x6 ),             /* Offset= 6 (812) */
/* 808 */    0x8,                           /* FC_LONG */
                                            0x39,                   /* FC_ALIGNM8
*/
/* 810 */    0x36,                          /* FC_POINTER */
                                            0x5b,                   /* FC_END */
/* 812 */
                                            0x12, 0x0,   /* FC_UP */
/* 814 */    NdrFcShort( 0xffffffe6 ),      /* Offset= -26 (788) */
/* 816 */
                                            0x15,                   /* FC_STRUCT
*/
                                            0x3,                    /* 3 */
/* 818 */    NdrFcShort( 0x8 ),             /* 8 */
/* 820 */    0x8,                           /* FC_LONG */
                                            0x8,                    /* FC_LONG */
/* 822 */    0x5c,                          /* FC_PAD */
                                            0x5b,                   /* FC_END */
/* 824 */
                                            0x1b,                   /* FC_CARRAY
*/
                                            0x3,                    /* 3 */
/* 826 */    NdrFcShort( 0x8 ),             /* 8 */
/* 828 */    0x7,                           /* Corr desc: FC_USHORT */
                                            0x0,                    /*  */
/* 830 */    NdrFcShort( 0xffc8 ),          /* -56 */
/* 832 */    NdrFcShort( 0x1 ),             /* Corr flags:  early, */
/* 834 */    0x4c,                          /* FC_EMBEDDED_COMPLEX */
                                            0x0,                    /* 0 */
/* 836 */    NdrFcShort( 0xfffffec ),       /* Offset= -20 (816) */
/* 838 */    0x5c,                          /* FC_PAD */
                                            0x5b,                   /* FC_END */
/* 840 */
                                            0x1a,                   /*
FC_BOGUS_STRUCT */
                                            0x3,                    /* 3 */
/* 842 */    NdrFcShort( 0x38 ),            /* 56 */
/* 844 */    NdrFcShort( 0xffffffec ),      /* Offset= -20 (824) */
/* 846 */    NdrFcShort( 0x0 ),             /* Offset= 0 (846) */
/* 848 */    0x6,                           /* FC_SHORT */
                                            0x6,                    /* FC_SHORT
*/
/* 850 */    0x38,                          /* FC_ALIGNM4 */
                                            0x8,                    /* FC_LONG */
/* 852 */    0x8,                           /* FC_LONG */
                                            0x4c,                   /*
FC_EMBEDDED_COMPLEX */
/* 854 */    0x4,                           /* 4 */
                                            NdrFcShort( 0xfffffe0d ),  /* Offset= -
499 (356) */
                                            0x5b,                   /* FC_END */
/* 858 */
                                            0x12, 0x0,   /* FC_UP */
/* 860 */    NdrFcShort( 0xffffff02 ),      /* Offset= -254 (606) */
/* 862 */
                                            0x12, 0x8,   /* FC_UP [simple_pointer]
*/
/* 864 */    0x1,                           /* FC_BYTE */
                                            0x5c,                   /* FC_PAD */
/* 866 */
                                            0x12, 0x8,   /* FC_UP [simple_pointer]
*/
/* 868 */    0x6,                           /* FC_SHORT */
                                            0x5c,                   /* FC_PAD */
/* 870 */
                                            0x12, 0x8,   /* FC_UP [simple_pointer]
*/
/* 872 */    0x8,                           /* FC_LONG */
                                            0x5c,                   /* FC_PAD */
```

```
/* 874 */
                                        0x12, 0x8,   /* FC_UP [simple_pointer]
*/
/* 876 */    0xa,                       /* FC_FLOAT */
                                        0x5c,                       /* FC_PAD */
/* 878 */
                                        0x12, 0x8,   /* FC_UP [simple_pointer]
*/
/* 880 */    0xc,                       /* FC_DOUBLE */
                                        0x5c,                       /* FC_PAD */
/* 882 */
                                        0x12, 0x0,   /* FC_UP */
/* 884 */    NdrFcShort( 0xfffffda4 ),  /* Offset= -604 (280) */
/* 886 */
                                        0x12, 0x10,  /* FC_UP [pointer_deref]
*/
/* 888 */    NdrFcShort( 0xfffffda6 ),  /* Offset= -602 (286) */
/* 890 */
                                        0x12, 0x10,  /* FC_UP [pointer_deref]
*/
/* 892 */    NdrFcShort( 0xfffffdbc ),  /* Offset= -580 (312) */
/* 894 */
                                        0x12, 0x10,  /* FC_UP [pointer_deref]
*/
/* 896 */    NdrFcShort( 0xfffffdca ),  /* Offset= -566 (330) */
/* 898 */
                                        0x12, 0x10,  /* FC_UP [pointer_deref]
*/
/* 900 */    NdrFcShort( 0xfffffdd8 ),  /* Offset= -552 (348) */
/* 902 */
                                        0x12, 0x10,  /* FC_UP [pointer_deref]
*/
/* 904 */    NdrFcShort( 0x2 ),         /* Offset= 2 (906) */
/* 906 */
                                        0x12, 0x0,   /* FC_UP */
/* 908 */    NdrFcShort( 0x16 ),        /* Offset= 22 (930) */
/* 910 */
                                        0x15,                       /* FC_STRUCT
*/
                                        0x7,                        /* 7 */
/* 912 */    NdrFcShort( 0x10 ),        /* 16 */
/* 914 */    0x6,                       /* FC_SHORT */
                                        0x1,                        /* FC_BYTE */
/* 916 */    0x1,                       /* FC_BYTE */
                                        0x38,                       /* FC_ALIGNM4
*/
/* 918 */    0x8,                       /* FC_LONG */
                                        0x39,                       /* FC_ALIGNM8
*/
/* 920 */    0xb,                       /* FC_HYPER */
                                        0x5b,                       /* FC_END */
/* 922 */
                                        0x12, 0x0,   /* FC_UP */
/* 924 */    NdrFcShort( 0xfffffff2 ),  /* Offset= -14 (910) */
/* 926 */
                                        0x12, 0x8,   /* FC_UP [simple_pointer]
*/
/* 928 */    0x2,                       /* FC_CHAR */
                                        0x5c,                       /* FC_PAD */
/* 930 */
                                        0x1a,                       /*
FC_BOGUS_STRUCT */
                                        0x7,                        /* 7 */
/* 932 */    NdrFcShort( 0x20 ),        /* 32 */
/* 934 */    NdrFcShort( 0x0 ),         /* 0 */
/* 936 */    NdrFcShort( 0x0 ),         /* Offset= 0 (936) */
/* 938 */    0x8,                       /* FC_LONG */
                                        0x8,                        /* FC_LONG */
/* 940 */    0x6,                       /* FC_SHORT */
                                        0x6,                        /* FC_SHORT
*/
/* 942 */    0x6,                       /* FC_SHORT */
                                        0x6,                        /* FC_SHORT
*/
/* 944 */    0x4c,                      /* FC_EMBEDDED_COMPLEX */
                                        0x0,                        /* 0 */
/* 946 */    NdrFcShort( 0xfffffc54 ),  /* Offset= -940 (6) */
/* 948 */    0x5c,                      /* FC_PAD */
                                        0x5b,                       /* FC_END */
/* 950 */    0xb4,                      /* FC_USER_MARSHAL */
                                        0x83,                       /* 131 */
/* 952 */    NdrFcShort( 0x0 ),         /* 0 */
/* 954 */    NdrFcShort( 0x18 ),        /* 24 */
/* 956 */    NdrFcShort( 0x0 ),         /* 0 */
/* 958 */    NdrFcShort( 0xfffffc44 ),  /* Offset= -956 (2) */
/* 960 */
                                        0x11, 0x4,   /* FC_RP
[alloced_on_stack] */
/* 962 */    NdrFcShort( 0x6 ),         /* Offset= 6 (968) */
/* 964 */
                                        0x13, 0x0,   /* FC_OP */
/* 966 */    NdrFcShort( 0xffffffdc ),  /* Offset= -36 (930) */
/* 968 */    0xb4,                      /* FC_USER_MARSHAL */
                                        0x83,                       /* 131 */
/* 970 */    NdrFcShort( 0x0 ),         /* 0 */
/* 972 */    NdrFcShort( 0x18 ),        /* 24 */
/* 974 */    NdrFcShort( 0x0 ),         /* 0 */
/* 976 */    NdrFcShort( 0xfffffff4 ),  /* Offset= -12 (964) */

                                        0x0
        }
    };

const CInterfaceProxyVtbl * _tpcc_com_ps_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
    0
};

const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[] =
{
    ( CInterfaceStubVtbl *) &_ITPCCStubVtbl,
    0
};

PCInterfaceName const _tpcc_com_ps_InterfaceNamesList[] =
{
    "ITPCC",
    0
};


#define _tpcc_com_ps_CHECK_IID(n)       IID_GENERIC_CHECK_IID( _tpcc_com_ps,
pIID, n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{

    if(!_tpcc_com_ps_CHECK_IID(0))
        {
        *pIndex = 0;
        return 1;
        }

    return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) & _tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) & _tpcc_com_ps_StubVtblList,
    (const PCInterfaceName * ) & _tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    & _tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0  /* Filler3 */
};


#endif /* defined(_M_IA64) || defined(_M_AXP64)*/
```

Appendix B : Database Design

Build Scripts

setup.cmd

```
::@ECHO OFF

@ECHO
********************************************************************************
@ECHO *
*
@ECHO *  Microsoft TPC-C V3 Benchmark Kit Ver. IA-64
*
@ECHO *
*
@ECHO
********************************************************************************

@rem if not '%PROCESSOR_ARCHITECTURE%' == 'IA64' goto wrongOS

@if '%1'=='' goto usage
@if '%2'=='' goto usage
@if '%3'=='' goto usage
@if not '%4'=='' if not '%4' == 'normal' if not '%4' == 'scale_down' goto usage

:: Cleanup any old .err files
@@if exist logs\*.err del logs\*.err
             >nul

@if  '%3'=='full' goto start
@if  '%3'=='builddb' goto builddb
@if  '%3'=='objects' goto objects
@if  '%3'=='bulkload' goto bulkload
@if  '%3'=='objectsfull' goto objects
@if  '%3'=='bulkloadfull' goto bulkload
@if  '%3'=='backup' goto backup
goto usage

:start
:: Cleanup the logs directory...
@if exist logs\version.log del logs\version.log
             >nul
@if exist logs\db.log del logs\db.log
             >nul
@if exist logs\objects.log del logs\objects.log
             >nul
@if exist logs\objects.log del logs\objects.log
             >nul
@if exist logs\bulkload.log del logs\bulkload.log
             >nul
@if exist logs\backup.log del logs\backup.log
             >nul

@osql -Usa -P -S%1 -Q"select @@version"
             >  logs\version.log
@osql -Usa -P -S%1 -Q"select getdate()"
             >> logs\version.log

:builddb
@@if exist logs\db.log del logs\db.log
             >nul
@ECHO Removing any existing TPCC database and backup devices...
@osql -Usa -P -S%1 -Usa -P < scripts\%2.war\database\removedb.sql  >
logs\db.log
@ECHO Creating Backup Device(s)...
@osql -Usa -P -S%1 -Usa -P < scripts\%2.war\database\backupdev.sql >>
logs\db.log
@if errorlevel 1 goto CREATE_ERROR
@ECHO Building database files and database...
@osql -Usa -P -S%1 -b -Usa -P < scripts\%2.war\database\createdb.sql
             >> logs\db.log
@if errorlevel 1 goto CREATE_ERROR
@ECHO Database build complete.
@if  '%3'=='full' goto objects
goto end

:objects
@if exist logs\objects.log del logs\objects.log
             >nul
@ECHO Creating TPC-C database tables...
@osql -Usa -P -S%1 -b -Usa -P < scripts\%2.war\ddl\tables.sql
             >  logs\objects.log
@if errorlevel 1 goto TABLES_ERROR
@ECHO Creating database objects...
@osql -Usa -P -S%1 -b -Usa -P < scripts\dml\neword.sql
             >> logs\objects.log
@if errorlevel 1 goto NEWORDER_ERROR
@osql -Usa -P -S%1 -b -Usa -P < scripts\dml\payment.sql
             >> logs\objects.log
@if errorlevel 1 goto PAYMENT_ERROR
@osql -Usa -P -S%1 -b -Usa -P < scripts\dml\ordstat.sql
             >> logs\objects.log
@if errorlevel 1 goto ORDERSTATUS_ERROR
@osql -Usa -P -S%1 -b -Usa -P < scripts\dml\delivery.sql
             >> logs\objects.log
@if errorlevel 1 goto DELIVERY_ERROR
@osql -Usa -P -S%1 -b -Usa -P < scripts\dml\stocklev.sql
             >> logs\objects.log
@if errorlevel 1 goto STOCKLEVEL_ERROR
@osql -Usa -P -S%1 -Usa -P < scripts\dml\version.sql
             >> logs\objects.log
@ECHO Database object creation complete.
@if  '%3'=='full' goto bulkload
@if  '%3'=='objectsfull' goto bulkload
goto end

:bulkload
@if exist logs\bulkload.log del logs\bulkload.log
             >nul
@ECHO Setting database options before load...
@osql -Usa -P -S%1 -b -Usa -P < scripts\utility\dbopt1.sql
             >> logs\objects.log
@if errorlevel 1 goto DBOPT1_ERROR
@ECHO Beginning data load and index creation...
@osql -Usa -P -S%1 -b -Usa -P < scripts\%2.war\ddl\idxhiscl.sql
             > logs\idxhiscl.log
@if '%4'==''     loader\IA64\bin\tpccldr -S%1 -W%2 -flogs\bulkload.log -
dscripts\%2.war\ddl -c0
@if errorlevel 1 goto END
@if '%4'=='normal' loader\IA64\bin\tpccldr -S%1 -W%2 -flogs\bulkload.log -
dscripts\%2.war\ddl -c0
@if errorlevel 1 goto END
@if '%4'=='scale_down' loader\IA64\bin\tpccldr -S%1 -W%2 -flogs\bulkload.log -
dscripts\%2.war\ddl -c1
@if errorlevel 1 goto END
goto bulkloaddone
:bulkloaddone
@ECHO Setting database options after load...
@osql -Usa -P -S%1 -b -Usa -P < scripts\utility\dbopt2.sql
             >> logs\bulkload.log
@if errorlevel 1 goto DBOPT2_ERROR
@ECHO Data load and index creation complete.

@ECHO.
@ECHO Calculating initial database space usage...
@cd..\acid\space
@call space.cmd %1
@cd..\..\setup

@if  '%3'=='full' goto backup
@if  '%3'=='objectsfull' goto backup
@if  '%3'=='bulkloadfull' goto backup
goto end

:backup
@if exist logs\backup.log del logs\backup.log
             >nul
@ECHO Backing up database...
@osql -Usa -P -S%1 -b -Usa -P < scripts\%2.war\database\backup.sql >
logs\backup.log
@if errorlevel 1 goto BACKUP_ERROR
@ECHO Database backup complete.
@if  '%3'=='full' goto verifyload
@if  '%3'=='objectsfull' goto verifyload
@if  '%3'=='bulkloadfull' goto verifyload
goto complete

:verifyload
@if exist logs\verifyload.log del logs\verifyload.log
             >nul
@Echo Verifying TPC-C database load...
@osql -Usa -P -S%1 -b -Usa -P < scripts\utility\verifytpccload.sql
             >  logs\verifyload.log
@if errorlevel 1 goto VERIFY_ERROR
@ECHO Check logs\verifyload.log to verify database load.

:complete
@ECHO
********************************************************************************
@ECHO *
*
@ECHO *  TPC-C V3 build complete.  Check logs directory for setup errors.
*
@ECHO *
*
@ECHO
********************************************************************************

goto end

:usage
@ECHO ********************************************************************************
@ECHO *                                                                              *
@ECHO *  The TPC-C setup command file requires the following parameters:              *
@ECHO *                                                                              *
@ECHO *  setup SERVER NUMWAR BLDOPT VERSION DBTYPE                                    *
@ECHO *                                                                              *
@ECHO *      SERVER  = machine name of server (use "" for local server)              *
@ECHO *      NUMWAR  = number of warehouses                                          *
@ECHO *      BLDOPT  = full, builddb, objects, objectsfull, bulkload,                *
@ECHO *                bulkloadfull, or backup                                       *
@ECHO *      DBTYPE  = normal or scale_down                                          *
@ECHO *                                                                              *
@ECHO *  Note #1: the BLDOPT and VERSION parameters are case sensitive.              *
@ECHO *                                                                              *
@ECHO *  Note #2: the DBTYPE is optional.  If no DBTYPE is specified, SETUP          *
@ECHO *           will default to NORMAL.                                            *
@ECHO *                                                                              *
@ECHO *  Example:                                                                    *
@ECHO *                                                                              *
@ECHO *  The following command would be used to build a complete 200                 *
@ECHO *  warehouse database on SQL Server 7.0 runnning on server \\myserver.         *
@ECHO *                                                                              *
@ECHO *      SETUP myserver 200 full                                                 *
@ECHO *                                                                              *
@ECHO *  NOTE 1: This command file does a backup of the database by default          *
@ECHO *  after the database build process is complete.  If you do not wish           *
@ECHO *  to make a backup (strongly discouraged), you must edit this file            *
@ECHO *  and comment that section out.  Also, if you need to run the dbcheck        *
@ECHO *  and the dbtables scripts on the fresh database load for an audit,          *
@ECHO *  you must either run them manually or edit this file to include them.       *
@ECHO *                                                                              *
@ECHO *  NOTE 2: The TPC-C setup program supports both Intel and Alpha               *
@ECHO *  systems.  It queries the %PROCESSOR_ARCHITECTURE% environment               *
@ECHO *  variable and runs the appropriate executables.                              *
@ECHO *                                                                              *
@ECHO ********************************************************************************
@goto end


:CREATE_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error in the database/backup device creation.
@echo.
@echo Check your CREATEDB.SQL, BACKUPDEV.SQL, LOGS\DB.LOG, and the
@echo SQL Server errorlog (PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL\LOG\ERRORLOG) for details.
@echo.
@goto END

:TABLES_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error in the table creation.
@echo.
@echo Verify that the FileGroup names specified in CREATEDB.SQL
@echo match those specified in SCRIPTS\DDL\TABLES.SQL.
@echo.
@goto END
```

```
:NEWORDER_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error in the creation of the New Order stored procedure.
@echo.
@echo Check your LOGS\OBJECTS.LOG, SCRIPTS\DML\NEWORD.SQL and the
@echo SQL Server errorlog (PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL\LOG\ERRORLOG) for details.
@echo.
@goto END

:PAYMENT_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error in the creation of the Payment stored procedure.
@echo.
@echo Check your LOGS\OBJECTS.LOG, SCRIPTS\DML\PAYMENT.SQL and the
@echo SQL Server errorlog (PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL\LOG\ERRORLOG) for details.
@echo.
@goto END

:ORDERSTATUS_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error in the creation of the Order Status stored procedure.
@echo.
@echo Check your LOGS\OBJECTS.LOG, SCRIPTS\DML\ORDSTAT.SQL and the
@echo SQL Server errorlog (PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL\LOG\ERRORLOG) for details.
@echo.
@goto END

:DELIVERY_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error in the creation of the Delivery stored procedure.
@echo.
@echo Check your LOGS\OBJECTS.LOG, SCRIPTS\DML\DELIVERY.SQL and the
@echo SQL Server errorlog (PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL\LOG\ERRORLOG) for details.
@echo.
@goto END

:STOCKLEVEL_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error in the creation of the Stock Level stored procedure.
@echo.
@echo Check your LOGS\OBJECTS.LOG, SCRIPTS\DML\STOCKLEV.SQL and the
@echo SQL Server errorlog (PROGRAM FILES\MICROSOFT SQL
SERVER\MSSQL\LOG\ERRORLOG) for details.
@echo.
@goto END

:DBOPT1_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error setting the database options before load.
@echo.
@echo Check your LOGS\OBJECTS.LOG and the SQL Server errorlog
@echo (PROGRAM FILES\MICROSOFT SQL SERVER\MSSQL\LOG\ERRORLOG) for details.
@echo.
@goto END

:DBOPT2_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error setting the database options after load.
@echo.
@echo Check your LOGS\OBJECTS.LOG and the SQL Server errorlog
@echo (MSSQL7\LOG\ERRORLOG) for details.
@echo.
@goto END

:BACKUP_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error backing up the database after load.
@echo.
@echo Check your LOGS\BACKUP.LOG and the SQL Server errorlog
@echo (PROGRAM FILES\MICROSOFT SQL SERVER\MSSQL\LOG\ERRORLOG) for details.
@echo.
@goto END

:VERIFY_ERROR
@echo.
@echo BUILD ABORTED!
@echo.
@echo There was an error performing TPC-C database verification.
@echo.
@echo Check your LOGS\VERIFYLOAD.LOG and the SQL Server errorlog
@echo (MSSQL7\LOG\ERRORLOG) for details.
@echo.
@goto END

:WRONGOS
@echo.
@echo BUILD ABORTED!
@echo.
@echo This version of the Microsoft TPC-C Kit is for use with SQL Server IA-64
ONLY!
@echo Please check http://mssqlperf.rte.microsoft.com for the latest Microsoft
TPC-C
@echo kit for use on non IA-64 systems.
@echo.
@goto END

:end

echo on
```

backup.sql

```
dump database tpcc to
tpccback01,
tpccback02,
tpccback03,
tpccback04,
tpccback05,
tpccback06,
tpccback07,
tpccback08,
tpccback09,
tpccback10,
tpccback11,
tpccback12,
tpccback13,
tpccback14,
tpccback15,
tpccback16,
tpccback17,
tpccback18,
tpccback19,
tpccback20,
tpccback21,
tpccback22,
tpccback23,
tpccback24,
tpccback25,
tpccback26,
tpccback27
with init, stats = 1
go

dump database tpcc to
tpccback28,
tpccback29,
tpccback30,
tpccback31,
tpccback32,
tpccback33,
tpccback34,
tpccback35,
tpccback36,
tpccback37,
tpccback38,
tpccback39,
tpccback40,
tpccback41,
tpccback42,
tpccback43,
tpccback44,
tpccback45,
tpccback46,
tpccback47,
tpccback48,
tpccback49,
tpccback50,
tpccback51,
tpccback52,
tpccback53,
tpccback54
with init, stats = 1
go
```

backupdev.sql

```
--  File:     BACKUPdevB.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Creates tpcc database Backup devices


use master
go

--  create backup devices



exec sp_addumpdevice
'disk','tpccback01','z:\dev\b001\tpccback01.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback02','z:\dev\b002\tpccback02.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback03','z:\dev\b003\tpccback03.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback04','z:\dev\b004\tpccback04.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback05','z:\dev\b005\tpccback05.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback06','z:\dev\b006\tpccback06.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback07','z:\dev\b007\tpccback07.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback08','z:\dev\b008\tpccback08.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback09','z:\dev\b009\tpccback09.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback10','z:\dev\b010\tpccback10.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback11','z:\dev\b011\tpccback11.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback12','z:\dev\b012\tpccback12.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback13','z:\dev\b013\tpccback13.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback14','z:\dev\b014\tpccback14.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback15','z:\dev\b015\tpccback15.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback16','z:\dev\b016\tpccback16.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback17','z:\dev\b017\tpccback17.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback18','z:\dev\b018\tpccback18.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback19','z:\dev\b019\tpccback19.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback20','z:\dev\b020\tpccback20.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback21','z:\dev\b021\tpccback21.37000w.72x.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback22','z:\dev\b022\tpccback22.37000w.72x.log600g.sqlsvr760'
```

```
exec sp_addumpdevice
'disk','tpccback23','z:\dev\b023\tpccback23.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback24','z:\dev\b024\tpccback24.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback25','z:\dev\b025\tpccback25.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback26','z:\dev\b026\tpccback26.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback27','z:\dev\b027\tpccback27.37000W.72X.log600g.sqlsvr760'

exec sp_addumpdevice
'disk','tpccback28','z:\dev\b028\tpccback28.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback29','z:\dev\b029\tpccback29.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback30','z:\dev\b030\tpccback30.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback31','z:\dev\b031\tpccback31.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback32','z:\dev\b032\tpccback32.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback33','z:\dev\b033\tpccback33.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback34','z:\dev\b034\tpccback34.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback35','z:\dev\b035\tpccback35.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback36','z:\dev\b036\tpccback36.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback37','z:\dev\b037\tpccback37.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback38','z:\dev\b038\tpccback38.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback39','z:\dev\b039\tpccback39.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback40','z:\dev\b040\tpccback40.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback41','z:\dev\b041\tpccback41.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback42','z:\dev\b042\tpccback42.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback43','z:\dev\b043\tpccback43.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback44','z:\dev\b044\tpccback44.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback45','z:\dev\b045\tpccback45.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback46','z:\dev\b046\tpccback46.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback47','z:\dev\b047\tpccback47.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback48','z:\dev\b048\tpccback48.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback49','z:\dev\b049\tpccback49.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback50','z:\dev\b050\tpccback50.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback51','z:\dev\b051\tpccback51.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback52','z:\dev\b052\tpccback52.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback53','z:\dev\b053\tpccback53.37000W.72X.log600g.sqlsvr760'
exec sp_addumpdevice
'disk','tpccback54','z:\dev\b054\tpccback54.37000W.72X.log600g.sqlsvr760'
go
```

createdb.sql

```
--  File:     CREATEDB.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Creates tpcc database and backup files


use master
go

--           Create temporary table for timing

if exists ( select name from sysobjects where name = 'tpcc_timer' )
            drop table tpcc_timer
go

create table tpcc_timer
(
            start_date                                          char(30),
            end_date                                            char(30)
)

insert      into tpcc_timer values (0,0)
go

--          Store starting time

update      tpcc_timer
set         start_date   = (select convert(char(30), getdate(),9))
go

--  create main database files

CREATE DATABASE tpcc
ON PRIMARY
(           NAME                        = MSSQL_tpcc_root,
            FILENAME     = "Z:\MSSQL_tpcc_root.mdf",
            SIZE                        = 8MB,
            FILEGROWTH   = 0),
FILEGROUP    MSSQL_misc_fg
(NAME=MSSQL_misc01, FILENAME='z:\dev\m001\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc02, FILENAME='z:\dev\m002\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc03, FILENAME='z:\dev\m003\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc04, FILENAME='z:\dev\m004\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc05, FILENAME='z:\dev\m005\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc06, FILENAME='z:\dev\m006\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc07, FILENAME='z:\dev\m007\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc08, FILENAME='z:\dev\m008\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc09, FILENAME='z:\dev\m009\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc10, FILENAME='z:\dev\m010\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc11, FILENAME='z:\dev\m011\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc12, FILENAME='z:\dev\m012\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc13, FILENAME='z:\dev\m013\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc14, FILENAME='z:\dev\m014\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc15, FILENAME='z:\dev\m015\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc16, FILENAME='z:\dev\m016\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc17, FILENAME='z:\dev\m017\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc18, FILENAME='z:\dev\m018\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc19, FILENAME='z:\dev\m019\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc20, FILENAME='z:\dev\m020\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc21, FILENAME='z:\dev\m021\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc22, FILENAME='z:\dev\m022\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc23, FILENAME='z:\dev\m023\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc24, FILENAME='z:\dev\m024\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc25, FILENAME='z:\dev\m025\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc26, FILENAME='z:\dev\m026\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc27, FILENAME='z:\dev\m027\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc28, FILENAME='z:\dev\m028\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc29, FILENAME='z:\dev\m029\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc30, FILENAME='z:\dev\m030\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc31, FILENAME='z:\dev\m031\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc32, FILENAME='z:\dev\m032\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc33, FILENAME='z:\dev\m033\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc34, FILENAME='z:\dev\m034\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc35, FILENAME='z:\dev\m035\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc36, FILENAME='z:\dev\m036\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc37, FILENAME='z:\dev\m037\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc38, FILENAME='z:\dev\m038\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc39, FILENAME='z:\dev\m039\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc40, FILENAME='z:\dev\m040\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc41, FILENAME='z:\dev\m041\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc42, FILENAME='z:\dev\m042\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc43, FILENAME='z:\dev\m043\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc44, FILENAME='z:\dev\m044\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc45, FILENAME='z:\dev\m045\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc46, FILENAME='z:\dev\m046\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc47, FILENAME='z:\dev\m047\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc48, FILENAME='z:\dev\m048\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc49, FILENAME='z:\dev\m049\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc50, FILENAME='z:\dev\m050\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc51, FILENAME='z:\dev\m051\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc52, FILENAME='z:\dev\m052\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc53, FILENAME='z:\dev\m053\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc54, FILENAME='z:\dev\m054\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc55, FILENAME='z:\dev\m055\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc56, FILENAME='z:\dev\m056\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc57, FILENAME='z:\dev\m057\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc58, FILENAME='z:\dev\m058\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc59, FILENAME='z:\dev\m059\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc60, FILENAME='z:\dev\m060\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc61, FILENAME='z:\dev\m061\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc62, FILENAME='z:\dev\m062\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc63, FILENAME='z:\dev\m063\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc64, FILENAME='z:\dev\m064\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc65, FILENAME='z:\dev\m065\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc66, FILENAME='z:\dev\m066\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc67, FILENAME='z:\dev\m067\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc68, FILENAME='z:\dev\m068\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc69, FILENAME='z:\dev\m069\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc70, FILENAME='z:\dev\m070\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc71, FILENAME='z:\dev\m071\', SIZE=20000MB, FILEGROWTH=0),
(NAME=MSSQL_misc72, FILENAME='z:\dev\m072\', SIZE=20000MB, FILEGROWTH=0),
FILEGROUP    MSSQL_cs_fg
(NAME=MSSQL_cs01, FILENAME='z:\dev\c001\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs02, FILENAME='z:\dev\c002\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs03, FILENAME='z:\dev\c003\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs04, FILENAME='z:\dev\c004\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs05, FILENAME='z:\dev\c005\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs06, FILENAME='z:\dev\c006\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs07, FILENAME='z:\dev\c007\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs08, FILENAME='z:\dev\c008\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs09, FILENAME='z:\dev\c009\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs10, FILENAME='z:\dev\c010\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs11, FILENAME='z:\dev\c011\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs12, FILENAME='z:\dev\c012\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs13, FILENAME='z:\dev\c013\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs14, FILENAME='z:\dev\c014\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs15, FILENAME='z:\dev\c015\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs16, FILENAME='z:\dev\c016\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs17, FILENAME='z:\dev\c017\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs18, FILENAME='z:\dev\c018\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs19, FILENAME='z:\dev\c019\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs20, FILENAME='z:\dev\c020\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs21, FILENAME='z:\dev\c021\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs22, FILENAME='z:\dev\c022\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs23, FILENAME='z:\dev\c023\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs24, FILENAME='z:\dev\c024\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs25, FILENAME='z:\dev\c025\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs26, FILENAME='z:\dev\c026\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs27, FILENAME='z:\dev\c027\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs28, FILENAME='z:\dev\c028\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs29, FILENAME='z:\dev\c029\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs30, FILENAME='z:\dev\c030\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs31, FILENAME='z:\dev\c031\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs32, FILENAME='z:\dev\c032\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs33, FILENAME='z:\dev\c033\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs34, FILENAME='z:\dev\c034\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs35, FILENAME='z:\dev\c035\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs36, FILENAME='z:\dev\c036\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs37, FILENAME='z:\dev\c037\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs38, FILENAME='z:\dev\c038\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs39, FILENAME='z:\dev\c039\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs40, FILENAME='z:\dev\c040\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs41, FILENAME='z:\dev\c041\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs42, FILENAME='z:\dev\c042\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs43, FILENAME='z:\dev\c043\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs44, FILENAME='z:\dev\c044\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs45, FILENAME='z:\dev\c045\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs46, FILENAME='z:\dev\c046\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs47, FILENAME='z:\dev\c047\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs48, FILENAME='z:\dev\c048\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs49, FILENAME='z:\dev\c049\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs50, FILENAME='z:\dev\c050\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs51, FILENAME='z:\dev\c051\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs52, FILENAME='z:\dev\c052\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs53, FILENAME='z:\dev\c053\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs54, FILENAME='z:\dev\c054\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs55, FILENAME='z:\dev\c055\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs56, FILENAME='z:\dev\c056\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs57, FILENAME='z:\dev\c057\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs58, FILENAME='z:\dev\c058\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs59, FILENAME='z:\dev\c059\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs60, FILENAME='z:\dev\c060\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs61, FILENAME='z:\dev\c061\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs62, FILENAME='z:\dev\c062\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs63, FILENAME='z:\dev\c063\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs64, FILENAME='z:\dev\c064\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs65, FILENAME='z:\dev\c065\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs66, FILENAME='z:\dev\c066\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs67, FILENAME='z:\dev\c067\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs68, FILENAME='z:\dev\c068\', SIZE=30000MB, FILEGROWTH=0),
```

```
(NAME=MSSQL_cs69, FILENAME='z:\dev\c069\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs70, FILENAME='z:\dev\c070\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs71, FILENAME='z:\dev\c071\', SIZE=30000MB, FILEGROWTH=0),
(NAME=MSSQL_cs72, FILENAME='z:\dev\c072\', SIZE=30000MB, FILEGROWTH=0)
LOG ON
(NAME=MSSQL_tpcc_log, FILENAME="z:\dev\log\", SIZE=200000MB, FILEGROWTH=0),
(NAME=MSSQL_tpcc_log2, FILENAME="z:\dev\log2\", SIZE=200000MB, FILEGROWTH=0),
(NAME=MSSQL_tpcc_log3, FILENAME="z:\dev\log3\", SIZE=200000MB, FILEGROWTH=0)
COLLATE Latin1_General_BIN
go

-- Store ending time
update       tpcc_timer
set          end_date     = (select convert(char(30), getdate(),9))
go

select "Elapsed time (in seconds): ", datediff(second,(select start_date from
tpcc_timer),(select end_date from tpcc_timer))

--           remove temporary table

if exists ( select name from sysobjects where name = 'tpcc_timer' )
             drop table tpcc_timer
go
```

dbopt1.sql

```
--  File:     DBOPT1.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Sets database options for data load


use master
go

exec sp_dboption tpcc,'select into/bulkcopy',true
exec sp_dboption tpcc,'trunc. log on chkpt.',true
exec sp_dboption tpcc,'torn page detection',false
go

use tpcc
go

checkpoint
go
```

dbopt2.sql

```
--  File:     DBOPT2.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Resets database options after data load

exec sp_dboption tpcc,'select into/bulkcopy',false
exec sp_dboption tpcc,'trunc. log on chkpt.',false
exec sp_dboption tpcc,'torn page detection',false
GO

USE tpcc
GO

CHECKPOINT
GO

sp_configure 'allow updates',1
GO

RECONFIGURE WITH OVERRIDE
GO

DECLARE      @msg                     varchar(50)

--                                    --
--        OPTIONS FOR SQL SERVER 2000 --
-- Set option values for user-defined indexes --
--                                    --

SET          @msg         = ' '
PRINT        @msg
SET          @msg         = 'Setting SQL Server indexoptions'
PRINT        @msg
SET          @msg         = ' '
PRINT        @msg

EXEC sp_indexoption       'customer',  'DisAllowPageLocks',      TRUE
EXEC sp_indexoption       'district',  'DisAllowPageLocks',      TRUE
EXEC sp_indexoption       'warehouse', 'DisAllowPageLocks',      TRUE
EXEC sp_indexoption       'stock',     'DisAllowPageLocks',      TRUE
EXEC sp_indexoption       'order_line','DisAllowRowLocks',       TRUE
EXEC sp_indexoption       'orders',    'DisAllowRowLocks',       TRUE
EXEC sp_indexoption       'new_order', 'DisAllowRowLocks',       TRUE
EXEC sp_indexoption       'item',                   'DisAllowRowLocks',
             TRUE
EXEC sp_indexoption       'item',                   'DisAllowPageLocks',
             TRUE
GO

Print ' '
Print '******************'
Print 'Pre-specified Locking Hierarchy:'
Print '        Lockflag = 0 ==> No pre-specified hierarchy'
Print '   Lockflag = 1 ==> Lock at Page-level then Table-level'
Print '   Lockflag = 2 ==> Lock at Row-level then Table-level'
Print '   Lockflag = 3 ==> Lock at Table-level'
Print ' '

SELECT       name,lockflags
FROM         sysindexes
WHERE        object_id('warehouse')    = id OR
             object_id('district')     = id OR
             object_id('customer')     = id OR
             object_id('stock')        = id OR
             object_id('orders')       = id OR
             object_id('order_line')   = id OR
             object_id('history')      = id OR
             object_id('new_order')    = id OR
             object_id('item')         = id
ORDER        BY lockflags asc
GO

sp_configure 'allow updates',0
GO

RECONFIGURE WITH OVERRIDE
GO

EXEC sp_dboption tpcc,     'auto update statistics',  FALSE
EXEC sp_dboption tpcc,     'auto create statistics',  FALSE
GO

EXEC sp_tableoption       'district',  'pintable',true
EXEC sp_tableoption       'warehouse', 'pintable',true
EXEC sp_tableoption       'new_order', 'pintable',true
EXEC sp_tableoption       'item',                   'pintable',true
GO
```

idxcuscl.sql

```
--  File:     IDXCUSCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Creates clustered index on customer table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'customer_c1' )
             drop index customer.customer_c1

create unique clustered index customer_c1 on customer(c_w_id, c_d_id, c_id)
             on MSSQL_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxcusnc.sql

```
--  File:     IDXCUSNC.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Creates non-clustered index on customer table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'customer_nc1' )
             drop index customer.customer_nc1

create unique nonclustered index customer_nc1 on customer(c_w_id, c_d_id, c_last,
c_first, c_id)
             on MSSQL_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxdiscl.sql

```
--  File:     IDXDISCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Creates clustered index on district table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'district_c1' )
             drop index district.district_c1

create unique clustered index  district_c1 on district(d_w_id, d_id)
             with fillfactor=100 on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxhiscl.sql

```
--  File:     IDXHISCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Creates clustered index on history table
--
--  CAUTION:  ****************************************
--  CAUTION:  This index is only beneficial for systems
--  CAUTION:  with 8 or more processors.
--  CAUTION:  It may negatively impact performance on
--  CAUTION:  on systems with less than 8 processors.
--  CAUTION:  ****************************************


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 = 'history_c1' )
             drop index history.history_c1

create unique clustered index history_c1 on history(h_c_w_id, h_date, h_c_d_id,
h_c_id, h_amount)
             on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxitmcl.sql

```
--  File:     IDXITMCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Creates clustered index on item table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'item_c1' )
             drop index item.item_c1

create unique clustered index item_c1 on item(i_id)
             on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxnodcl.sql

```
--  File:     IDXNODCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Creates clustered index on new_order table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'new_order_c1' )
             drop index new_order.new_order_c1

create unique clustered index new_order_c1 on new_order(no_w_id, no_d_id,
no_o_id)
             on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxodlcl.sql

```
--  File:     IDXODLCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Creates clustered index on order_line table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'order_line_c1' )
             drop index order_line.order_line_c1

create unique clustered index order_line_c1 on order_line(ol_w_id, ol_d_id,
ol_o_id, ol_number)
             on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxordcl.sql

```
--  File:     IDXORDCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Creates clustered index on orders table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'orders_c1' )
             drop index orders.orders_c1

create unique clustered index orders_c1 on orders(o_w_id, o_d_id, o_id)
             on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxordnc.sql

```
--  File:     IDXORDNC.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Creates non-clustered index on orders table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'orders_nc1' )
             drop index orders.orders_nc1

create index orders_nc1 on orders(o_w_id, o_d_id, o_c_id, o_id)
             on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxstkcl.sql

```
--  File:     IDXSTKCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Creates clustered index on stock table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'stock_c1' )
             drop index stock.stock_c1

create unique clustered index stock_c1 on stock(s_i_id, s_w_id)
             on MSSQL_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxwarcl.sql

```
--  File:     IDXWARCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Creates clustered index on warehouse table


use tpcc
go
```

```
declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'warehouse_c1' )
             drop index warehouse.warehouse_c1

create unique clustered index warehouse_c1 on warehouse(w_id)
             with fillfactor=100 on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

removedb.sql

```
--  File:     REMOVEDB.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Removes tpcc database and backup files


use master
go

--  remove any existing database and backup files

exec sp_dbremove tpcc, dropdev
go

exec sp_dropdevice 'tpccback01'
exec sp_dropdevice 'tpccback02'
exec sp_dropdevice 'tpccback03'
exec sp_dropdevice 'tpccback04'
exec sp_dropdevice 'tpccback05'
exec sp_dropdevice 'tpccback06'
exec sp_dropdevice 'tpccback07'
exec sp_dropdevice 'tpccback08'
exec sp_dropdevice 'tpccback09'
exec sp_dropdevice 'tpccback10'
exec sp_dropdevice 'tpccback11'
exec sp_dropdevice 'tpccback12'
exec sp_dropdevice 'tpccback13'
exec sp_dropdevice 'tpccback14'
exec sp_dropdevice 'tpccback15'
exec sp_dropdevice 'tpccback16'
exec sp_dropdevice 'tpccback17'
exec sp_dropdevice 'tpccback18'
exec sp_dropdevice 'tpccback19'
exec sp_dropdevice 'tpccback20'
exec sp_dropdevice 'tpccback21'
exec sp_dropdevice 'tpccback22'
exec sp_dropdevice 'tpccback23'
exec sp_dropdevice 'tpccback24'
exec sp_dropdevice 'tpccback25'
exec sp_dropdevice 'tpccback26'
exec sp_dropdevice 'tpccback27'
exec sp_dropdevice 'tpccback28'
exec sp_dropdevice 'tpccback29'
exec sp_dropdevice 'tpccback30'
exec sp_dropdevice 'tpccback31'
exec sp_dropdevice 'tpccback32'
exec sp_dropdevice 'tpccback33'
exec sp_dropdevice 'tpccback34'
exec sp_dropdevice 'tpccback35'
exec sp_dropdevice 'tpccback36'
exec sp_dropdevice 'tpccback37'
exec sp_dropdevice 'tpccback38'
exec sp_dropdevice 'tpccback39'
exec sp_dropdevice 'tpccback40'
exec sp_dropdevice 'tpccback41'
exec sp_dropdevice 'tpccback42'
exec sp_dropdevice 'tpccback43'
exec sp_dropdevice 'tpccback44'
exec sp_dropdevice 'tpccback45'
exec sp_dropdevice 'tpccback46'
exec sp_dropdevice 'tpccback47'
exec sp_dropdevice 'tpccback48'
exec sp_dropdevice 'tpccback49'
exec sp_dropdevice 'tpccback50'
exec sp_dropdevice 'tpccback51'
exec sp_dropdevice 'tpccback52'
exec sp_dropdevice 'tpccback53'
exec sp_dropdevice 'tpccback54'
```

tables.sql

```
--  File:     TABLES.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.42
--            Copyright Microsoft, 2002
--  Purpose:  Creates TPC-C tables

SET ANSI_NULL_DFLT_OFF ON
go

use tpcc
go

--
--  Remove all existing TPC-C tables
--

if exists ( select name from sysobjects where name = 'warehouse' )
             drop table warehouse
go
if exists ( select name from sysobjects where name = 'district' )
             drop table district
go
if exists ( select name from sysobjects where name = 'customer' )
             drop table customer
go
if exists ( select name from sysobjects where name = 'history' )
             drop table history
go
if exists ( select name from sysobjects where name = 'new_order' )
             drop table new_order
go
if exists ( select name from sysobjects where name = 'orders' )
             drop table orders
go
if exists ( select name from sysobjects where name = 'order_line' )
             drop table order_line
go
if exists ( select name from sysobjects where name = 'item' )
             drop table item
go
if exists ( select name from sysobjects where name = 'stock' )
             drop table stock
go

--
--  Create new tables
--

create table warehouse
(
             w_id                                    int,
             w_name                                  char(10),
             w_street_1                    char(20),
             w_street_2                    char(20),
             w_city                                  char(20),
             w_state                                 char(2),
             w_zip                                   char(9),
             w_tax                                   numeric(4,4),
             w_ytd                                   numeric(12,2)
) on MSSQL_misc_fg
go

create table district
(
             d_id                                    tinyint,
             d_w_id                                  int,
             d_name                                  char(10),
             d_street_1                    char(20),
             d_street_2                    char(20),
             d_city                                  char(20),
             d_state                                 char(2),
             d_zip                                   char(9),
             d_tax                                   numeric(4,4),
             d_ytd                                   numeric(12,2),
             d_next_o_id                   int
) on MSSQL_misc_fg
go

create table customer
(
             c_id                                    int,
             c_d_id                                  tinyint,
             c_w_id                                  int,
             c_first                                 char(16),
             c_middle                      char(2),
             c_last                                  char(16),
             c_street_1                    char(20),
             c_street_2                    char(20),
             c_city                                  char(20),
             c_state                                 char(2),
             c_zip                                   char(9),
             c_phone                                 char(16),
             c_since                                 datetime,
             c_credit                      char(2),
             c_credit_lim                  numeric(12,2),
             c_discount                    numeric(4,4),
             c_balance                     numeric(12,2),
             c_ytd_payment                 numeric(12,2),
             c_payment_cnt                 smallint,
             c_delivery_cnt                          smallint,
             c_data                                  char(500)
) on MSSQL_cs_fg
go

create table history
(
             h_c_id                                  int,
             h_c_d_id                      tinyint,
             h_c_w_id                      int,
             h_d_id                                  tinyint,
             h_w_id                                  int,
             h_date                                  datetime,
             h_amount                      numeric(6,2),
             h_data                                  char(24)
) on MSSQL_misc_fg
go

create table new_order
(
             no_o_id                                 int,
             no_d_id                                 tinyint,
             no_w_id                                 int
) on MSSQL_misc_fg
go

create table orders
(
             o_id                                    int,
             o_d_id                                  tinyint,
             o_w_id                                  int,
             o_c_id                                  int,
             o_entry_d                     datetime,
             o_carrier_id                  tinyint,
             o_ol_cnt                      tinyint,
             o_all_local                   tinyint
) on MSSQL_misc_fg
go

create table order_line
(
             ol_o_id                                 int,
             ol_d_id                                 tinyint,
             ol_w_id                                 int,
             ol_number                     tinyint,
             ol_i_id                                 int,
             ol_supply_w_id                          int,
             ol_delivery_d                 datetime,
             ol_quantity                   smallint,
             ol_amount                     numeric(6,2),
             ol_dist_info                  char(24)
) on MSSQL_misc_fg
go

create table item
(
```

```
            i_id                                                int,
            i_im_id                                             int,
            i_name                                              char(24),
            i_price                                             numeric(5,2),
            i_data                                              char(50)
) on MSSQL_misc_fg
go

create table stock
(
            s_i_id                                              int,
            s_w_id                                              int,
            s_quantity                          smallint,
            s_dist_01                           char(24),
            s_dist_02                           char(24),
            s_dist_03                           char(24),
            s_dist_04                           char(24),
            s_dist_05                           char(24),
            s_dist_06                           char(24),
            s_dist_07                           char(24),
            s_dist_08                           char(24),
            s_dist_09                           char(24),
            s_dist_10                           char(24),
            s_ytd                                               int,
            s_order_cnt                         smallint,
            s_remote_cnt                        smallint,
            s_data                                              char(50)
) on MSSQL_cs_fg
go
```

VerifyTpccLoad.sql

```
--  File:     VERIFYTPCCLOAD.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.41
--            Copyright Microsoft, 2001
--  Purpose:  Performs series of TPCC database checks to verify
--                that database load completed correctly

print       " "
select      convert(char(30), getdate(),9)
print       " "

use tpcc
go

--          ************************************
--
--          Check rows per table from SYSINDEXES
--
--          ************************************

print       'WAREHOUSE TABLE'

select      rowcnt
from        sysindexes
where       id          = object_id("warehouse")
go

print       'DISTRICT TABLE = (10 * No of warehouses)'

select      rowcnt
from        sysindexes
where       id          =object_id("district")
go

print       'ITEM TABLE = 100,000'

select      rowcnt
from        sysindexes
where       id          =object_id("item")
go

print       'CUSTOMER TABLE = (30,000 * No of warehouses)'

select      rowcnt
from        sysindexes
where       id          =object_id("customer")
go

print 'ORDERS TABLE = (30,000 * No of warehouses)'

select      rowcnt
from        sysindexes
where       id          =object_id("orders")
go

print       'HISTORY TABLE = (30,000 * No of warehouses)'

select      rowcnt
from        sysindexes
where       id          =object_id("history")
go

print       'STOCK TABLE = (100,000 * No of warehouses)'

select      rowcnt
from        sysindexes
where       id          =object_id("stock")
go

print       'ORDER_LINE TABLE = (300,000 * No of warehouses + some change)'

select      rowcnt
from        sysindexes
where       id          =object_id("order_line")
go

print       'NEW_ORDER TABLE = (9000 * No of warehouses)'

select      rowcnt
from        sysindexes
where       id          =object_id("new_order")
go

--          *************
--
--          Check indices
--
--          *************

print '**************Index Check****************'

use tpcc
go

sp_helpindex customer
go

sp_helpindex stock
go

sp_helpindex district
go

sp_helpindex item
go

sp_helpindex new_order
go

sp_helpindex orders
go

sp_helpindex order_line
go

sp_helpindex warehouse
go

sp_helpindex history
go
```

Stored Procedures

delivery.sql

```
--  File:     DELIVERY.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.42
--            Copyright Microsoft, 2002
--  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          int,

                                                    @o_carrier_id   smallint
as


declare @d_id tinyint,
        @o_id int,
        @c_id int,
        @total              numeric(12,2),
        @oid1 int,
        @oid2 int,
        @oid3 int,
        @oid4 int,
        @oid5 int,
        @oid6 int,
        @oid7 int,
        @oid8 int,
        @oid9 int,
        @oid10              int

select @d_id = 0

begin tran d

            while (@d_id < 10)
            begin

                  select            @d_id  = @d_id + 1,
                                    @total = 0,
                                    @o_id  = 0

                        select      top 1
                                    @o_id        = no_o_id
                        from        new_order (serializable updlock)
                        where       no_w_id      = @w_id and
                                    no_d_id      = @d_id
                        order       by no_o_id asc

                  if (@@rowcount <> 0)
                        begin

--  claim the order for this district

                                    delete      new_order
                                    where       no_w_id      = @w_id and
                                                no_d_id      = @d_id and
                                                no_o_id      = @o_id

--  set carrier_id on this order (and get customer id)

                                    update      orders
                                    set         o_carrier_id =
@o_carrier_id,
                                                @c_id
            = o_c_id
                                    where       o_w_id
            = @w_id and
                                                o_d_id
            = @d_id and
                                                o_id
            = @o_id

--  set date in all lineitems for this order (and sum amounts)

                                    update      order_line
                                    set         ol_delivery_d= getdate(),
                                                @total
            = @total + ol_amount
                                    where       ol_w_id
            = @w_id and
                                                ol_d_id
            = @d_id and
```

```
                                        ol_o_id
            = @o_id

--  accummulate lineitem amounts for this order into customer

                                    update      customer
                                    set         c_balance    = c_balance +
@total,
                                                c_delivery_cnt
            = c_delivery_cnt + 1
                                    where       c_w_id
            = @w_id and
                                                c_d_id
            = @d_id and
                                                c_id
            = @c_id

                    end

            select @oid1 = case @d_id when  1  then @o_id else @oid1 end,
              @oid2 = case @d_id when  2  then @o_id else @oid2 end,
              @oid3 = case @d_id when  3  then @o_id else @oid3 end,
              @oid4 = case @d_id when  4  then @o_id else @oid4 end,
              @oid5 = case @d_id when  5  then @o_id else @oid5 end,
              @oid6 = case @d_id when  6  then @o_id else @oid6 end,
              @oid7 = case @d_id when  7  then @o_id else @oid7 end,
              @oid8 = case @d_id when  8  then @o_id else @oid8 end,
              @oid9 = case @d_id when  9  then @o_id else @oid9 end,
              @oid10 = case @d_id when 10 then @o_id else @oid10 end

    end

commit tran d

-- return delivery data to client

select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

go
```

neword.sql

```
--  File:     NEWORD.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.42
--            Copyright Microsoft, 2002
--  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
int,
                                                                    @d_id
tinyint,
                                                                    @c_id
int,
                                                                    @o_ol_cnt
tinyint,
                                                                    @o_all_local
tinyint,
                                                                    @i_id1  int =
0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                                                                    @i_id2  int =
0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                                                                    @i_id3  int =
0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                                                                    @i_id4  int =
0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                                                                    @i_id5  int =
0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                                                                    @i_id6  int =
0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                                                                    @i_id7  int =
0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                                                                    @i_id8  int =
0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                                                                    @i_id9  int =
0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                                                                    @i_id10 int =
0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
                                                                    @i_id11 int =
0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
                                                                    @i_id12 int =
0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
                                                                    @i_id13 int =
0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
                                                                    @i_id14 int =
0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
                                                                    @i_id15 int =
0, @s_w_id15 int = 0, @ol_qty15 smallint = 0


as
declare     @w_tax          numeric(4,4),
        @d_tax          numeric(4,4),
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     numeric(4,4),
        @i_price        numeric(5,2),
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
             @li_no          int,
             @o_id                       int,
             @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
             @ol_number   int,
             @c_id_local  int

begin

begin transaction n

--  get district tax and next availible order id and update
--  plus initialize local variables

             update      district
             set         @d_tax       = d_tax,
                         @o_id        = d_next_o_id,
                         d_next_o_id  = d_next_o_id + 1,
                         @o_entry_d   = getdate(),
                         @li_no       = 0,
                         @commit_flag = 1
             where       d_w_id       = @w_id and
                         d_id         = @d_id

--  process orderlines

             while (@li_no < @o_ol_cnt)
             begin

                         select @li_no = @li_no + 1

--  set i_id, s_w_id, and qty for this lineitem

                         select      @li_id = case @li_no
                                              when 1 then @i_id1
                                              when 2 then @i_id2
                                              when 3 then @i_id3
                                              when 4 then @i_id4
                                              when 5 then @i_id5
                                              when 6 then @i_id6
                                              when 7 then @i_id7
                                              when 8 then @i_id8
                                              when 9 then @i_id9
                                              when 10 then @i_id10
                                              when 11 then @i_id11
                                              when 12 then @i_id12
                                              when 13 then @i_id13
                                              when 14 then @i_id14
                                              when 15 then @i_id15
                                              end,
                                     @li_s_w_id = case @li_no
                                              when 1 then @s_w_id1
                                              when 2 then @s_w_id2
                                              when 3 then @s_w_id3
                                              when 4 then @s_w_id4
                                              when 5 then @s_w_id5
                                              when 6 then @s_w_id6
                                              when 7 then @s_w_id7
                                              when 8 then @s_w_id8
                                              when 9 then @s_w_id9
                                              when 10 then
@s_w_id10
                                              when 11 then
@s_w_id11
                                              when 12 then
@s_w_id12
                                              when 13 then
@s_w_id13
                                              when 14 then
@s_w_id14
                                              when 15 then
@s_w_id15
                                              end,
                                     @li_qty = case  @li_no
                                              when 1 then @ol_qty1
                                              when 2 then @ol_qty2
                                              when 3 then @ol_qty3
                                              when 4 then @ol_qty4
                                              when 5 then @ol_qty5
                                              when 6 then @ol_qty6
                                              when 7 then @ol_qty7
                                              when 8 then @ol_qty8
                                              when 9 then @ol_qty9
                                              when 10 then @ol_qty10
                                              when 11 then @ol_qty11
                                              when 12 then @ol_qty12
                                              when 13 then @ol_qty13
                                              when 14 then @ol_qty14
                                              when 15 then @ol_qty15
                                              end

--  get item data (no one updates item)

                         select      @i_price = i_price,
                                     @i_name  = i_name,
                                     @i_data  = i_data
                         from        item (tablock repeatableread)
                         where       i_id = @li_id

--  update stock values

                         update      stock
                         set         s_ytd                    = s_ytd +
@li_qty,
                                     @s_quantity  = s_quantity   =
s_quantity - @li_qty +

case when (s_quantity - @li_qty < 10) then 91 else 0 end,
                                     s_order_cnt  = s_order_cnt + 1,
                                     s_remote_cnt = s_remote_cnt + case when
(@li_s_w_id = @w_id) then 0 else 1 end,
                                     @s_data                  = s_data,
                                     @s_dist                  = case @d_id
                                       when 1
then s_dist_01
                                       when 2
then s_dist_02
                                       when 3
then s_dist_03
                                       when 4
then s_dist_04
```

```
                                                        when 5
then s_dist_05
                                                        when 6
then s_dist_06
                                                        when 7
then s_dist_07
                                                        when 8
then s_dist_08
                                                        when 9
then s_dist_09
                                                        when 10
then s_dist_10
                                                        end
                        where       s_i_id              = @li_id and
                                    s_w_id              = @li_s_w_id

--  if there actually is a stock (and item) with these ids, go to work

                        if (@@rowcount > 0)
                        begin

--  insert order_line data (using data from item and stock)

                                    insert into order_line values(@o_id,

@d_id,

@w_id,

@li_no,

@li_id,

@li_s_w_id,

'dec 31, 1899',

@li_qty,

@i_price * @li_qty,

@s_dist)

--  send line-item data to client

                                    select      @i_name,
                                                @s_quantity,
                                                b_g = case when
( (patindex('%ORIGINAL%',@i_data) > 0) and

(patindex('%ORIGINAL%',@s_data) > 0) )
                                                      then 'B' else 'G'
end,
                                                @i_price,
                                                @i_price * @li_qty

                        end
                        else
                        begin

--  no item (or stock) found - triggers rollback condition

                                    select '',0,'',0,0
                                    select @commit_flag = 0

                        end
            end

--  get customer last name, discount, and credit rating

            select      @c_last     = c_last,
                        @c_discount = c_discount,
                        @c_credit   = c_credit,
                        @c_id_local = c_id
            from        customer (repeatableread)
            where       c_id                    = @c_id and
                        c_w_id                  = @w_id and
                        c_d_id                  = @d_id

--  insert fresh row into orders table

            insert into orders values (             @o_id,
                                                                @d_id,
                                                                @w_id,
                                                                @c_id_local,
                                                                @o_entry_d,
                                                                0,
                                                                @o_ol_cnt,
                                                                @o_all_local)

--  insert corresponding row into new-order table

            insert into new_order values (          @o_id,
                                                                @d_id,
                                                                @w_id)

--  select warehouse tax

            select      @w_tax      = w_tax
            from        warehouse (repeatableread)
            where       w_id        = @w_id

            if (@commit_flag = 1)

                        commit transaction n
            else

--  all that work for nuthin!!!

                        rollback transaction n

--  return order data to client

            select      @w_tax,
                        @d_tax,
                        @o_id,
                        @c_last,
                        @c_discount,
                        @c_credit,
                        @o_entry_d,
                        @commit_flag

end

go
```

ordstat.sql

```
--  File:     ORDSTAT.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.42
--            Copyright Microsoft, 2002
--  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       int,
                                        @d_id       tinyint,

                                        @c_id       int,
                                        @c_last     char(16) = ''


as

declare @c_balance                      numeric(12,2),
        @c_first                        char(16),
        @c_middle                       char(2),
        @o_id                           int,
        @o_entry_d                      datetime,
        @o_carrier_id                   smallint,
        @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
```

payment.sql

```
--  File:     PAYMENT.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.42
--            Copyright Microsoft, 2002
--  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           int,
                                                        @c_w_id         int,
                                                        @h_amount
numeric(6,2),
                                                        @d_id           tinyint,
                                                        @c_d_id         tinyint,
                                                        @c_id           int,
                                                        @c_last         char(16) =
''


as
declare  @w_street_1     char(20),
         @w_street_2     char(20),
         @w_city         char(20),
         @w_state        char(2),
         @w_zip          char(9),
         @w_name         char(10),
         @d_street_1     char(20),
         @d_street_2     char(20),
         @d_city         char(20),
         @d_state        char(2),
         @d_zip          char(9),
         @d_name         char(10),
         @c_first        char(16),
         @c_middle       char(2),
         @c_street_1     char(20),
         @c_street_2     char(20),
         @c_city         char(20),
         @c_state        char(2),
         @c_zip          char(9),
         @c_phone        char(16),
         @c_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   int,
              @c_id_local   int

select @screen_data = ''

begin tran p

--  get payment date

              select       @datetime = getdate()

              if (@c_id = 0)
                        begin

--  get customer id and info using last name

                                  select       @cnt         = count(*)
                                  from         customer (repeatableread)
                                  where        c_last       = @c_last and
                                               c_w_id       = @c_w_id and
                                               c_d_id       = @c_d_id


                                  select       @val = (@cnt + 1) / 2
                                  set          rowcount @val

                                  select       @c_id        = c_id
                                  from         customer (repeatableread)
                                  where        c_last       = @c_last and
                                               c_w_id       = @c_w_id and
                                               c_d_id       = @c_d_id
                                  order        by c_last, c_first

                                  set          rowcount 0
                        end

--  get customer info and update balances

                        update       customer
                        set          @c_balance    = c_balance  = c_balance - @h_amount,
                                     c_payment_cnt = c_payment_cnt + 1,
                                     c_ytd_payment = c_ytd_payment + @h_amount,
                                     @c_first      = c_first,
                                     @c_middle     = c_middle,
                                     @c_last                    = c_last,
                                     @c_street_1   = c_street_1,
                                     @c_street_2   = c_street_2,
                                     @c_city                    = c_city,
                                     @c_state      = c_state,
                                     @c_zip                     = c_zip,
                                     @c_phone      = c_phone,
                                     @c_credit     = c_credit,
                                     @c_credit_lim = c_credit_lim,
                                     @c_discount   = c_discount,
                                     @c_since      = c_since,
                                     @data                      = c_data,
                                     @c_id_local   = c_id
                        where        c_id                       = @c_id and
                                     c_w_id                     = @c_w_id and
                                     c_d_id                     = @c_d_id

--  if customer has bad credit get some more info

                        if (@c_credit = 'BC')
                        begin

--                      compute new info

                                  select @c_data             = convert(char(5),@c_id) +
                                                               convert(char(4),@c_d_id)
+
                                                               convert(char(5),@c_w_id)
+
                                                               convert(char(4),@d_id) +
                                                               convert(char(5),@w_id) +

convert(char(19),@h_amount) +
                                                               substring(@data, 1, 458)

--  update customer info

                                  update       customer
                                  set          c_data       = @c_data

                                  where        c_id         = @c_id and
                                               c_w_id       = @c_w_id and
                                               c_d_id       = @c_d_id

                                  select       @screen_data = substring (@c_data,1,200)
                        end

--  get district data and update year-to-date

                        update       district
                        set          d_ytd                      = d_ytd + @h_amount,
                                     @d_street_1   = d_street_1,
                                     @d_street_2   = d_street_2,
                                     @d_city                    = d_city,
                                     @d_state      = d_state,
                                     @d_zip                     = d_zip,
                                     @d_name                    = d_name,
                                     @d_id_local   = d_id
                        where        d_w_id                     = @w_id and
                                     d_id                       = @d_id

--  get warehouse data and update year-to-date

                        update       warehouse
                        set          w_ytd                      = w_ytd + @h_amount,
                                     @w_street_1   = w_street_1,
                                     @w_street_2   = w_street_2,
                                     @w_city                    = w_city,
                                     @w_state      = w_state,
                                     @w_zip                     = w_zip,
                                     @w_name                    = w_name,
                                     @w_id_local   = w_id
                        where        w_id                       = @w_id

--  create history record

                        insert into history values (            @c_id_local,
                                                                              @c_d_id,
                                                                              @c_w_id,
                                                                              @d_id_local,
                                                                              @w_id_local,
                                                                              @datetime,
                                                                              @h_amount,
                                                                              @w_name + '
' + @d_name)
commit tran p

--  return data to client

select       @c_id,
             @c_last,
             @datetime,
             @w_street_1,
             @w_street_2,
             @w_city,
             @w_state,
             @w_zip,
             @d_street_1,
             @d_street_2,
             @d_city,
             @d_state,
             @d_zip,
             @c_first,
             @c_middle,
             @c_street_1,
             @c_street_2,
             @c_city,
             @c_state,
             @c_zip,
             @c_phone,
             @c_since,
             @c_credit,
             @c_credit_lim,
             @c_discount,
             @c_balance,
             @screen_data

go
```

stocklev.sql

```
--  File:     STOCKLEV.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.42
--            Copyright Microsoft, 2002
--  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        int,
                                                     @d_id        tinyint,
                                                     @threshhold  smallint
as

declare      @o_id_low int,
             @o_id_high int

select       @o_id_low    = (d_next_o_id - 20),
             @o_id_high   = (d_next_o_id - 1)
from         district
where        d_w_id                    = @w_id and
             d_id                      = @d_id

select       count(distinct(s_i_id))
from         stock, order_line
where        ol_w_id                   =  @w_id and
             ol_d_id                   =  @d_id and
             ol_o_id                      between @o_id_low and
                                          @o_id_high and
             s_w_id                    =  ol_w_id and
             s_i_id                    =  ol_i_id and
             s_quantity   <  @threshhold

go
```

Loader Source Code

./buildia64.cmd

```
@if "%ECHOON%"=="" echo off

setlocal

REM
REM Setup environment variables
REM

if /I "%DRIVE%" == "" set DRIVE=d:
if /I "%ROOT%" == "" set ROOT=%DRIVE%
if /I "%SPHINX%" == "" set SPHINX=%ROOT%\shiloh
if /I "%SQLTOOLS%" == "" set SQLTOOLS=%ROOT%\sqltools
if /I "%MSVC%" == "" set MSVC=%SQLTOOLS%\msdev\vc98
if /I "%SHAREDIDE%" == "" set SHAREDIDE=%SQLTOOLS%\msdev\common\msdev98
if /I "%COMMON%" == "" set COMMON=%SPHINX%\common
if /I "%MSSDK%" == "" set MSSDK=%DRIVE%\mssdk

REM Sanity checks
if not exist %SPHINX% echo %SPHINX% cannot be found&goto usage
if not exist %MSSDK%\bin\win64\cl.exe echo 64-bit compiler cannot be found
in %MSSDK%\bin\win64&goto usage
if not exist %SQLTOOLS% echo SQLTOOLS not found in %SQLTOOLS%&goto usage
if not exist %MSVC% echo Visual C not found in %MSVC%&goto usage
if not exist %SHAREDIDE% echo Visual Studio Common files not found
in %SHAREDIDE%&goto usage

set BUILDCFGDEBUG="tpccldr - Win64 Debug"
set BUILDCFGRELEASE="tpccldr - Win64 Release"

REM Set some defaults
set BUILDCFG=%BUILDCFGRELEASE%
set MFLAGS=

:nextparm
if "%1" == "" goto noparm
if "%1" == "-?" goto usage
if "%1" == "/?" goto usage
if "%1" == "debug" set BUILDCFG=%BUILDCFGDEBUG% & goto shift
if "%1" == "DEBUG" set BUILDCFG=%BUILDCFGDEBUG% & goto shift
if "%1" == "normal" set BUILDCFG=%BUILDCFGRELEASE% & goto shift
if "%1" == "NORMAL" set BUILDCFG=%BUILDCFGRELEASE% & goto shift
if "%1" == "-a" set MFLAGS=%1
if "%1" == "clean" set MFLAGS=%1
if "%1" == "CLEAN" set MFLAGS=%1

:shift
shift
goto nextparm

:noparm

pushd %MSSDK%
call SetWin64.bat
popd

REM Override Important Stuff...
set Include=%COMMON%\include;%include%
set Lib=%COMMON%\%CPU%\lib\retail;%lib%

call nmake -f tpccldr.mak CFG=%BUILDCFG% %MFLAGS%

goto leave

:usage
echo.
echo Usage: %0 [type] [flags]
echo.
echo         'type' is one of:
echo                         debug
echo                         normal
echo
echo         'flags' is zero or more of:
echo                         -a           build all
echo                         clean        cleans up crufty bits
echo                         -?           display this message
echo.

:leave
endlocal
```

./tpccldr.mak

```
# Microsoft Developer Studio Generated NMAKE File, Format Version 4.10
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Console Application" 0x0103

!IF "$(CFG)" == ""
CFG=tpccldr - Win32 Debug
!MESSAGE No configuration specified.  Defaulting to tpccldr - Win32 Debug.
!ENDIF

!IF "$(CFG)" != "tpccldr - Win32 Release" && "$(CFG)" !=\
 "tpccldr - Win32 Debug"
!MESSAGE Invalid configuration "$(CFG)" specified.
!MESSAGE You can specify a configuration when running NMAKE on this makefile
!MESSAGE by defining the macro CFG on the command line.  For example:
!MESSAGE
!MESSAGE NMAKE /f "tpccldr.mak" CFG="tpccldr - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpccldr - Win32 Release" (based on "Win32 (x86) Console Application")
!MESSAGE "tpccldr - Win32 Debug" (based on "Win32 (x86) Console Application")
!MESSAGE
!ERROR An invalid configuration is specified.
!ENDIF

!IF "$(OS)" == "Windows_NT"
NULL=
!ELSE
NULL=nul
!ENDIF
################################################################################
# Begin Project
# PROP Target_Last_Scanned "tpccldr - Win32 Debug"
RSC=rc.exe
CPP=ecl.exe

!IF  "$(CFG)" == "tpccldr - 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 "objects"
# PROP Target_Dir ""
OUTDIR=.\bin
INTDIR=.\objects

ALL : "$(OUTDIR)\tpccldr.exe"

CLEAN :
        -@erase "$(INTDIR)\getargs.obj"
        -@erase "$(INTDIR)\random.obj"
        -@erase "$(INTDIR)\strings.obj"
        -@erase "$(INTDIR)\time.obj"
        -@erase "$(INTDIR)\tpccldr.obj"
        -@erase "$(OUTDIR)\tpccldr.exe"

"$(OUTDIR)" :
    if not exist "$(OUTDIR)/$(NULL)" mkdir "$(OUTDIR)"

"$(INTDIR)" :
    if not exist "$(INTDIR)/$(NULL)" mkdir "$(INTDIR)"

# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_CONSOLE" /YX /c
# ADD CPP /nologo /MT /W3 /GX /O2 /I "c:\mssql\dblib\include" /D "NDEBUG" /D
"WIN32" /D "_CONSOLE" /D "DBNTWIN32" /c
# SUBTRACT CPP /YX
CPP_PROJ=/nologo /MT /G2 /O3 /Qipo /I "c:\mssql\dblib\include" /D "NDEBUG" /D\
 "WIN32" /D "_CONSOLE" /D "DBNTWIN32" /Fo"$(INTDIR)/" /c
CPP_OBJS=.\objects/
CPP_SBRS=.\.
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
BSC32_FLAGS=/nologo /o"$(OUTDIR)/tpccldr.bsc"
BSC32_SBRS= \

LINK32=xilink.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:console /machine:I386
# ADD LINK32 ntwdblib.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:console /pdb:none /machine:I386
LINK32_FLAGS=odbcbcp.lib ntwdblib.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:console /pdb:none\
 /machine:IA64 /out:"$(OUTDIR)/tpccldr.exe"
LINK32_OBJS= \
             "$(INTDIR)\getargs.obj" \
             "$(INTDIR)\random.obj" \
             "$(INTDIR)\strings.obj" \
             "$(INTDIR)\time.obj" \
             "$(INTDIR)\tpccldr.obj"

"$(OUTDIR)\tpccldr.exe" : "$(OUTDIR)" $(DEF_FILE) $(LINK32_OBJS)
    $(LINK32) @<<
  $(LINK32_FLAGS) $(LINK32_OBJS)
<<

!ELSEIF  "$(CFG)" == "tpccldr - 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 "objects"
# PROP Target_Dir ""
OUTDIR=.\bin
INTDIR=.\objects

ALL : "$(OUTDIR)\tpccldr.exe"

CLEAN :
        -@erase "$(INTDIR)\getargs.obj"
        -@erase "$(INTDIR)\random.obj"
        -@erase "$(INTDIR)\strings.obj"
        -@erase "$(INTDIR)\time.obj"
        -@erase "$(INTDIR)\tpccldr.obj"
        -@erase "$(INTDIR)\vc40.idb"
        -@erase "$(INTDIR)\vc40.pdb"
        -@erase "$(OUTDIR)\tpccldr.exe"

"$(OUTDIR)" :
    if not exist "$(OUTDIR)/$(NULL)" mkdir "$(OUTDIR)"

"$(INTDIR)" :
    if not exist "$(INTDIR)/$(NULL)" mkdir "$(INTDIR)"

# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_CONSOLE"
/YX /c
# ADD CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /I "c:\mssql\dblib\include" /D
"_DEBUG" /D "WIN32" /D "_CONSOLE" /D "DBNTWIN32" /c
# SUBTRACT CPP /YX
CPP_PROJ=/nologo /MTd /W3 /Gm /GX /Zi /Od /I "c:\mssql\dblib\include" /D\
 "_DEBUG" /D "WIN32" /D "_CONSOLE" /D "DBNTWIN32" /Fo"$(INTDIR)/"\
 /Fd"$(INTDIR)/" /c
CPP_OBJS=.\objects/
CPP_SBRS=.\.
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
BSC32_FLAGS=/nologo /o"$(OUTDIR)/tpccldr.bsc"
BSC32_SBRS= \

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:console /debug /machine:I386
# ADD LINK32 ntwdblib.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:console /pdb:none /debug /machine:I386
LINK32_FLAGS=odbcbcp.lib ntwdblib.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:console /pdb:none /debug\
 /machine:I386 /out:"$(OUTDIR)/tpccldr.exe"
LINK32_OBJS= \
             "$(INTDIR)\getargs.obj" \
             "$(INTDIR)\random.obj" \
             "$(INTDIR)\strings.obj" \
             "$(INTDIR)\time.obj" \
             "$(INTDIR)\tpccldr.obj"

"$(OUTDIR)\tpccldr.exe" : "$(OUTDIR)" $(DEF_FILE) $(LINK32_OBJS)
    $(LINK32) @<<
  $(LINK32_FLAGS) $(LINK32_OBJS)
<<

!ENDIF

.c{$(CPP_OBJS)}.obj:
   $(CPP) $(CPP_PROJ) $<

.cpp{$(CPP_OBJS)}.obj:
   $(CPP) $(CPP_PROJ) $<

.cxx{$(CPP_OBJS)}.obj:
   $(CPP) $(CPP_PROJ) $<

.c{$(CPP_SBRS)}.sbr:
   $(CPP) $(CPP_PROJ) $<

.cpp{$(CPP_SBRS)}.sbr:
   $(CPP) $(CPP_PROJ) $<
```

```
.cxx{$(CPP_SBRS)}.sbr:
   $(CPP) $(CPP_PROJ) $<

################################################################################
# Begin Target

# Name "tpccldr - Win32 Release"
# Name "tpccldr - Win32 Debug"

!IF  "$(CFG)" == "tpccldr - Win32 Release"

!ELSEIF  "$(CFG)" == "tpccldr - Win32 Debug"

!ENDIF

################################################################################
# Begin Source File

SOURCE=.\src\random.c
DEP_CPP_RANDO=\
             ".\src\tpcc.h"\
             "\mssql\dblib\include\sqldb.h"\
             "\mssql\dblib\include\sqlfront.h"\


"$(INTDIR)\random.obj" : $(SOURCE) $(DEP_CPP_RANDO) "$(INTDIR)"
   $(CPP) $(CPP_PROJ) $(SOURCE)


# End Source File
################################################################################
# Begin Source File

SOURCE=.\src\strings.c
DEP_CPP_STRIN=\
             ".\src\tpcc.h"\
             "\mssql\dblib\include\sqldb.h"\
             "\mssql\dblib\include\sqlfront.h"\


"$(INTDIR)\strings.obj" : $(SOURCE) $(DEP_CPP_STRIN) "$(INTDIR)"
   $(CPP) $(CPP_PROJ) $(SOURCE)


# End Source File
################################################################################
# Begin Source File

SOURCE=.\src\time.c
DEP_CPP_TIME_=\
             ".\src\tpcc.h"\
             "\mssql\dblib\include\sqldb.h"\
             "\mssql\dblib\include\sqlfront.h"\


"$(INTDIR)\time.obj" : $(SOURCE) $(DEP_CPP_TIME_) "$(INTDIR)"
   $(CPP) $(CPP_PROJ) $(SOURCE)


# End Source File
################################################################################
# Begin Source File

SOURCE=.\src\tpccldr.c
DEP_CPP_TPCCL=\
             ".\src\tpcc.h"\
             "\mssql\dblib\include\sqldb.h"\
             "\mssql\dblib\include\sqlfront.h"\


"$(INTDIR)\tpccldr.obj" : $(SOURCE) $(DEP_CPP_TPCCL) "$(INTDIR)"
   $(CPP) $(CPP_PROJ) $(SOURCE)


# End Source File
################################################################################
# Begin Source File

SOURCE=.\src\getargs.c
DEP_CPP_GETAR=\
             ".\src\tpcc.h"\
             "\mssql\dblib\include\sqldb.h"\
             "\mssql\dblib\include\sqlfront.h"\


"$(INTDIR)\getargs.obj" : $(SOURCE) $(DEP_CPP_GETAR) "$(INTDIR)"
   $(CPP) $(CPP_PROJ) $(SOURCE)


# End Source File
# End Target
# End Project
################################################################################
```

./src/getargs.c

```
//           File:                       GETARGS.C
//                                                   Microsoft TPC-C Kit Ver.
4.20
//                                                   Copyright Microsoft, 1996,
1997, 1998, 1999
//           Purpose:     Source file for command line processing


// Includes
#include "tpcc.h"

//===================================================================
//
// Function name: GetArgsLoader
//
//===================================================================

void GetArgsLoader(int argc, char **argv, TPCCLDR_ARGS *pargs)
{
    int                 i;
    char     *ptr;

#ifdef DEBUG
    printf("[%ld]DBG: Entering GetArgsLoader()\n", (int) GetCurrentThreadId());
#endif

    /* init args struct with some useful values */
    pargs->server                                   =  SERVER;
    pargs->user                                     =  USER;
    pargs->password                                 =  PASSWORD;
    pargs->database                                 =  DATABASE;
    pargs->batch                                    =  BATCH;
    pargs->num_warehouses                           =  UNDEF;
            pargs->tables_all                                       =  TRUE;
            pargs->table_item                                       =  FALSE;
            pargs->table_warehouse                  =  FALSE;
            pargs->table_customer                   =  FALSE;
            pargs->table_orders                                     =  FALSE;
            pargs->loader_res_file                  =  LOADER_RES_FILE;
            pargs->pack_size                                =  DEFLDPACKSIZE;
            pargs->starting_warehouse  =  DEF_STARTING_WAREHOUSE;
            pargs->build_index                                      =
BUILD_INDEX;
            pargs->index_order                                      =
INDEX_ORDER;
            pargs->index_script_path   =  INDEX_SCRIPT_PATH;
            pargs->scale_down                                       =
SCALE_DOWN;

    /* check for zero command line args */
    if ( argc == 1 )
                        GetArgsLoaderUsage();

    for (i = 1; i < argc; ++i)
    {
                        if (argv[i][0] != '-' && argv[i][0] != '/')
                        {
                        printf("\nUnrecognized command");
                        GetArgsLoaderUsage();
                        exit(1);
                        }

                        ptr = argv[i];

                        switch (ptr[1])
                        {
                        case 'h':     /* Fall throught */
                        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);
                                                break;

                        case 's':
                                                pargs->starting_warehouse
= atol(ptr+2);
                                                break;

                        case 't':
                                                {
                                                        pargs-
>tables_all = FALSE;
                                                        if
(strcmp(ptr+2,"item") == 0)
                                                        pargs->table_item = TRUE;
                                                        else if
(strcmp(ptr+2,"warehouse") == 0)
                                                        pargs->table_warehouse = TRUE;
                                                        else if
(strcmp(ptr+2,"customer") == 0)
                                                        pargs->table_customer = TRUE;
                                                        else if
(strcmp(ptr+2,"orders") == 0)
                                                        pargs->table_orders = TRUE;
                                                        else
                                                        {
                        printf("\nUnrecognized command");
                                                        GetArgsLoaderUsage();
                                                        exit(1);
                                                        }
                                                        break;
                                                }
                                        case 'f':
                                                        pargs->loader_res_file =
ptr+2;
                                                        break;

                                        case 'p':
                                                        pargs->pack_size =
atol(ptr+2);
                                                        break;

                                        case 'i':
                                                        pargs->build_index =
atol(ptr+2);
                                                        break;

                                        case 'o':
                                                        pargs->index_order =
atol(ptr+2);
                                                        break;

                                        case 'c':
                                                        pargs->scale_down =
atol(ptr+2);
                                                        break;

                                case 'd':
                                                        pargs->index_script_path =
ptr+2;
                                                        break;

                                default:
                                                GetArgsLoaderUsage();
                                                exit(-1);
                                                break;
                                }
    }

    /* check for required args */
    if (pargs->num_warehouses == UNDEF )
    {
                        printf("Number of Warehouses is required\n");
                        exit(-2);
    }

    return;
}


//===================================================================
//
// Function name: GetArgsLoaderUsage
//
//===================================================================

void GetArgsLoaderUsage()
{

#ifdef DEBUG
    printf("[%ld]DBG: Entering GetArgsLoaderUsage()\n", (int)
GetCurrentThreadId());
#endif


    printf("TPCCLDR:\n\n");
            printf("Parameter
Default\n");
            printf("-------------------------------------------------------
----\n");
    printf("-W Number of Warehouses to Load                         Required \n");
    printf("-S Server                                               %s\n",
SERVER);
    printf("-U Username                                             %s\n", USER);
    printf("-P Password                                             %s\n",
PASSWORD);
    printf("-D Database                                             %s\n",
DATABASE);
            printf("-b Batch
Size                            %ld\n", (long) BATCH);
            printf("-p TDS packet
size                          %ld\n", (long) DEFLDPACKSIZE);
            printf("-f Loader Results Output
Filename                 %s\n", LOADER_RES_FILE);
            printf("-s Starting
Warehouse                        %ld\n", (long)
DEF_STARTING_WAREHOUSE);
            printf("-i Build Option (data = 0, data and index =
1)        %ld\n", (long) BUILD_INDEX);
            printf("-o Cluster Index Build Order (before = 1, after =
0)  %ld\n", (long) INDEX_ORDER);
            printf("-c Build Scaled Database (normal = 0, tiny =
1)       %ld\n", (long) SCALE_DOWN);
            printf("-d Index Script
Path              %s\n", INDEX_SCRIPT_PATH);
            printf("-t Table to Load                                                all
tables \n");
    printf("   [item|warehouse|customer|orders]\n");
    printf("   Notes: \n");
    printf("     - the '-t' parameter may be included multiple times to \n");
    printf("       specify multiple tables to be loaded \n");
    printf("     - 'item' loads ITEM table \n");
    printf("     - 'warehouse' loads WAREHOUSE, DISTRICT, and STOCK tables \n");
```

```
    printf("     - 'customer' loads CUSTOMER and HISTORY tables \n");
    printf("     - 'orders' load NEW-ORDER, ORDERS, ORDER-LINE tables \n");

            printf("\nNote:  Command line switches are case sensitive.\n");

    exit(0);
}
```

./src/random.c

```
//            File:                         RANDOM.C
//                                                      Microsoft TPC-C Kit Ver.
4.20
//                                                      Copyright Microsoft, 1996,
1997, 1998, 1999
//            Purpose:      Random number generation routines for database loader


// Includes
#include "tpcc.h"
#include "math.h"

// Defines
#define A             16807
#define M        2147483647
#define Q           127773      /* M div A */
#define R             2836      /* M mod A */
#define Thread              __declspec(thread)

// Globals
long          Thread Seed = 0;       /* thread local seed */

/*****************************************************************************
*                                                                            *
* random -                                                                   *
*       Implements a GOOD pseudo random number generator.  This generator    *
*       will/should? run the complete period before repeating.               *
*                                                                            *
* Copied from:                                                               *
*       Random Numbers Generators: Good Ones Are Hard to Find.               *
*       Communications of the ACM - October 1988 Volume 31 Number 10         *
*                                                                            *
* Machine Dependencies:                                                      *
*       long must be 2 ^ 31 - 1 or greater.                                   *
*                                                                            *
*****************************************************************************/

/*****************************************************************************
* seed - load the Seed value used in irand and drand.  Should be used before  *
*       first call to irand or drand.                                        *
*****************************************************************************/

void seed(long val)
{

#ifdef DEBUG
    printf("[%ld]DBG: Entering seed()...\n", (int) GetCurrentThreadId());
            printf("Old Seed %ld New Seed %ld\n",Seed, val);
#endif

    if ( val < 0 )
                val = abs(val);

    Seed = val;

}


/*****************************************************************************
*                                                                            *
* irand - returns a 32 bit integer pseudo random number with a period of     *
*       1 to 2 ^ 32 - 1.                                                     *
*                                                                            *
* parameters:                                                                *
*       none.                                                                *
*                                                                            *
* returns:                                                                   *
*       32 bit integer - defined as long ( see above ).                       *
*                                                                            *
* side effects:                                                              *
*       seed get recomputed.                                                 *
*****************************************************************************/

long irand()
{
    register long    s;      /* copy of seed */
    register long    test;   /* test flag */
    register long    hi;     /* tmp value for speed */
    register long    lo;     /* tmp value for speed */

#ifdef DEBUG
    printf("[%ld]DBG: Entering irand()...\n", (int) GetCurrentThreadId());
#endif

    s = Seed;
    hi = s / Q;
    lo = s % Q;

    test = A * lo - R * hi;
    if ( test > 0 )
                        Seed = test;

    else
                        Seed = test + M;

    return( Seed );
}


/*****************************************************************************
*                                                                            *
* drand - returns a double pseudo random number between 0.0 and 1.0.         *
*       See irand.                                                           *
*****************************************************************************/
double drand()
{

#ifdef DEBUG
    printf("[%ld]DBG: Entering drand()...\n", (int) GetCurrentThreadId());
#endif

    return( (double)irand() / 2147483647.0);
}


//===========================================================================
// Function   : RandomNumber
//
// Description:
//===========================================================================
long RandomNumber(long lower, long upper)
{
    long rand_num;

#ifdef DEBUG
    printf("[%ld]DBG: Entering RandomNumber()...\n", (int)
GetCurrentThreadId());
#endif

                if ( upper == lower )      /* pgd 08-13-96 perf enhancement */
                        return lower;

                upper++;

    if ( upper <= lower )
                        rand_num = upper;
    else                rand_num = lower + irand() % (upper - lower); /* pgd
08-13-96 perf enhancement */


#ifdef DEBUG
    printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                                                        (int)  GetCurrentThreadId(),
lower, upper, rand_num);
#endif


    return rand_num;
}


#if 0

//Orginal code pgd 08/13/96

long RandomNumber(long lower,
                                                        long upper)
{
    long rand_num;

#ifdef DEBUG
    printf("[%ld]DBG: Entering RandomNumber()...\n", (int)
GetCurrentThreadId());
#endif

                upper++;

    if ((upper <= lower))
                        rand_num = upper;
    else                rand_num = lower + irand() % ((upper > lower) ? upper
- lower : upper);


#ifdef DEBUG
    printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                                                        (int)  GetCurrentThreadId(),
lower, upper, rand_num);
#endif


    return rand_num;
}
#endif

//===========================================================================
// Function   : NURand
//
// Description:
//===========================================================================
long NURand(int iConst,
                long x,
                long y,
                long C)
{
    long rand_num;

#ifdef DEBUG
    printf("[%ld]DBG: Entering NURand()...\n", (int) GetCurrentThreadId());
#endif

    rand_num = (((RandomNumber(0,iConst) | RandomNumber(x,y)) + C) % (y-x+1))+x;

#ifdef DEBUG
    printf("[%ld]DBG: NURand: num = %d\n", (int) GetCurrentThreadId(),
rand_num);
#endif

    return rand_num;
}
```

./src/strings.c

```
//            File:                         STRINGS.C
//                                                      Microsoft TPC-C Kit Ver.
4.20
//                                                      Copyright Microsoft, 1996,
1997, 1998, 1999
//            Purpose:      Source file for database loader string functions


// Includes
#include "tpcc.h"
#include <string.h>
#include <ctype.h>

//===========================================================================
//
// Function name: MakeAddress
//
//===========================================================================

void MakeAddress(char *street_1,
                                                        char *street_2,
                                                        char *city,
                                                        char *state,
                                                        char *zip)
{

#ifdef DEBUG
    printf("[%ld]DBG: Entering MakeAddress()\n", (int) GetCurrentThreadId());
#endif

    MakeAlphaString (10, 20, ADDRESS_LEN, street_1);
    MakeAlphaString (10, 20, ADDRESS_LEN, street_2);
    MakeAlphaString (10, 20, ADDRESS_LEN, city);
    MakeAlphaString ( 2,  2, STATE_LEN, state);
    MakeZipNumberString( 9,  9, ZIP_LEN, zip);

#ifdef DEBUG
    printf("[%ld]DBG: MakeAddress: street_1: %s, street_2: %s, city: %s,
state: %s, zip: %s\n",
                                                        (int) GetCurrentThreadId(), street_1,
street_2, city, state, zip);
#endif

    return;

}


//===========================================================================
//
// Function name: LastName
//
//===========================================================================

void LastName(int  num,
                                char *name)
{
    static char *n[] =
    {
                        "BAR" , "OUGHT", "ABLE" , "PRI"  , "PRES",
                        "ESE" , "ANTI" , "CALLY", "ATION", "EING"
    };

#ifdef DEBUG
    printf("[%ld]DBG: Entering LastName()\n", (int) GetCurrentThreadId());
#endif

    if ((num >= 0) && (num < 1000))
    {
                        strcpy(name, n[(num/100)%10]);
                        strcat(name, n[(num/10)%10]);
                        strcat(name, n[(num/1)%10]);

                        if (strlen(name) < LAST_NAME_LEN)
                        {
                                PaddString(LAST_NAME_LEN, name);
                        }

    }
    else
    {
                        printf("\nError in LastName()... num <%ld> out of
range (0,999)\n", num);
                        exit(-1);
    }


#ifdef DEBUG
    printf("[%ld]DBG: LastName: num = [%d] ==> [%d][%d][%d]\n",
                                                        (int) GetCurrentThreadId(), num, num/100,
(num/10)%10, num%10);
            printf("[%ld]DBG: LastName: String = %s\n", (int)
GetCurrentThreadId(), name);
#endif

    return;

}
```

```
//==============================================================
//
// Function name: MakeAlphaString
//
//==============================================================

//philipdu 08/13/96 Changed MakeAlphaString to use A-Z, a-z, and 0-9 in
//accordance with spec see below:
//The spec says:
//4.3.2.2     The notation random a-string [x .. y]
//(respectively, n-string [x .. y]) represents a string of random alphanumeric
//(respectively, numeric) characters of a random length of minimum x, maximum y,
//and mean (y+x)/2. Alphanumerics are A..Z, a..z, and 0..9.  The only other
//requirement is that the character set used "must be able to represent a
minimum
//of 128 different characters".  We are using 8-bit chars, so this is a non
issue.
//It is completely unreasonable to stuff non-printing chars into the text fields.
//-CLevine 08/13/96

int MakeAlphaString( int  x, int  y, int  z, char *str)
{
            int                     len;
            int                     i;
    char    cc = 'a';
            static      char chArray[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
            static      int         chArrayMax = 61;

#ifdef DEBUG
    printf("[%ld]DBG: Entering MakeAlphaString()\n", (int)
GetCurrentThreadId());
#endif

            len= RandomNumber(x, y);

            for (i=0; i<len; i++)
            {
                        cc = chArray[RandomNumber(0, chArrayMax)];
        str[i] = cc;
    }
            if ( len < z )
                        memset(str+len, ' ', z - len);
            str[len] = 0;

    return len;
}


//==============================================================
//
// Function name: MakeOriginalAlphaString
//
//==============================================================

int MakeOriginalAlphaString(int  x,

int  y,

            int  z,

char *str,

            int  percent)
{
    int                 len;
    int                 val;
    int                 start;

#ifdef DEBUG
    printf("[%ld]DBG: Entering MakeOriginalAlphaString()\n", (int)
GetCurrentThreadId());
#endif

    // verify prercentage is valid
    if ((percent < 0) || (percent > 100))
    {
                        printf("MakeOrigianlAlphaString: Invalid
percentage: %d\n", percent);
                        exit(-1);
    }

    // verify string is at least 8 chars in length
    if ((x + y) <= 8)
    {
                        printf("MakeOriginalAlphaString: string length must
be >= 8\n");
                        exit(-1);
    }

    // Make Alpha String
    len = MakeAlphaString(x,y, z, str);

    val = RandomNumber(1,100);
    if (val <= percent)
    {
                        start = RandomNumber(0, len - 8);
                        strncpy(str + start, "ORIGINAL", 8);
    }


#ifdef DEBUG
    printf("[%ld]DBG: MakeOriginalAlphaString: : %s\n",
                                    (int) GetCurrentThreadId(), str);
#endif

    return (int)strlen(str);
}

//==============================================================
//
// Function name: MakeNumberString
//
//==============================================================
int MakeNumberString(int  x, int  y, int  z, char *str)
{
            char tmp[16];

           //MakeNumberString is always called MakeZipNumberString(16, 16, 16,
string)

            memset(str, '0', 16);
            _itoa(RandomNumber(0, 99999999), tmp, 10);
            memcpy(str, tmp, strlen(tmp));

            _itoa(RandomNumber(0, 99999999), tmp, 10);
            memcpy(str+8, tmp, strlen(tmp));

            str[16] = 0;

    return 16;
}


//==============================================================
//
// Function name: MakeZipNumberString
//
//==============================================================
int MakeZipNumberString(int  x, int  y, int  z, char *str)
{
            char tmp[16];

           //MakeZipNumberString is always called MakeZipNumberString(9, 9, 9,
string)

            strcpy(str, "000011111");

            _itoa(RandomNumber(0, 9999), tmp, 10);
            memcpy(str, tmp, strlen(tmp));

    return 9;
}


//==============================================================
//
// Function name: InitString
//
//==============================================================
void InitString(char *str, int len)
{
#ifdef DEBUG
    printf("[%ld]DBG: Entering InitString()\n", (int) GetCurrentThreadId());
#endif

            memset(str, ' ', len);
    str[len] = 0;
}


//==============================================================
// Function name: InitAddress
//
// Description:
//
//==============================================================

void InitAddress(char *street_1, char *street_2, char *city, char *state, char
*zip)
{
            memset(street_1, ' ', ADDRESS_LEN+1);
            memset(street_2, ' ', ADDRESS_LEN+1);
            memset(city, ' ', ADDRESS_LEN+1);

    street_1[ADDRESS_LEN+1] = 0;
    street_2[ADDRESS_LEN+1] = 0;
    city[ADDRESS_LEN+1]     = 0;

            memset(state, ' ', STATE_LEN+1);
    state[STATE_LEN+1] = 0;

            memset(zip, ' ', ZIP_LEN+1);
    zip[ZIP_LEN+1] = 0;
}


//==============================================================
//
// Function name: PaddString
//
//==============================================================

void PaddString(int max, char *name)
{
            size_t       len;

            len = strlen(name);
            if ( len < max )
                        memset(name+len, ' ', max - len);
            name[max] = 0;

            return;
}
```

./src/time.c

```
//            File:                       TIME.C
//                                                      Microsoft TPC-C Kit Ver.
4.20
//                                                      Copyright Microsoft, 1996,
1997, 1998, 1999
//            Purpose:      Source file for time functions


// Includes
#include "tpcc.h"

// Globals
static time_t start_sec;


//==============================================================
//
// Function name: TimeNow
//
//==============================================================

time_t TimeNow()
{
   time_t    time_now;
   struct    _timeb el_time;

#ifdef DEBUG
    printf("[%ld]DBG: Entering TimeNow()\n", (int) GetCurrentThreadId());
#endif

   _ftime(&el_time);

   time_now = ((el_time.time - start_sec) * 1000) + el_time.millitm;

   return time_now;
}
```

./src/tpcc.h

```
//            File:                       TPCC.H
//                                                      Microsoft TPC-C Kit Ver.
4.41
//                                                      Copyright Microsoft, 1996,
1997, 1998, 1999, 2000, 2001
//            Purpose:      Header file for TPC-C database loader


// Build number of TPC Benchmark Kit
#define TPCKIT_VER          "4.41"

// General headers
#include <windows.h>
#include <winbase.h>
#include <stdlib.h>
#include <stdio.h>
#include <process.h>
#include <stddef.h>
#include <stdarg.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <sys\types.h>

// ODBC headers
#include <sql.h>
#include <sqlext.h>
#include <odbcss.h>

// General constants
#define MILLI                               1000
#define FALSE                                                   0
#define TRUE                                                    1
#define UNDEF                                                                   -1
#define MINPRINTASCII                                             32
#define MAXPRINTASCII                                             126

// Default environment constants
#define SERVER                                                  ""
#define DATABASE                                                "tpcc"
#define USER                                                    "sa"
#define PASSWORD                                  ""

// Default loader arguments
#define BATCH                               10000
#define DEFLDPACKSIZE                                           32768
#define LOADER_RES_FILE
"C:\\MSTPCC.442\\SETUP\\logs\\load.out"
#define LOG_PATH                                        "C:\\MSTPCC.442\\SETUP\\LOGS\\";
#define LOADER_NURAND_C                                           123
#define DEF_STARTING_WAREHOUSE                                  1
#define BUILD_INDEX
            1           // build both data and indexes
#define INDEX_ORDER
            1           // build indexes before load
#define SCALE_DOWN
            0   // build a normal scale database
#define INDEX_SCRIPT_PATH  "scripts"

typedef struct
{
    char                                                    *server;
```

```
    char                                        *database;
    char                                        *user;
    char                                        *password;
            BOOL                                                tables_all;
                    // set if loading all tables
            BOOL                                                table_item;
                    // set if loading ITEM table specifically
            BOOL
            table_warehouse;           // set if loading WAREHOUSE, DISTRICT,
and STOCK
            BOOL
            table_customer;                     // set if loading CUSTOMER
and HISTORY
            BOOL                                                table_orders;
            // set if loading NEW-ORDER, ORDERS, ORDER-LINE
    long                                        num_warehouses;
    long                                        batch;
    long                                        verbose;
            long                                                pack_size;
            char
            *loader_res_file;
            char                                                *log_path;
            char
            *synch_servername;
            long
            case_sensitivity;
            long
            starting_warehouse;
            long                                                build_index;
            long                                                index_order;
            long                                                scale_down;
            char
            *index_script_path;
} TPCCLDR_ARGS;

// String length constants
            #define SERVER_NAME_LEN             20
            #define DATABASE_NAME_LEN           20
            #define USER_NAME_LEN               20
            #define PASSWORD_LEN                20
            #define TABLE_NAME_LEN              20
            #define I_DATA_LEN                  50
            #define I_NAME_LEN                  24
            #define BRAND_LEN                    1
            #define LAST_NAME_LEN               16
            #define W_NAME_LEN                  10
            #define ADDRESS_LEN                 20
            #define STATE_LEN                    2
            #define ZIP_LEN                      9
            #define S_DIST_LEN                  24
            #define S_DATA_LEN                  50
            #define D_NAME_LEN                  10
            #define FIRST_NAME_LEN              16
            #define MIDDLE_NAME_LEN              2
            #define PHONE_LEN                   16
            #define CREDIT_LEN                   2
            #define C_DATA_LEN                 500
            #define H_DATA_LEN                  24
            #define DIST_INFO_LEN               24
            #define MAX_OL_NEW_ORDER_ITEMS      15
            #define MAX_OL_ORDER_STATUS_ITEMS   15
            #define STATUS_LEN                  25
            #define OL_DIST_INFO_LEN                    24
            #define C_SINCE_LEN
23
            #define H_DATE_LEN
23
            #define OL_DELIVERY_D_LEN                   23
            #define O_ENTRY_D_LEN               23


// Functions in random.c
void        seed();
long        irand();
double      drand();
void        WUCreate();
short       WURand();
long        RandomNumber(long lower, long upper);

// Functions in getargs.c;
void        GetArgsLoader();
void        GetArgsLoaderUsage();

// Functions in time.c
long        TimeNow();

// Functions in strings.c
void        MakeAddress();
void        LastName();
int         MakeAlphaString();
int         MakeOriginalAlphaString();
int         MakeNumberString();
int         MakeZipNumberString();
void        InitString();
void        InitAddress();
void        PaddString();
```

./src/tpccldr.c

```
//          File:                       TPCCLDR.C
//                                                  Microsoft TPC-C Kit Ver.
4.41
//                                                  Copyright Microsoft, 1996,
1997, 1998, 1999, 2000, 2001
//          Purpose:     Source file for TPC-C database loader


// Includes
#include "tpcc.h"
#include "search.h"

// Defines
#define MAXITEMS                100000
#define MAXITEMS_SCALE_DOWN             100
#define CUSTOMERS_PER_DISTRICT  3000
#define CUSTOMERS_SCALE_DOWN            30
#define DISTRICT_PER_WAREHOUSE  10
#define ORDERS_PER_DISTRICT     3000
#define ORDERS_SCALE_DOWN       30
#define MAX_CUSTOMER_THREADS    2
#define MAX_ORDER_THREADS       3
#define MAX_MAIN_THREADS        4

// Functions declarations

void HandleErrorDBC (SQLHDBC  hdbc1);

void CheckDataBase();

long NURand();
void LoadItem();
void LoadWarehouse();

void Stock();
void District();

void LoadCustomer();
void CustomerBufInit();
void CustomerBufLoad();
void LoadCustomerTable();
void LoadHistoryTable();

void LoadOrders();
void OrdersBufInit();
void OrdersBufLoad();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void GetPermutation();
void CheckForCommit();
void OpenConnections();
void BuildIndex();
void FormatDate ();

// Shared memory structures

typedef struct
{
    long                ol;
    long                ol_i_id;
    long                ol_supply_w_id;
    short               ol_quantity;
    double              ol_amount;
    char                ol_dist_info[DIST_INFO_LEN+1];
            char
            ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCT;

typedef struct
{
    long                o_id;
    short               o_d_id;
    long                o_w_id;
    long                o_c_id;
    short               o_carrier_id;
    short               o_ol_cnt;
    short               o_all_local;
            ORDER_LINE_STRUCT   o_ol[15];
} ORDERS_STRUCT;

typedef struct
{
    long                                        c_id;
    short                                       c_d_id;
    long                                        c_w_id;
            char
            c_first[FIRST_NAME_LEN+1];
    char
            c_middle[MIDDLE_NAME_LEN+1];
    char                                        c_last[LAST_NAME_LEN+1];
    char                                        c_street_1[ADDRESS_LEN+1];
    char                                        c_street_2[ADDRESS_LEN+1];
    char                                        c_city[ADDRESS_LEN+1];
    char                                        c_state[STATE_LEN+1];
    char                                        c_zip[ZIP_LEN+1];
    char                                        c_phone[PHONE_LEN+1];
    char                                        c_credit[CREDIT_LEN+1];
    double                                      c_credit_lim;
    double                                      c_discount;
//  fix to avoid ODBC float to numeric conversion problem.
//  double                                      c_balance;
            char                                                c_balance[6];

    double                                      c_ytd_payment;

            short                               c_payment_cnt;
            short
            c_delivery_cnt;
            char
            c_data[C_DATA_LEN+1];
            double                                              h_amount;
    char                                        h_data[H_DATA_LEN+1];
} CUSTOMER_STRUCT;

typedef struct
{
            char
            c_last[LAST_NAME_LEN+1];
            char
            c_first[FIRST_NAME_LEN+1];
    long                                                        c_id;
} CUSTOMER_SORT_STRUCT;

typedef struct
{
    long                time_start;
} LOADER_TIME_STRUCT;


// Global variables

char          szLastError[300];

HENV          henv;

HDBC          v_hdbc;
              // for SQL Server version verification
HDBC          i_hdbc1;
              // for ITEM table
HDBC          w_hdbc1;
              // for WAREHOUSE, DISTRICT, STOCK
HDBC          c_hdbc1;
              // for CUSTOMER
HDBC          c_hdbc2;
              // for HISTORY
HDBC          o_hdbc1;
              // for ORDERS
HDBC          o_hdbc2;
              // for NEW-ORDER
HDBC          o_hdbc3;
              // for ORDER-LINE

HSTMT         v_hstmt;
              // for SQL Server version verification
HSTMT         i_hstmt1;
HSTMT         w_hstmt1;
HSTMT         c_hstmt1, c_hstmt2;
HSTMT         o_hstmt1, o_hstmt2, o_hstmt3;

ORDERS_STRUCT   orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT customer_buf[CUSTOMERS_PER_DISTRICT];
long            orders_rows_loaded;
long            new_order_rows_loaded;
long            order_line_rows_loaded;
long            history_rows_loaded;
long            customer_rows_loaded;
long            stock_rows_loaded;
long            district_rows_loaded;
long            item_rows_loaded;
long            warehouse_rows_loaded;
long            main_time_start;
long            main_time_end;
long                                max_items;
long                                customers_per_district;
long                                orders_per_district;
long                                first_new_order;
long                                last_new_order;

TPCCLDR_ARGS    *aptr, args;

//==================================================================
//
// Function name: main
//
//==================================================================

int main(int  argc, char **argv)
{

              DWORD           dwThreadID[MAX_MAIN_THREADS];
              HANDLE          hThread[MAX_MAIN_THREADS];
              FILE            *fLoader;
              char            buffer[255];
              int                                                   i;
              int                                                   num_procs;

              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);
            }

            // if build index before load
            // start creating Index
            if ((aptr->build_index == 1) && (aptr->index_order == 1))
            {
                        if (aptr->tables_all || aptr->table_item)
                                    BuildIndex("idxitmcl");

                        if (aptr->tables_all || aptr->table_warehouse)
                        {
                                    BuildIndex("idxwarcl");
                                    BuildIndex("idxdiscl");
                                    BuildIndex("idxstkcl");
                        }

                        if (aptr->tables_all || aptr->table_customer)
                        {
                                    BuildIndex("idxcuscl");
                                    // check the number of processors on
this system
                                    // if 8 or more processors, then build
index on History.
                                    // if less than 8 processors, do not
build the index
                                    //num_procs =
atoi(getenv( "NUMBER_OF_PROCESSORS" ));
                                    //if ( num_procs >= 8 )
                                                BuildIndex("idxhiscl");
                        }

                        if (aptr->tables_all || aptr->table_orders)
                        {
                                    BuildIndex("idxordcl");
                                    BuildIndex("idxnodcl");
                                    BuildIndex("idxodlcl");
                        }
            }

    // start loading data

    sprintf(buffer,"TPC-C load started for %ld warehouses.\n",aptr-
>num_warehouses);

            printf("%s",buffer);
            fprintf(fLoader,"%s",buffer);

            main_time_start = (TimeNow() / MILLI);

            // start parallel load threads


            if (aptr->tables_all || aptr->table_item)
            {
                        fprintf(fLoader, "\nStarting loader threads for:
item\n");

                        hThread[0] = CreateThread(NULL,

                          0,

                          (LPTHREAD_START_ROUTINE) LoadItem,

                          NULL,

                          0,

                          &dwThreadID[0]);

                        if (hThread[0] == NULL)
                        {
                                    printf("Error, failed in creating
creating thread = 0.\n");
                                    exit(-1);
                        }
            }

            if (aptr->tables_all || aptr->table_warehouse)
            {
                        fprintf(fLoader, "Starting loader threads for:
warehouse\n");

                        hThread[1] = CreateThread(NULL,

                          0,

                          (LPTHREAD_START_ROUTINE) LoadWarehouse,

                          NULL,

                          0,

                          &dwThreadID[1]);

                        if (hThread[1] == NULL)
                        {
                                    printf("Error, failed in creating
creating thread = 1.\n");
                                    exit(-1);
                        }
            }

            if (aptr->tables_all || aptr->table_customer)
            {
                        fprintf(fLoader, "Starting loader threads for:
customer\n");

                        hThread[2] = CreateThread(NULL,

                          0,

                          (LPTHREAD_START_ROUTINE) LoadCustomer,

                          NULL,

                          0,

                          &dwThreadID[2]);

                        if (hThread[2] == NULL)
                        {
                                    printf("Error, failed in creating
creating main thread = 2.\n");
                                    exit(-1);
                        }
            }

            if (aptr->tables_all || aptr->table_orders)
            {
                        fprintf(fLoader, "Starting loader threads for:
orders\n");

                        hThread[3] = CreateThread(NULL,

                          0,

                          (LPTHREAD_START_ROUTINE) LoadOrders,

                          NULL,

                          0,

                          &dwThreadID[3]);

                        if (hThread[3] == NULL)
                        {
                                    printf("Error, failed in creating
creating main thread = 3.\n");
                                    exit(-1);
                        }
            }

            // Wait for threads to finish...
            for (i=0; i<MAX_MAIN_THREADS; i++)
            {
                        if (hThread[i] != NULL)
                        {
                                    WaitForSingleObject( hThread[i],
INFINITE );
                                    CloseHandle(hThread[i]);
                                    hThread[i] = NULL;
                        }
            }

            main_time_end = (TimeNow() / MILLI);

    sprintf(buffer,"\nTPC-C load completed successfully in %ld minutes.\n",
                                    (main_time_end - main_time_start)/60);

            printf("%s",buffer);
            fprintf(fLoader, "%s", buffer);

            fclose(fLoader);

            SQLFreeEnv(henv);

    exit(0);

            return 0;
}


//==============================================================
//
// Function name: LoadItem
//
//==============================================================

void LoadItem()
{
    long                i_id;
    long                i_im_id;
    char                i_name[I_NAME_LEN+1];
    double              i_price;
    char                i_data[I_DATA_LEN+1];
            char                name[20];
            long                time_start;
            RETCODE             rc;
            DBINT               rcint;
            char                bcphint[128];
            char                err_log_path[256];

            // Seed with unique number
            seed(1);

            printf("Loading item table...\n");

            InitString(i_name, I_NAME_LEN+1);
            InitString(i_data, I_DATA_LEN+1);

            sprintf(name, "%s..%s", aptr->database, "item");

            //rc = bcp_init(i_hdbc1, name, NULL, "logs\\item.err", DB_IN);
            strcpy(err_log_path,aptr->log_path);
            strcat(err_log_path,"item.err");
            rc = bcp_init(i_hdbc1, name, NULL, err_log_path , DB_IN);
            if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

            if ((aptr->build_index == 1) && (aptr->index_order == 1))
            {
                        sprintf(bcphint, "tablock, order (i_id),
ROWS_PER_BATCH = 100000");
                        rc = bcp_control(i_hdbc1, BCPHINTS, (void*) bcphint);
                        if (rc != SUCCEED)
                                    HandleErrorDBC(i_hdbc1);
            }

           rc = bcp_bind(i_hdbc1, (BYTE *) &i_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
            if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

           rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 2);
            if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

            rc = bcp_bind(i_hdbc1, (BYTE *) i_name, 0, I_NAME_LEN, NULL, 0, 0,
3);
            if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

           rc = bcp_bind(i_hdbc1, (BYTE *) &i_price, 0, SQL_VARLEN_DATA, NULL,
0, SQLFLT8, 4);
            if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

            rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0, I_DATA_LEN, NULL, 0, 0,
5);
            if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

            time_start = (TimeNow() / MILLI);

            item_rows_loaded = 0;

            for (i_id = 1; i_id <= max_items; i_id++)
            {
                        i_im_id = RandomNumber(1L, 10000L);

                        MakeAlphaString(14, 24, I_NAME_LEN, i_name);

                        i_price = ((float) RandomNumber(100L, 10000L))/100.0;

                        MakeOriginalAlphaString(26, 50, I_DATA_LEN, i_data,
10);

                        rc = bcp_sendrow(i_hdbc1);
                        if (rc != SUCCEED)
                                    HandleErrorDBC(i_hdbc1);

                        item_rows_loaded++;
                        CheckForCommit(i_hdbc1, i_hstmt1, item_rows_loaded,
"item", &time_start);
            }

            rcint = bcp_done(i_hdbc1);
            if (rcint < 0)
                        HandleErrorDBC(i_hdbc1);

            printf("Finished loading item table.\n");

            SQLFreeStmt(i_hstmt1, SQL_DROP);
            SQLDisconnect(i_hdbc1);
            SQLFreeConnect(i_hdbc1);

            // if build index after load
            if ((aptr->build_index == 1) && (aptr->index_order == 0))
                        BuildIndex("idxitmcl");
}


//==============================================================================
//
// Function   : LoadWarehouse
//
// Loads WAREHOUSE table and loads Stock and District as Warehouses are created
//
//==============================================================================
```

```
void LoadWarehouse()
{
    long      w_id;
    char      w_name[W_NAME_LEN+1];
    char      w_street_1[ADDRESS_LEN+1];
    char      w_street_2[ADDRESS_LEN+1];
    char      w_city[ADDRESS_LEN+1];
    char      w_state[STATE_LEN+1];
    char      w_zip[ZIP_LEN+1];
    double    w_tax;
    double    w_ytd;
              char          name[20];
              long          time_start;
              RETCODE       rc;
              DBINT         rcint;
              char          bcphint[128];
              char          err_log_path[256];

              // Seed with unique number
              seed(2);

              printf("Loading warehouse table...\n");

              InitString(w_name, W_NAME_LEN+1);
              InitAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

              sprintf(name, "%s..%s", aptr->database, "warehouse");

              //rc = bcp_init(w_hdbc1, name, NULL, "logs\\whouse.err", DB_IN);
              strcpy(err_log_path,aptr->log_path);
              strcat(err_log_path,"whouse.err");
              rc = bcp_init(w_hdbc1, name, NULL, err_log_path, DB_IN);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              if ((aptr->build_index == 1) && (aptr->index_order == 1))
              {
                        sprintf(bcphint, "tablock, order (w_id),
ROWS_PER_BATCH = %d", aptr->num_warehouses);
                        rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
                        if (rc != SUCCEED)
                                  HandleErrorDBC(w_hdbc1);
              }

             rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0, W_NAME_LEN, NULL, 0, 0,
2);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) w_street_1, 0, ADDRESS_LEN, NULL,
0, 0, 3);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) w_street_2, 0, ADDRESS_LEN, NULL,
0, 0, 4);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

             rc = bcp_bind(w_hdbc1, (BYTE *) w_city, 0, ADDRESS_LEN, NULL, 0, 0,
5);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) w_state, 0, STATE_LEN, NULL, 0, 0,
6);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) w_zip, 0, ZIP_LEN, NULL, 0, 0, 7);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0, SQL_VARLEN_DATA, NULL,
0, SQLFLT8, 8);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0, SQL_VARLEN_DATA, NULL,
0, SQLFLT8, 9);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              time_start = (TimeNow() / MILLI);

              warehouse_rows_loaded = 0;

              for (w_id = aptr->starting_warehouse; w_id <= aptr-
>num_warehouses; w_id++)
              {
                        MakeAlphaString(6,10, W_NAME_LEN, w_name);

                        MakeAddress(w_street_1, w_street_2, w_city, w_state,
w_zip);

                        w_tax = ((float) RandomNumber(0L,2000L))/10000.00;

                        w_ytd = 300000.00;

                        rc = bcp_sendrow(w_hdbc1);
                        if (rc != SUCCEED)
                                  HandleErrorDBC(w_hdbc1);

                        warehouse_rows_loaded++;
                        CheckForCommit(w_hdbc1, i_hstmt1,
warehouse_rows_loaded, "warehouse", &time_start);
              }

              rcint = bcp_done(w_hdbc1);
              if (rcint < 0)
                        HandleErrorDBC(w_hdbc1);

              printf("Finished loading warehouse table.\n");

              // if build index after load...
              if ((aptr->build_index == 1) && (aptr->index_order == 0))
                        BuildIndex("idxwarcl");

              stock_rows_loaded = 0;
              district_rows_loaded = 0;

              District();
              Stock();

}


//================================================================
//
// Function   : District
//
//================================================================

void District()
{
    short     d_id;
    long      d_w_id;
    char      d_name[D_NAME_LEN+1];
    char      d_street_1[ADDRESS_LEN+1];
    char      d_street_2[ADDRESS_LEN+1];
    char      d_city[ADDRESS_LEN+1];
    char      d_state[STATE_LEN+1];
    char      d_zip[ZIP_LEN+1];
    double    d_tax;
    double    d_ytd;
              char          name[20];
    long      d_next_o_id;
              long          time_start;
              long          w_id;
              RETCODE       rc;
              DBINT         rcint;
              char          bcphint[128];
              char          err_log_path[256];

              // Seed with unique number
              seed(4);

              printf("Loading district table...\n");

              InitString(d_name, D_NAME_LEN+1);
              InitAddress(d_street_1, d_street_2, d_city, d_state, d_zip);
              sprintf(name, "%s..%s", aptr->database, "district");

              //rc = bcp_init(w_hdbc1, name, NULL, "logs\\district.err", DB_IN);
              strcpy(err_log_path,aptr->log_path);
              strcat(err_log_path,"district.err");
              rc = bcp_init(w_hdbc1, name, NULL, err_log_path, DB_IN);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              if ((aptr->build_index == 1) && (aptr->index_order == 1))
              {
                        sprintf(bcphint, "tablock, order (d_w_id, d_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 10));
                        rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
                        if (rc != SUCCEED)
                                  HandleErrorDBC(w_hdbc1);
              }

             rc = bcp_bind(w_hdbc1, (BYTE *) &d_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 1);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) &d_w_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 2);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) d_name, 0, D_NAME_LEN, NULL, 0, 0,
3);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) d_street_1, 0, ADDRESS_LEN, NULL,
0, 0, 4);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) d_street_2, 0, ADDRESS_LEN, NULL,
0, 0, 5);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

             rc = bcp_bind(w_hdbc1, (BYTE *) d_city, 0, ADDRESS_LEN, NULL, 0, 0,
6);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) d_state, 0, STATE_LEN, NULL, 0, 0,
7);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) d_zip, 0, ZIP_LEN, NULL, 0, 0, 8);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) &d_tax, 0, SQL_VARLEN_DATA, NULL,
0, SQLFLT8, 9);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) &d_ytd, 0, SQL_VARLEN_DATA, NULL,
0, SQLFLT8, 10);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              rc = bcp_bind(w_hdbc1, (BYTE *) &d_next_o_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT4, 11);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              d_ytd  = 30000.0;

              d_next_o_id = orders_per_district+1;

              time_start = (TimeNow() / MILLI);

              for (w_id = aptr->starting_warehouse; w_id <= aptr-
>num_warehouses; w_id++)
              {
                        d_w_id = w_id;

                        for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE;
d_id++)
                        {
                                  MakeAlphaString(6,10,D_NAME_LEN,
d_name);

                                  MakeAddress(d_street_1, d_street_2,
d_city, d_state, d_zip);

                                  d_tax = ((float)
RandomNumber(0L,2000L))/10000.00;

                                  rc = bcp_sendrow(w_hdbc1);
                                  if (rc != SUCCEED)
                                            HandleErrorDBC(w_hdbc1);

                                  district_rows_loaded++;
                                  CheckForCommit(w_hdbc1, w_hstmt1,
district_rows_loaded, "district", &time_start);
                        }
    }

              rcint = bcp_done(w_hdbc1);
              if (rcint < 0)
                        HandleErrorDBC(w_hdbc1);

              printf("Finished loading district table.\n");

              // if build index after load...
              if ((aptr->build_index == 1) && (aptr->index_order == 0))
                        BuildIndex("idxdiscl");

    return;
}


//================================================================
//
// Function   : Stock
//
//================================================================

void Stock()
{
    long      s_i_id;
    long      s_w_id;
    short     s_quantity;
    char      s_dist_01[S_DIST_LEN+1];
    char      s_dist_02[S_DIST_LEN+1];
    char      s_dist_03[S_DIST_LEN+1];
    char      s_dist_04[S_DIST_LEN+1];
    char      s_dist_05[S_DIST_LEN+1];
    char      s_dist_06[S_DIST_LEN+1];
    char      s_dist_07[S_DIST_LEN+1];
    char      s_dist_08[S_DIST_LEN+1];
    char      s_dist_09[S_DIST_LEN+1];
    char      s_dist_10[S_DIST_LEN+1];
    long      s_ytd;
    short     s_order_cnt;
    short     s_remote_cnt;
    char      s_data[S_DATA_LEN+1];
    short     len;
              char          name[20];
              long          time_start;
              RETCODE       rc;
              DBINT         rcint;
              char          bcphint[128];
              char          err_log_path[256];

              // Seed with unique number
              seed(3);

              sprintf(name, "%s..%s", aptr->database, "stock");

              //rc = bcp_init(w_hdbc1, name, NULL, "logs\\stock.err", DB_IN);
              strcpy(err_log_path,aptr->log_path);
              strcat(err_log_path,"stock.err");
              rc = bcp_init(w_hdbc1, name, NULL, err_log_path, DB_IN);
              if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

              if ((aptr->build_index == 1) && (aptr->index_order == 1))
              {
                        sprintf(bcphint, "tablock, order (s_i_id, s_w_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 100000));
                        rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
```

```
                        if (rc != SUCCEED)
                                HandleErrorDBC(w_hdbc1);
                }

                rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 1);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 2);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) &s_quantity, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 3);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01, 0, S_DIST_LEN, NULL, 0,
0, 4);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02, 0, S_DIST_LEN, NULL, 0,
0, 5);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03, 0, S_DIST_LEN, NULL, 0,
0, 6);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04, 0, S_DIST_LEN, NULL, 0,
0, 7);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05, 0, S_DIST_LEN, NULL, 0,
0, 8);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06, 0, S_DIST_LEN, NULL, 0,
0, 9);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07, 0, S_DIST_LEN, NULL, 0,
0, 10);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0, S_DIST_LEN, NULL, 0,
0, 11);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09, 0, S_DIST_LEN, NULL, 0,
0, 12);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0, S_DIST_LEN, NULL, 0,
0, 13);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 14);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) &s_order_cnt, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 15);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) &s_remote_cnt, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 16);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0, S_DATA_LEN, NULL, 0, 0,
17);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                s_ytd = s_order_cnt = s_remote_cnt = 0;

                time_start = (TimeNow() / MILLI);

                printf("...Loading stock table\n");

                for (s_i_id=1; s_i_id <= max_items; s_i_id++)
                {

                        for (s_w_id = aptr->starting_warehouse; s_w_id <=
aptr->num_warehouses; s_w_id++)
                        {
                                        s_quantity =
(short)RandomNumber(10L,100L);
                                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_01);
                                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_02);
                                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_03);
                                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_04);
                                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_05);
                                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_06);
                                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_07);
                                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_08);
                                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_09);
                                        len = MakeAlphaString(24,24,S_DIST_LEN,
s_dist_10);

                                        len = MakeOriginalAlphaString(26,50,
S_DATA_LEN, s_data,10);

                                        rc = bcp_sendrow(w_hdbc1);
                                        if (rc != SUCCEED)
                                                HandleErrorDBC(w_hdbc1);

                                        stock_rows_loaded++;
                                        CheckForCommit(w_hdbc1, w_hstmt1,
stock_rows_loaded, "stock", &time_start);

                        }
                }

                rcint = bcp_done(w_hdbc1);
                if (rcint < 0)
                        HandleErrorDBC(w_hdbc1);

                printf("Finished loading stock table.\n");

                SQLFreeStmt(w_hstmt1, SQL_DROP);
                SQLDisconnect(w_hdbc1);
                SQLFreeConnect(w_hdbc1);

                // if build index after load...
                if ((aptr->build_index == 1) && (aptr->index_order == 0))
                        BuildIndex("idxstkcl");

                return;
}


//==================================================================
//
// Function   : LoadCustomer
//
//==================================================================

void LoadCustomer()
{
                LOADER_TIME_STRUCT      customer_time_start;
                LOADER_TIME_STRUCT      history_time_start;
                long                                    w_id;
    short               d_id;
                DWORD
                dwThreadID[MAX_CUSTOMER_THREADS];
                HANDLE
                hThread[MAX_CUSTOMER_THREADS];
                char                                    name[20];
                RETCODE
                rc;
                DBINT
                rcint;
                char
                bcphint[128];
                char
                cmd[256];
                int
                num_procs;
                char
                err_log_path_cust[256];
                char
                err_log_path_hist[256];
                // 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");

                // Initialize bulk copy
                sprintf(name, "%s..%s", aptr->database, "customer");

                //rc = bcp_init(c_hdbc1, name, NULL, "logs\\customer.err", DB_IN);
                strcpy(err_log_path_cust,aptr->log_path);
                strcat(err_log_path_cust,"customer.err");
                rc = bcp_init(c_hdbc1, name, NULL, err_log_path_cust, DB_IN);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                {
                        sprintf(bcphint, "tablock, order (c_w_id, c_d_id,
c_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
                        rc = bcp_control(c_hdbc1, BCPHINTS, (void*) bcphint);
                        if (rc != SUCCEED)
                                HandleErrorDBC(c_hdbc1);
                }

                sprintf(name, "%s..%s", aptr->database, "history");

                rc = bcp_init(c_hdbc2, name, NULL, "logs\\history.err", DB_IN);
                strcpy(err_log_path_hist,aptr->log_path);
                strcat(err_log_path_hist,"history.err");
                rc = bcp_init(c_hdbc2, name, NULL, err_log_path_hist, DB_IN);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

                sprintf(bcphint, "tablock");
                rc = bcp_control(c_hdbc2, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

                customer_rows_loaded    = 0;
                history_rows_loaded     = 0;

                CustomerBufInit();

                customer_time_start.time_start = (TimeNow() / MILLI);
                history_time_start.time_start = (TimeNow() / MILLI);

                for (w_id = aptr->starting_warehouse; w_id <= aptr-
>num_warehouses; w_id++)
                {
                        for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE;
d_id++)
                        {

                                        CustomerBufLoad(d_id, w_id);

                                       // Start parallel loading threads here...

                                        // Start customer table thread

                                        printf("...Loading customer table for:
d_id = %d, w_id = %d\n", d_id, w_id);

                                        hThread[0] = CreateThread(NULL,

                                          0,

                                          (LPTHREAD_START_ROUTINE)
LoadCustomerTable,

                                          &customer_time_start,

                                          0,

                                          &dwThreadID[0]);

                                        if (hThread[0] == NULL)
                                        {
                                                printf("Error, failed in
creating creating thread = 0.\n");
                                                exit(-1);
                                        }

                                        // Start History table thread

                                        printf("...Loading history table for:
d_id = %d, w_id = %d\n", d_id, w_id);

                                        hThread[1] = CreateThread(NULL,

                                          0,

                                          (LPTHREAD_START_ROUTINE)
LoadHistoryTable,

                                          &history_time_start,

                                          0,

                                          &dwThreadID[1]);

                                        if (hThread[1] == NULL)
                                        {
                                                printf("Error, failed in
creating creating thread = 1.\n");
                                                exit(-1);
                                        }

                                        WaitForSingleObject( hThread[0],
INFINITE );
                                        WaitForSingleObject( hThread[1],
INFINITE );

                                        if (CloseHandle(hThread[0]) == FALSE)
                                        {
                                                printf("Error, failed in
closing customer thread handle with errno: %d\n", GetLastError());
                                        }

                                        if (CloseHandle(hThread[1]) == FALSE)
                                        {
                                                printf("Error, failed in
closing history thread handle with errno: %d\n", GetLastError());
                                        }


                        }

                }

                // flush the bulk connection
                rcint = bcp_done(c_hdbc1);
                if (rcint < 0)
                        HandleErrorDBC(c_hdbc1);

                rcint = bcp_done(c_hdbc2);
                if (rcint < 0)
                        HandleErrorDBC(c_hdbc2);

                printf("Finished loading customer table.\n");

                // if build index after load...
```

```
            if ((aptr->build_index == 1) && (aptr->index_order == 0))
            {
                        BuildIndex("idxcuscl");
                        // check the number of processors on this system
                        // if 8 or more processors, then build index on
History.
                        // if less than 8 processors, do not build the index
                        num_procs = atoi(getenv( "NUMBER_OF_PROCESSORS" ));
                        if (num_procs >= 8)
                                    BuildIndex("idxhiscl");
            }

            // build non-clustered index
            if (aptr->build_index == 1)
                        BuildIndex("idxcusnc");

           // Output the NURAND used for the loader into C_FIRST for C_ID = 1,
            // C_W_ID = 1, and C_D_ID = 1
            //sprintf(cmd, "osql -S%s -U%s -P%s -d%s -e -Q\"update customer
set c_first = 'C_LOAD = %d' where c_id = 1 and c_w_id = 1 and c_d_id = 1\" >
logs\\nurand_load.log",
            sprintf(cmd, "osql -S%s -U%s -P%s -d%s -e -Q\"update customer set
c_first = 'C_LOAD = %d' where c_id = 1 and c_w_id = 1 and c_d_id = 1\"
> %snurand_load.log",
                                                aptr->server,
                                                aptr->user,
                                                aptr->password,
                                                aptr->database,
                                                LOADER_NURAND_C,
                                                aptr->log_path);

            system(cmd);

            SQLFreeStmt(c_hstmt1, SQL_DROP);
            SQLDisconnect(c_hdbc1);
            SQLFreeConnect(c_hdbc1);

            SQLFreeStmt(c_hstmt2, SQL_DROP);
            SQLDisconnect(c_hdbc2);
            SQLFreeConnect(c_hdbc2);


    return;

}


//==============================================================
//
// Function   : CustomerBufInit
//
//==============================================================

void CustomerBufInit()
{
            int     i;

            for (i=0;i<customers_per_district;i++)
            {
                        customer_buf[i].c_id = 0;
                        customer_buf[i].c_d_id = 0;
                        customer_buf[i].c_w_id = 0;

                        strcpy(customer_buf[i].c_first,"");
                        strcpy(customer_buf[i].c_middle,"");
                        strcpy(customer_buf[i].c_last,"");
                        strcpy(customer_buf[i].c_street_1,"");
                        strcpy(customer_buf[i].c_street_2,"");
                        strcpy(customer_buf[i].c_city,"");
                        strcpy(customer_buf[i].c_state,"");
                        strcpy(customer_buf[i].c_zip,"");
                        strcpy(customer_buf[i].c_phone,"");
                        strcpy(customer_buf[i].c_credit,"");

                        customer_buf[i].c_credit_lim = 0;
                        customer_buf[i].c_discount = (float) 0;

                        //  fix to avoid ODBC float to numeric conversion
problem.
                        //          customer_buf[i].c_balance = 0;
                        strcpy(customer_buf[i].c_balance,"");

                        customer_buf[i].c_ytd_payment = 0;
                        customer_buf[i].c_payment_cnt = 0;
                        customer_buf[i].c_delivery_cnt = 0;

                        strcpy(customer_buf[i].c_data,"");

                        customer_buf[i].h_amount = 0;

                        strcpy(customer_buf[i].h_data,"");

            }



}


//==============================================================
//
// Function   : CustomerBufLoad
//
// Fills shared buffer for HISTORY and CUSTOMER
//==============================================================

void CustomerBufLoad(int d_id, int w_id)
{
            long                                                i;
    CUSTOMER_SORT_STRUCT    c[CUSTOMERS_PER_DISTRICT];

            for (i=0;i<customers_per_district;i++)
            {
                        if (i < 1000)
                                    LastName(i, c[i].c_last);
                        else

            LastName(NURand(255,0,999,LOADER_NURAND_C), c[i].c_last);

                        MakeAlphaString(8,16,FIRST_NAME_LEN, c[i].c_first);

                        c[i].c_id = i+1;

            }

            printf("...Loading customer buffer for: d_id = %d, w_id = %d\n",
                                    d_id, w_id);

            for (i=0;i<customers_per_district;i++)
            {

                        customer_buf[i].c_d_id = d_id;
                        customer_buf[i].c_w_id = w_id;
                        customer_buf[i].h_amount = 10.0;

                        customer_buf[i].c_ytd_payment  = 10.0;

                        customer_buf[i].c_payment_cnt  = 1;
                        customer_buf[i].c_delivery_cnt = 0;

                        // Generate CUSTOMER and HISTORY data

                        customer_buf[i].c_id = c[i].c_id;

                        strcpy(customer_buf[i].c_first, c[i].c_first);
                        strcpy(customer_buf[i].c_last, c[i].c_last);

                        customer_buf[i].c_middle[0] = 'O';
                        customer_buf[i].c_middle[1] = 'E';

                        MakeAddress(customer_buf[i].c_street_1,
customer_buf[i].c_street_2,
                                                customer_buf[i].c_city,
customer_buf[i].c_state,
customer_buf[i].c_zip);

                        MakeNumberString(16, 16, PHONE_LEN,
customer_buf[i].c_phone);

                        if (RandomNumber(1L, 100L) > 10)
                                    customer_buf[i].c_credit[0] = 'G';
                        else
                                    customer_buf[i].c_credit[0] = 'B';
                        customer_buf[i].c_credit[1] = 'C';

                        customer_buf[i].c_credit_lim = 50000.0;
                        customer_buf[i].c_discount = ((float) RandomNumber(0L,
5000L)) / 10000.0;

                        //  fix to avoid ODBC float to numeric conversion
problem.
                        //  customer_buf[i].c_balance = -10.0;
                        strcpy(customer_buf[i].c_balance,"-10.0");

                        MakeAlphaString(300, 500, C_DATA_LEN,
customer_buf[i].c_data);

                        // Generate HISTORY data
                        MakeAlphaString(12, 24, H_DATA_LEN,
customer_buf[i].h_data);

            }
}

//==============================================================
//
// Function   : LoadCustomerTable
//
//==============================================================

void LoadCustomerTable(LOADER_TIME_STRUCT *customer_time_start)
{
            int         i;
    long        c_id;
    short       c_d_id;
    long        c_w_id;
    char        c_first[FIRST_NAME_LEN+1];
    char        c_middle[MIDDLE_NAME_LEN+1];
    char        c_last[LAST_NAME_LEN+1];
    char        c_street_1[ADDRESS_LEN+1];
    char        c_street_2[ADDRESS_LEN+1];
    char        c_city[ADDRESS_LEN+1];
    char        c_state[STATE_LEN+1];
    char        c_zip[ZIP_LEN+1];
    char        c_phone[PHONE_LEN+1];
    char        c_credit[CREDIT_LEN+1];
    double      c_credit_lim;
    double      c_discount;

            //  fix to avoid ODBC float to numeric conversion problem.

            //  double                  c_balance;
            char                        c_balance[6];

    double      c_ytd_payment;
    short       c_payment_cnt;
    short       c_delivery_cnt;
    char        c_data[C_DATA_LEN+1];
            char                        c_since[C_SINCE_LEN+1];
            RETCODE                     rc;

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
1);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 2);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

            rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 3);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0, FIRST_NAME_LEN, NULL, 0, 0, 4);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0, MIDDLE_NAME_LEN,NULL, 0, 0, 5);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN, NULL, 0, 0, 6);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN, NULL, 0, 0, 7);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN, NULL,0,0, 8);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL, 0, 0, 9);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0, 0, 10);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0, 11);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL, 0, 0, 12);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

            rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0, C_SINCE_LEN, NULL, 0,
SQLCHARACTER, 13);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL, 0, 0, 14);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 15);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 16);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);


            //  fix to avoid ODBC float to numeric conversion problem.

    //  rc = bcp_bind(c_hdbc1, (BYTE *) &c_balance, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 17);
            //  if (rc != SUCCEED)
            //          HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0, SQLCHARACTER, 17);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);


    rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 18);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 19);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt,0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 20);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0, 500, NULL, 0, 0, 21);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

    for (i = 0; i < customers_per_district; i++)
    {
                        c_id = customer_buf[i].c_id;
                        c_d_id = customer_buf[i].c_d_id;
                        c_w_id = customer_buf[i].c_w_id;

                        strcpy(c_first, customer_buf[i].c_first);
                        strcpy(c_middle, customer_buf[i].c_middle);
                        strcpy(c_last, customer_buf[i].c_last);
                        strcpy(c_street_1, customer_buf[i].c_street_1);
```

```
                        strcpy(c_street_2, customer_buf[i].c_street_2);
                        strcpy(c_city, customer_buf[i].c_city);
                        strcpy(c_state, customer_buf[i].c_state);
                        strcpy(c_zip, customer_buf[i].c_zip);
                        strcpy(c_phone, customer_buf[i].c_phone);
                        strcpy(c_credit, customer_buf[i].c_credit);

                        FormatDate(&c_since);


                        c_credit_lim = customer_buf[i].c_credit_lim;
                        c_discount = customer_buf[i].c_discount;

                        //  fix to avoid ODBC float to numeric conversion
problem.
                        //  c_balance = customer_buf[i].c_balance;
                        strcpy(c_balance, customer_buf[i].c_balance);

                        c_ytd_payment = customer_buf[i].c_ytd_payment;
                        c_payment_cnt = customer_buf[i].c_payment_cnt;
                        c_delivery_cnt = customer_buf[i].c_delivery_cnt;

                        strcpy(c_data, customer_buf[i].c_data);

                        // Send data to server
                        rc = bcp_sendrow(c_hdbc1);
                        if (rc != SUCCEED)
                                HandleErrorDBC(c_hdbc1);

                        customer_rows_loaded++;
                        CheckForCommit(c_hdbc1, c_hstmt1,
customer_rows_loaded, "customer", &customer_time_start->time_start);
    }
}


//====================================================================
//
// Function   : LoadHistoryTable
//
//====================================================================

void LoadHistoryTable(LOADER_TIME_STRUCT *history_time_start)
{
            int         i;
    long        c_id;
    short       c_d_id;
    long        c_w_id;
            double      h_amount;
    char        h_data[H_DATA_LEN+1];
            char                    h_date[H_DATE_LEN+1];
            RETCODE                 rc;

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
1);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 2);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 3);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 4);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 5);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

            rc = bcp_bind(c_hdbc2, (BYTE *) &h_date, 0, H_DATE_LEN, NULL, 0,
SQLCHARACTER, 6);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 7);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0, H_DATA_LEN, NULL, 0, 0, 8);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

    for (i = 0; i < customers_per_district; i++)
    {
                        c_id = customer_buf[i].c_id;
                        c_d_id = customer_buf[i].c_d_id;
                        c_w_id = customer_buf[i].c_w_id;
                        h_amount = customer_buf[i].h_amount;
                        strcpy(h_data, customer_buf[i].h_data);

                        FormatDate(&h_date);

                        // send to server
                        rc = bcp_sendrow(c_hdbc2);
                        if (rc != SUCCEED)
                                HandleErrorDBC(c_hdbc2);

                        history_rows_loaded++;
                       CheckForCommit(c_hdbc2, c_hstmt2, history_rows_loaded,
"history", &history_time_start->time_start);
    }
}


//====================================================================
//
// Function   : LoadOrders
//
//====================================================================

void LoadOrders()
{
            LOADER_TIME_STRUCT      orders_time_start;
            LOADER_TIME_STRUCT      new_order_time_start;
            LOADER_TIME_STRUCT      order_line_time_start;
            long                                w_id;
    short               d_id;
            DWORD
            dwThreadID[MAX_ORDER_THREADS];
            HANDLE
            hThread[MAX_ORDER_THREADS];
            char                                name[20];
            RETCODE
            rc;
            char
            bcphint[128];
            char
            err_log_path_ord[256];
            char
            err_log_path_nord[256];
            char
            err_log_path_ordl[256];

            // seed with unique number
            seed(6);

            printf("Loading orders...\n");

            // initialize bulk copy
            sprintf(name, "%s..%s", aptr->database, "orders");

            rc = bcp_init(o_hdbc1, name, NULL, "logs\\orders.err", DB_IN);
            strcpy(err_log_path_ord,aptr->log_path);
            strcat(err_log_path_ord,"orders.err");
            rc = bcp_init(o_hdbc1, name, NULL, err_log_path_ord, DB_IN);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);

            if ((aptr->build_index == 1) && (aptr->index_order == 1))
            {
                        sprintf(bcphint, "tablock, order (o_w_id, o_d_id,
o_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
                        rc = bcp_control(o_hdbc1, BCPHINTS, (void*) bcphint);
                        if (rc != SUCCEED)
                                    HandleErrorDBC(o_hdbc1);
            }

            sprintf(name, "%s..%s", aptr->database, "new_order");

            rc = bcp_init(o_hdbc2, name, NULL, "logs\\neword.err", DB_IN);
            strcpy(err_log_path_nord,aptr->log_path);
            strcat(err_log_path_nord,"neword.err");
            rc = bcp_init(o_hdbc2, name, NULL, err_log_path_nord, DB_IN);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc2);

            if ((aptr->build_index == 1) && (aptr->index_order == 1))
            {
                        sprintf(bcphint, "tablock, order (no_w_id, no_d_id,
no_o_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 9000));
                        rc = bcp_control(o_hdbc2, BCPHINTS, (void*) bcphint);
                        if (rc != SUCCEED)
                                    HandleErrorDBC(o_hdbc2);
            }

            sprintf(name, "%s..%s", aptr->database, "order_line");

            rc = bcp_init(o_hdbc3, name, NULL, "logs\\ordline.err", DB_IN);
            strcpy(err_log_path_ordl,aptr->log_path);
            strcat(err_log_path_ordl,"ordline.err");
            rc = bcp_init(o_hdbc3, name, NULL, err_log_path_ordl, DB_IN);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
            {
                        sprintf(bcphint, "tablock, order (ol_w_id, ol_d_id,
ol_o_id, ol_number), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 300000));
                        rc = bcp_control(o_hdbc3, BCPHINTS, (void*) bcphint);
                        if (rc != SUCCEED)
                                    HandleErrorDBC(o_hdbc3);
            }

            orders_rows_loaded      = 0;
            new_order_rows_loaded   = 0;
            order_line_rows_loaded  = 0;

            OrdersBufInit();

            orders_time_start.time_start = (TimeNow() / MILLI);
            new_order_time_start.time_start = (TimeNow() / MILLI);
            order_line_time_start.time_start = (TimeNow() / MILLI);

            for (w_id = aptr->starting_warehouse; w_id <= aptr-
>num_warehouses; w_id++)
            {
                        for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE;
d_id++)
                        {
                                    OrdersBufLoad(d_id, w_id);

                                    // start parallel loading threads here...
                                    // start Orders table thread
                                    printf("...Loading Order Table for: d_id
= %d, w_id = %d\n", d_id, w_id);

                                    hThread[0] = CreateThread(NULL,
                                      0,
                                      (LPTHREAD_START_ROUTINE)
LoadOrdersTable,
                                      &orders_time_start,
                                      0,
                                      &dwThreadID[0]);
                                    if (hThread[0] == NULL)
                                    {
                                                printf("Error, failed in
creating creating thread = 0.\n");
                                                exit(-1);
                                    }
                                    // start NewOrder table thread
                                    printf("...Loading New-Order Table for:
d_id = %d, w_id = %d\n", d_id, w_id);

                                    hThread[1] = CreateThread(NULL,
                                      0,
                                      (LPTHREAD_START_ROUTINE)
LoadNewOrderTable,
                                      &new_order_time_start,
                                      0,
                                      &dwThreadID[1]);
                                    if (hThread[1] == NULL)
                                    {
                                                printf("Error, failed in
creating creating thread = 1.\n");
                                                exit(-1);
                                    }
                                    // start Order-Line table thread
                                    printf("...Loading Order-Line Table for:
d_id = %d, w_id = %d\n", d_id, w_id);

                                    hThread[2] = CreateThread(NULL,
                                      0,
                                      (LPTHREAD_START_ROUTINE)
LoadOrderLineTable,
                                      &order_line_time_start,
                                      0,
                                      &dwThreadID[2]);
                                    if (hThread[2] == NULL)
                                    {
                                                printf("Error, failed in
creating creating thread = 2.\n");
                                                exit(-1);
                                    }
                                    WaitForSingleObject( hThread[0],
INFINITE );
                                    WaitForSingleObject( hThread[1],
INFINITE );
                                    WaitForSingleObject( hThread[2],
INFINITE );

                                    if (CloseHandle(hThread[0]) == FALSE)
                                    {
                                                printf("Error, failed in
closing Orders thread handle with errno: %d\n", GetLastError());
                                    }
                                    if (CloseHandle(hThread[1]) == FALSE)
                                    {
                                                printf("Error, failed in
closing NewOrder thread handle with errno: %d\n", GetLastError());
                                    }
                                    if (CloseHandle(hThread[2]) == FALSE)
                                    {
                                                printf("Error, failed in
closing OrderLine thread handle with errno: %d\n", GetLastError());
                                    }
                        }
            }
            printf("Finished loading orders.\n");

    return;
}
```

```
//==============================================================
//
// Function   : OrdersBufInit
//
// Clears shared buffer for ORDERS, NEWORDER, and ORDERLINE
//
//==============================================================

void OrdersBufInit()
{
        int     i;
        int                     j;

        for (i=0;i<orders_per_district;i++)
        {
                orders_buf[i].o_id = 0;
                orders_buf[i].o_d_id = 0;
                orders_buf[i].o_w_id = 0;
                orders_buf[i].o_c_id = 0;
                orders_buf[i].o_carrier_id = 0;
                orders_buf[i].o_ol_cnt = 0;
                orders_buf[i].o_all_local = 0;

                for (j=0;j<=14;j++)
                {
                        orders_buf[i].o_ol[j].ol = 0;
                        orders_buf[i].o_ol[j].ol_i_id = 0;
                        orders_buf[i].o_ol[j].ol_supply_w_id =
0;
                        orders_buf[i].o_ol[j].ol_quantity = 0;
                        orders_buf[i].o_ol[j].ol_amount = 0;

        strcpy(orders_buf[i].o_ol[j].ol_dist_info,"");
                }
        }


}


//==============================================================
//
// Function   : OrdersBufLoad
//
// Fills shared buffer for ORDERS, NEWORDER, and ORDERLINE
//
//==============================================================

void OrdersBufLoad(int d_id, int w_id)
{

        int     cust[ORDERS_PER_DISTRICT+1];
        long    o_id;
        short   ol;

        printf("...Loading Order Buffer for: d_id = %d, w_id = %d\n",
                                d_id, w_id);

        GetPermutation(cust, orders_per_district);

        for (o_id=0;o_id<orders_per_district;o_id++)
        {

                // Generate ORDER and NEW-ORDER data

                orders_buf[o_id].o_d_id = d_id;
                orders_buf[o_id].o_w_id = w_id;
                orders_buf[o_id].o_id = o_id+1;
                orders_buf[o_id].o_c_id = cust[o_id+1];
                orders_buf[o_id].o_ol_cnt = (short)RandomNumber(5L,
15L);

                if (o_id < first_new_order)
                {
                        orders_buf[o_id].o_carrier_id =
(short)RandomNumber(1L, 10L);
                        orders_buf[o_id].o_all_local  = 1;
                }
                else
                {
                        orders_buf[o_id].o_carrier_id = 0;
                        orders_buf[o_id].o_all_local  = 1;
                }

                for (ol=0; ol<orders_buf[o_id].o_ol_cnt; ol++)
                {

                        orders_buf[o_id].o_ol[ol].ol = ol+1;
                        orders_buf[o_id].o_ol[ol].ol_i_id =
RandomNumber(1L, max_items);
                        orders_buf[o_id].o_ol[ol].ol_supply_w_id
= w_id;
                        orders_buf[o_id].o_ol[ol].ol_quantity =
5;
                       MakeAlphaString(24, 24, OL_DIST_INFO_LEN,
&orders_buf[o_id].o_ol[ol].ol_dist_info);

                        // Generate ORDER-LINE data
                        if (o_id < first_new_order)
                        {

        orders_buf[o_id].o_ol[ol].ol_amount = 0;
                                                // Added to insure
ol_delivery_d set properly during load

        FormatDate(&orders_buf[o_id].o_ol[ol].ol_delivery_d);


                        }
                        else
                        {

                orders_buf[o_id].o_ol[ol].ol_amount =
RandomNumber(1,999999)/100.0;
                                                // Added to insure
ol_delivery_d set properly during load

                                                // odbc datetime format

                strcpy(orders_buf[o_id].o_ol[ol].ol_delivery_d,"1899-12-31
00:00:00.000");

                        }
                }
        }
}


//==============================================================
//
// Function   : LoadOrdersTable
//
//==============================================================

void LoadOrdersTable(LOADER_TIME_STRUCT *orders_time_start)
{
        int         i;
    long        o_id;
        short       o_d_id;
        long        o_w_id;
    long        o_c_id;
    short       o_carrier_id;
    short       o_ol_cnt;
    short       o_all_local;
        char                    o_entry_d[O_ENTRY_D_LEN+1];
        RETCODE                 rc;
        DBINT                   rcint;

        // bind ORDER data
    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,
SQLINT4, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

       rc = bcp_bind(o_hdbc1, (BYTE *) &o_entry_d, 0, O_ENTRY_D_LEN, NULL,
0, SQLCHARACTER, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 6);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 7);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 8);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    for (i = 0; i < orders_per_district; i++)
    {
                o_id         = orders_buf[i].o_id;
                o_d_id       = orders_buf[i].o_d_id;
                o_w_id       = orders_buf[i].o_w_id;
                o_c_id       = orders_buf[i].o_c_id;
                o_carrier_id = orders_buf[i].o_carrier_id;
                o_ol_cnt     = orders_buf[i].o_ol_cnt;
                o_all_local  = orders_buf[i].o_all_local;


                FormatDate(&o_entry_d);

                // send data to server
                rc = bcp_sendrow(o_hdbc1);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);

                orders_rows_loaded++;
                CheckForCommit(o_hdbc1, o_hstmt1, orders_rows_loaded,
"orders", &orders_time_start->time_start);
    }

        // rcint = bcp_batch(o_hdbc1);
        // if (rcint < 0)
        //      HandleErrorDBC(o_hdbc1);

        if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
        {
                rcint = bcp_done(o_hdbc1);
                if (rcint < 0)
                        HandleErrorDBC(o_hdbc1);

                SQLFreeStmt(o_hstmt1, SQL_DROP);
                SQLDisconnect(o_hdbc1);
                SQLFreeConnect(o_hdbc1);

                // if build index after load...
                if ((aptr->build_index == 1) && (aptr->index_order ==
0))
                        BuildIndex("idxordcl");

                // build non-clustered index
                if (aptr->build_index == 1)
                        BuildIndex("idxordnc");
        }

}


//==============================================================
//
// Function   : LoadNewOrderTable
//
//==============================================================

void LoadNewOrderTable(LOADER_TIME_STRUCT *new_order_time_start)
{
    int         i;
    long        o_id;
    short       o_d_id;
    long        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,
SQLINT4, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

        for (i = first_new_order; i < last_new_order; i++)
        {
                o_id    = orders_buf[i].o_id;
                o_d_id  = orders_buf[i].o_d_id;
                o_w_id  = orders_buf[i].o_w_id;

                rc = bcp_sendrow(o_hdbc2);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc2);

                new_order_rows_loaded++;

                CheckForCommit(o_hdbc2, o_hstmt2,
new_order_rows_loaded, "new_order", &new_order_time_start->time_start);
        }

        // rcint = bcp_batch(o_hdbc2);
        // if (rcint < 0)
        //      HandleErrorDBC(o_hdbc2);

        if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
        {
                rcint = bcp_done(o_hdbc2);
                if (rcint < 0)
                        HandleErrorDBC(o_hdbc2);

                SQLFreeStmt(o_hstmt2, SQL_DROP);
                SQLDisconnect(o_hdbc2);
                SQLFreeConnect(o_hdbc2);

                // if build index after load...
                if ((aptr->build_index == 1) && (aptr->index_order ==
0))
                        BuildIndex("idxnodcl");

        }

}


//==============================================================
//
// Function   : LoadOrderLineTable
//
//==============================================================

void LoadOrderLineTable(LOADER_TIME_STRUCT *order_line_time_start)
{
        int         i,j;
    long        o_id;
        short       o_d_id;
        long        o_w_id;
    long        ol;
        long        ol_i_id;
    long        ol_supply_w_id;
    short       ol_quantity;
    double      ol_amount;
    char        ol_dist_info[DIST_INFO_LEN+1];
        char                    ol_delivery_d[OL_DELIVERY_D_LEN+1];
        RETCODE                 rc;
        DBINT                   rcint;

        // bind ORDER-LINE data
```

```
    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,
SQLINT4, 3);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
4);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 5);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 6);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

            rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0,
OL_DELIVERY_D_LEN, NULL, 0, SQLCHARACTER, 7);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 8);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 9);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0, DIST_INFO_LEN, NULL, 0, 0,
10);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

            for (i = 0; i < orders_per_district; i++)
    {
                        o_id    = orders_buf[i].o_id;
                        o_d_id  = orders_buf[i].o_d_id;
                        o_w_id  = orders_buf[i].o_w_id;

                        for (j=0; j < orders_buf[i].o_ol_cnt; j++)
                        {
                                    ol              =
orders_buf[i].o_ol[j].ol;
                                    ol_i_id         =
orders_buf[i].o_ol[j].ol_i_id;
                                    ol_supply_w_id =
orders_buf[i].o_ol[j].ol_supply_w_id;
                                    ol_quantity     =
orders_buf[i].o_ol[j].ol_quantity;
                                    ol_amount       =
orders_buf[i].o_ol[j].ol_amount;

            strcpy(ol_delivery_d,orders_buf[i].o_ol[j].ol_delivery_d);


            strcpy(ol_dist_info,orders_buf[i].o_ol[j].ol_dist_info);

                                    rc = bcp_sendrow(o_hdbc3);
                                    if (rc != SUCCEED)

                                                HandleErrorDBC(o_hdbc3);


                                    order_line_rows_loaded++;
                                    CheckForCommit(o_hdbc3, o_hstmt3,
order_line_rows_loaded, "order_line", &order_line_time_start->time_start);
                        }
            }

            // rcint = bcp_batch(o_hdbc3);
            // if (rcint < 0)
            //          HandleErrorDBC(o_hdbc3);

            if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
            {
                        rcint = bcp_done(o_hdbc3);
                        if (rcint < 0)
                                    HandleErrorDBC(o_hdbc3);

                        SQLFreeStmt(o_hstmt3, SQL_DROP);
                        SQLDisconnect(o_hdbc3);
                        SQLFreeConnect(o_hdbc3);

                        // if build index after load...
                        if ((aptr->build_index == 1) && (aptr->index_order ==
0))
                                    BuildIndex("idxodlcl");

            }

}

//================================================================
```

```
//
// Function   : GetPermutation
//
//================================================================

void GetPermutation(int perm[], int n)
{
            int i, r, t;

            for (i=1;i<=n;i++)
                        perm[i] = i;

            for (i=1;i<=n;i++)
            {
                        r = RandomNumber(i,n);
                        t = perm[i];
                        perm[i] = perm[r];
                        perm[r] = t;
            }

}


//================================================================
//
// Function   : CheckForCommit
//
//================================================================

void CheckForCommit(HDBC hdbc,                                          HSTMT hstmt,
                                                            int rows_loaded,
                                                                        char
*table_name,
                                                            long *time_start)

{

    long      time_end, time_diff;
            // DBINT      rcint;

    if ( !(rows_loaded % aptr->batch) )
    {

                        // rcint = bcp_batch(hdbc);
                        // if (rcint < 0)
                        //          HandleErrorDBC(hdbc);

                        time_end = (TimeNow() / MILLI);
                        time_diff = time_end - *time_start;

                        printf("-> Loaded %ld rows into %s in %ld sec - Total
= %d (%.2f rps)\n",
                                    aptr->batch,
                                    table_name,
                                                            time_diff,
                                    rows_loaded,
                                                            (float) aptr->batch /
(time_diff ? time_diff : 1L));

                        *time_start = time_end;

    }

    return;
}


//================================================================
//
// Function   : OpenConnections
//
//================================================================

void OpenConnections()
{

            RETCODE         rc;

            char                                    szDriverString[300];
            char                                    szDriverStringOut[1024];
            SQLSMALLINT                 cbDriverStringOut;


            SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv );

            SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3,
0 );

            SQLAllocHandle(SQL_HANDLE_DBC, henv , &i_hdbc1);
            SQLAllocHandle(SQL_HANDLE_DBC, henv , &w_hdbc1);
            SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc1);
            SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc2);
            SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc1);
            SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc2);
            SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc3);

            SQLSetConnectAttr(i_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
            SQLSetConnectAttr(w_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
            SQLSetConnectAttr(c_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
            SQLSetConnectAttr(c_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
            SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
            SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
            SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );

            // Open connections to SQL Server

            // Connection 1

            sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,

aptr->user,

aptr->password,

aptr->database );

            rc = SQLSetConnectOption (i_hdbc1, SQL_PACKET_SIZE, aptr-
>pack_size);
            if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

            rc = SQLDriverConnect ( i_hdbc1,                                                    NULL,

            (SQLCHAR*)&szDriverString[0] ,

            SQL_NTS,

            (SQLCHAR*)&szDriverStringOut[0],

            sizeof(szDriverStringOut),

            &cbDriverStringOut,

            SQL_DRIVER_NOPROMPT );
            if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

            // Connection 2

            sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,

aptr->user,

aptr->password,

aptr->database );

            rc = SQLSetConnectOption (w_hdbc1, SQL_PACKET_SIZE, aptr-
>pack_size);
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            rc = SQLDriverConnect ( w_hdbc1,

            NULL,

            (SQLCHAR*)&szDriverString[0] ,

            SQL_NTS,

            (SQLCHAR*)&szDriverStringOut[0],

            sizeof(szDriverStringOut),

            &cbDriverStringOut,

            SQL_DRIVER_NOPROMPT );
            if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

            // Connection 3

            sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,

aptr->user,

aptr->password,

aptr->database );

            rc = SQLSetConnectOption (c_hdbc1, SQL_PACKET_SIZE, aptr-
>pack_size);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

            rc = SQLDriverConnect ( c_hdbc1,

            NULL,

            (SQLCHAR*)&szDriverString[0] ,

            SQL_NTS,

            (SQLCHAR*)&szDriverStringOut[0],

            sizeof(szDriverStringOut),

            &cbDriverStringOut,

            SQL_DRIVER_NOPROMPT );
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);

            // Connection 4

            sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
```

```
aptr->server,
aptr->user,
aptr->password,
aptr->database );
            rc = SQLSetConnectOption (c_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);
            rc = SQLDriverConnect ( c_hdbc2,
            NULL,
            (SQLCHAR*)&szDriverString[0] ,
            SQL_NTS,
            (SQLCHAR*)&szDriverStringOut[0],
            sizeof(szDriverStringOut),
            &cbDriverStringOut,
            SQL_DRIVER_NOPROMPT );
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);
            // Connection 5
            sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );
            rc = SQLSetConnectOption (o_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);
            rc = SQLDriverConnect ( o_hdbc1,
            NULL,
            (SQLCHAR*)&szDriverString[0] ,
            SQL_NTS,
            (SQLCHAR*)&szDriverStringOut[0],
            sizeof(szDriverStringOut),
            &cbDriverStringOut,
            SQL_DRIVER_NOPROMPT );
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc1);
            // Connection 6
            sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );
            rc = SQLSetConnectOption (o_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc2);
            rc = SQLDriverConnect ( o_hdbc2,
            NULL,
            (SQLCHAR*)&szDriverString[0] ,
            SQL_NTS,
            (SQLCHAR*)&szDriverStringOut[0],
            sizeof(szDriverStringOut),
            &cbDriverStringOut,
            SQL_DRIVER_NOPROMPT );
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc2);
            // Connection 7
            sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );
            rc = SQLSetConnectOption (o_hdbc3, SQL_PACKET_SIZE, aptr->pack_size);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
            rc = SQLDriverConnect ( o_hdbc3,
            NULL,
            (SQLCHAR*)&szDriverString[0] ,
            SQL_NTS,
            (SQLCHAR*)&szDriverStringOut[0],
            sizeof(szDriverStringOut),
            &cbDriverStringOut,
            SQL_DRIVER_NOPROMPT );
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
}
//================================================================
//
// Function name: BuildIndex
//
//================================================================
void BuildIndex(char       *index_script)
{
            char          cmd[256];
            printf("Starting index creation:  %s\n",index_script);
            sprintf(cmd, "osql -S%s -U%s -P%s -e -i%s\\%s.sql > %s%s.log",
                                                aptr->server,
                                                aptr->user,
                                                aptr->password,
                                                aptr->index_script_path,
                                                index_script,
                                                aptr->log_path,
                                                index_script);
            system(cmd);
            printf("Finished index creation:  %s\n",index_script);
}
void HandleErrorDBC (SQLHDBC  hdbc1)
{
            SQLCHAR                   SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
            SQLINTEGER  NativeError;
            SQLSMALLINT i, MsgLen;
            SQLRETURN   rc2;
            char                      timebuf[128];
            char                      datebuf[128];
            char                      err_log_path[256];
            FILE                      *fp1;
            i = 1;
            while (( rc2 = SQLGetDiagRec(SQL_HANDLE_DBC , hdbc1, i, SqlState ,
&NativeError,
                                                Msg, sizeof(Msg) ,
&MsgLen )) != SQL_NO_DATA )
            {
                        sprintf( szLastError , "%s" ,  Msg );
                        _strtime(timebuf);
                        _strdate(datebuf);
                        printf( "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);
                        strcpy(err_log_path,aptr->log_path);
                        strcat(err_log_path,"tpccldr.err");
                        fp1 = fopen(err_log_path,"w");
                        //fp1 = fopen("logs\\tpccldr.err","w");
                        if (fp1 == NULL)
                                    printf("ERROR:  Unable to open errorlog
file.\n");
                        else
                        {
                                    fprintf(fp1, "[%s : %s] %s\n" , datebuf,
timebuf, szLastError);
                                    fclose(fp1);
                        }
                        i++;
            }
}
void HandleErrorSTMT (HSTMT  hstmt1)
{
            SQLCHAR                   SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
            SQLINTEGER  NativeError;
            SQLSMALLINT i, MsgLen;
            SQLRETURN   rc2;
            char                      timebuf[128];
            char                      datebuf[128];
            char                      err_log_path[256];
            FILE                      *fp1;
            i = 1;
            while (( rc2 = SQLGetDiagRec(SQL_HANDLE_STMT , hstmt1, i,
SqlState , &NativeError,
                                                Msg, sizeof(Msg) ,
&MsgLen )) != SQL_NO_DATA )
            {
                        sprintf( szLastError , "%s" ,  Msg );
                        _strtime(timebuf);
                        _strdate(datebuf);
                        printf( "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);
                        strcpy(err_log_path,aptr->log_path);
                        strcat(err_log_path,"tpccldr.err");
                        fp1 = fopen(err_log_path,"w");
                        //fp1 = fopen("logs\\tpccldr.err","w");
                        if (fp1 == NULL)
                                    printf("ERROR:  Unable to open errorlog
file.\n");
                        else
                        {
                                    fprintf(fp1, "[%s : %s] %s\n" , datebuf,
timebuf, szLastError);
                                    fclose(fp1);
                        }
                        i++;
            }
}
void FormatDate ( char* szTimeCOutput )
{
            struct tm when;
            time_t now;
            time( &now );
            when  = *localtime( &now );
            mktime( &when );
            // odbc datetime format
            strftime( szTimeCOutput , 30 , "%Y-%m-%d %H:%M:%S.000", &when );
            return;
}
//================================================================
//
// Function   : CheckDataBase
//
//================================================================
void CheckDataBase()
{
            RETCODE         rc;
            char                                szDriverString[300];
            char                                szDriverStringOut[1024];
            char                                TablesBitMap[9] =
{"000000000"};
            int                                 i, ExitFlag;
            SQLSMALLINT                 cbDriverStringOut;
            SQLCHAR                             TabName[10];
            SQLINTEGER                  TabNameInd, TabCount, TabCountInd;
            ExitFlag = 0;
            SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv );
            SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3,
0 );
            SQLAllocHandle(SQL_HANDLE_DBC, henv , &v_hdbc);
            SQLSetConnectAttr(v_hdbc, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
            // Open connection to SQL Server
            sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );
            rc = SQLSetConnectAttr( v_hdbc, SQL_ATTR_PACKET_SIZE,
(SQLPOINTER)aptr->pack_size, SQL_IS_UINTEGER );
            if (rc != SQL_SUCCESS)
                        HandleErrorDBC(v_hdbc);
            rc = SQLDriverConnect ( v_hdbc,
                                                                        NULL,
```

```
            (SQLCHAR*)&szDriverString[0] ,
            SQL_NTS,
            (SQLCHAR*)&szDriverStringOut[0],
            sizeof(szDriverStringOut),
            &cbDriverStringOut,
            SQL_DRIVER_NOPROMPT );

            // if the rc is SQL_ERROR, the the TPCC database probably does not
exist
            if (rc == SQL_ERROR)
            {
                        printf("The database TPCC does not appear to
exist!\n");
                        printf("\nCheck LOGS\\ directory for database
creation errors.\n");

                        // cleanup database connections and handles
                        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
                        SQLDisconnect(v_hdbc);
                        SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

                        // since there is not a database, exit back to
SETUP.CMD
                        exit(1);
            }

            if ( SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt) !=
SQL_SUCCESS )
                        HandleErrorDBC(v_hdbc);

            if ( SQLBindCol(v_hstmt, 1, SQL_C_ULONG, &TabCount, 0,
&TabCountInd) != SQL_SUCCESS )
                        HandleErrorSTMT(v_hstmt);

            // count the number of user tables from sysobjects
            rc = SQLExecDirect(v_hstmt, "select count(*) from sysobjects where
xtype = \'U\'", SQL_NTS);
            if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
                        HandleErrorSTMT(v_hstmt);

            if ( SQLFetch(v_hstmt) != SQL_SUCCESS )
                        HandleErrorSTMT(v_hstmt);

            // if the number of tables is less than 9, select all the user
tables in TPCC
            if (TabCount != 9)
            {
                        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);

                        SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt);

                        if ( SQLBindCol(v_hstmt, 1, SQL_C_CHAR, &TabName,
sizeof(TabName), &TabNameInd) != SQL_SUCCESS )
                                    HandleErrorSTMT(v_hstmt);

                        // select the list of user tables into a result set
                        rc = SQLExecDirect(v_hstmt, "select * from sysobjects
where xtype = \'U\'", SQL_NTS);
                        if ((rc != SQL_SUCCESS) && (rc !=
SQL_SUCCESS_WITH_INFO))
                                    HandleErrorSTMT(v_hstmt);

                        // go through the result set and set the bitmap for
each found table
                        // set the bitmap to '1' if the table name is found
                        while ((rc = SQLFetch(v_hstmt)) != SQL_NO_DATA)
                        {
                                    switch( TabName[0] )
                                    {
                                    case 'w':
                                                TablesBitMap[0] = '1';
                                                break;
                                    case 'd':
                                                TablesBitMap[1] = '1';
                                                break;
                                    case 'c':
                                                TablesBitMap[2] = '1';
                                                break;
                                    case 'h':
                                                TablesBitMap[3] = '1';
                                                break;
                                    case 'n':
                                                TablesBitMap[4] = '1';
                                                break;
                                    case 'o':
                                                if (TabName[5] = 's')
            TablesBitMap[5] = '1';
                                                if (TabName[5] = '_')
            TablesBitMap[6] = '1';
                                                break;
                                    case 'i':
                                                TablesBitMap[7] = '1';
                                                break;
                                    case 's':
                                                TablesBitMap[8] = '1';
                                                break;
                                    }
                        }

                        // a '0' ExitFlag means do NOT exit the loader early,
a '1' means exit the loader early
                        ExitFlag = 0;

                        // interate through the bitmap to display which
table(s) is actually missing
                        for (i = 0; i <= 8; i++)
                        {
                                    switch(i)
                                    {
                                    case 0:
                                                if (TablesBitMap[i] ==
'0')
                                                {
                                                            printf("The
Warehouse table is missing or damaged.\n");
                                                            ExitFlag = 1;
                                                }
                                                break;
                                    case 1:
                                                if (TablesBitMap[i] ==
'0')
                                                {
                                                            printf("The
District table is missing or damaged.\n");
                                                            ExitFlag = 1;
                                                }
                                                break;
                                    case 2:
                                                if (TablesBitMap[i] ==
'0')
                                                {
                                                            printf("The
Customer table is missing or damaged.\n");
                                                            ExitFlag = 1;
                                                }
                                                break;
                                    case 3:
                                                if (TablesBitMap[i] ==
'0')
                                                {
                                                            printf("The
History table is missing or damaged.\n");
                                                            ExitFlag = 1;
                                                }
                                                break;
                                    case 4:
                                                if (TablesBitMap[i] ==
'0')
                                                {
                                                            printf("The
New_Order table is missing or damaged.\n");
                                                            ExitFlag = 1;
                                                }
                                                break;
                                    case 5:
                                                if (TablesBitMap[i] ==
'0')
                                                {
                                                            printf("The
Orders table is missing or damaged.\n");
                                                            ExitFlag = 1;
                                                }
                                                break;
                                    case 6:
                                                if (TablesBitMap[i] ==
'0')
                                                {
                                                            printf("The
Order_Line table is missing or damaged.\n");
                                                            ExitFlag = 1;
                                                }
                                                break;
                                    case 7:
                                                if (TablesBitMap[i] ==
'0')
                                                {
                                                            printf("The
Item table is missing or damaged.\n");
                                                            ExitFlag = 1;
                                                }
                                                break;
                                    case 8:
                                                if (TablesBitMap[i] ==
'0')
                                                {
                                                            printf("The
Stock table is missing or damaged.\n");
                                                            ExitFlag = 1;
                                                }
                                                break;
                                    }
                        }

                        // if one or more tables are missing, display message
and exit the loader
                        if (ExitFlag = 1)
                        {
                                    printf("\nExiting TPC-C Loader!\n");
                                    printf("\nCheck LOGS\\ directory for
database\n");
                                    printf("or table creation errors.\n");

                                    // cleanup database connections and
handles
                                    SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
                                    SQLDisconnect(v_hdbc);
                                    SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

                                    exit(1);
                        }
            }

            // cleanup database connections and handles
            SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
            SQLDisconnect(v_hdbc);
            SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

                        return;
}
```

Appendix C : Tunable Parameters

RTE input parameter

The following parameters were used with Microsoft BenchCraft RTE..

Profile: 42000w96d31cl96segVIA_3tier442
File Path: C:\BenchCraft\42000w96d31cl96segVIA_3tier442.pro
Version: 3

Number of Engines: 93

Name: DRIVER01
Description:
Directory: \drv01
Machine: rte01
Parameter Set: TPCC
Index: 0
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER15613328
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER02
Description:
Directory: \drv02
Machine: rte02
Parameter Set: TPCC
Index: 10000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER25748171
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER03
Description:
Directory: \drv03
Machine: rte03
Parameter Set: TPCC
Index: 20000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER35788125
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER04
Description:
Directory: \drv04
Machine: rte04
Parameter Set: TPCC
Index: 30000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER45807890
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER05
Description:
Directory: \drv05
Machine: rte05
Parameter Set: TPCC
Index: 40000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER55830703
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER06
Description:
Directory: \drv06
Machine: rte06
Parameter Set: TPCC
Index: 50000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER65843265
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER07
Description:
Directory: \drv07
Machine: rte07
Parameter Set: TPCC
Index: 60000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER75856421
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER08
Description:
Directory: \drv08
Machine: rte08
Parameter Set: TPCC
Index: 70000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER85872031
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER09
Description:
Directory: \drv09
Machine: rte09
Parameter Set: TPCC
Index: 80000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER95885734
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER10
Description:
Directory: \drv10
Machine: rte10
Parameter Set: TPCC
Index: 90000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER105900859
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER11
Description:
Directory: \drv11
Machine: rte11
Parameter Set: TPCC
Index: 100000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER115917906
Connect Rate: 1000

Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER12
Description:
Directory: \drv12
Machine: rte12
Parameter Set: TPCC
Index: 110000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER125975890
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER13
Description:
Directory: \drv13
Machine: rte13
Parameter Set: TPCC
Index: 120000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER135988968
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER14
Description:
Directory: \drv14
Machine: rte14
Parameter Set: TPCC
Index: 130000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER145999875
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER15
Description:
Directory: \drv15
Machine: rte15
Parameter Set: TPCC
Index: 140000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER156012468
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER16
Description:
Directory: \drv16
Machine: rte16
Parameter Set: TPCC
Index: 150000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER166034515
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER17
Description:
Directory: \drv17
Machine: rte17
Parameter Set: TPCC
Index: 160000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER176056890
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER18
Description:
Directory: \drv18
Machine: rte18
Parameter Set: TPCC
Index: 170000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER186067421
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER19
Description:
Directory: \drv19
Machine: rte19
Parameter Set: TPCC
Index: 180000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER196077875
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER20
Description:
Directory: \drv20
Machine: rte20
Parameter Set: TPCC
Index: 190000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER206089687
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER21
Description:
Directory: \drv21
Machine: rte21
Parameter Set: TPCC
Index: 200000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER216105171
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER22
Description:
Directory: \drv22
Machine: rte22
Parameter Set: TPCC
Index: 210000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER226121546
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER23
Description:
Directory: \drv23
Machine: rte23

Parameter Set: TPCC
Index: 220000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER236221312
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER24
Description:
Directory: \drv24
Machine: rte24
Parameter Set: TPCC
Index: 230000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER24946781
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER25
Description:
Directory: \drv25
Machine: rte25
Parameter Set: TPCC
Index: 240000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER251024640
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER26
Description:
Directory: \drv26
Machine: rte26
Parameter Set: TPCC
Index: 250000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER261066781
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER27
Description:
Directory: \drv27
Machine: rte27
Parameter Set: TPCC
Index: 260000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER271128296
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER28
Description:
Directory: \drv28
Machine: rte28
Parameter Set: TPCC
Index: 270000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER281180234
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER29
Description:
Directory: \drv29
Machine: rte29
Parameter Set: TPCC
Index: 280000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER291209718
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER30
Description:
Directory: \drv30
Machine: rte30
Parameter Set: TPCC
Index: 290000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER301242343
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER31
Description:
Directory: \drv31
Machine: rte31
Parameter Set: TPCC
Index: 300000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER311320234
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER33
Description:
Directory: \drv012
Machine: rte01
Parameter Set: TPCC
Index: 320000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER331410390
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER34
Description:
Directory: \drv022
Machine: rte02
Parameter Set: TPCC
Index: 330000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER341437859
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER35
Description:
Directory: \drv032
Machine: rte03
Parameter Set: TPCC
Index: 340000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER351467500
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233

CPU: 1

Name: DRIVER36
Description:
Directory: \drv042
Machine: rte04
Parameter Set: TPCC
Index: 350000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER361496468
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER37
Description:
Directory: \drv052
Machine: rte05
Parameter Set: TPCC
Index: 360000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER371522218
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER38
Description:
Directory: \drv062
Machine: rte06
Parameter Set: TPCC
Index: 370000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER381548062
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER39
Description:
Directory: \drv072
Machine: rte07
Parameter Set: TPCC
Index: 380000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER391579906
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER40
Description:
Directory: \drv082
Machine: rte08
Parameter Set: TPCC
Index: 390000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER401606109
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER41
Description:
Directory: \drv092
Machine: rte09
Parameter Set: TPCC
Index: 400000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER411631687
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER42
Description:
Directory: \drv102
Machine: rte10
Parameter Set: TPCC
Index: 410000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER421651875
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER43
Description:
Directory: \drv112
Machine: rte11
Parameter Set: TPCC
Index: 420000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER431686984
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER44
Description:
Directory: \drv122
Machine: rte12
Parameter Set: TPCC
Index: 430000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER441708750
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER45
Description:
Directory: \drv132
Machine: rte13
Parameter Set: TPCC
Index: 440000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER451733937
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER46
Description:
Directory: \drv142
Machine: rte14
Parameter Set: TPCC
Index: 450000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER461768046
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER47
Description:
Directory: \drv152
Machine: rte15
Parameter Set: TPCC
Index: 460000000
Seed: 1423
Configured Users: 4520

Pipe Name: DRIVER4713701375
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER48
Description:
Directory: \drv162
Machine: rte16
Parameter Set: TPCC
Index: 470000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER4813767953
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER49
Description:
Directory: \drv172
Machine: rte17
Parameter Set: TPCC
Index: 480000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER4913797062
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER50
Description:
Directory: \drv182
Machine: rte18
Parameter Set: TPCC
Index: 490000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER5013818890
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER51
Description:
Directory: \drv192
Machine: rte19
Parameter Set: TPCC
Index: 500000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER5113843562
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER52
Description:
Directory: \drv202
Machine: rte20
Parameter Set: TPCC
Index: 510000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER5213863718
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER53
Description:
Directory: \drv212
Machine: rte21
Parameter Set: TPCC
Index: 520000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER5313890265
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER54
Description:
Directory: \drv222
Machine: rte22
Parameter Set: TPCC
Index: 530000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER5413908312
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER55
Description:
Directory: \drv232
Machine: rte23
Parameter Set: TPCC
Index: 540000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER5513924546
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER56
Description:
Directory: \drv242
Machine: rte24
Parameter Set: TPCC
Index: 550000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER5613943203
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER57
Description:
Directory: \drv252
Machine: rte25
Parameter Set: TPCC
Index: 560000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER5713963031
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER58
Description:
Directory: \drv262
Machine: rte26
Parameter Set: TPCC
Index: 570000000
Seed: 1423
Configured Users: 4520
Pipe Name: DRIVER5813981406
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER59
Description:

Directory: \drv272
Machine: rte27
Parameter Set: TPCC
Index: 580000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER5913994921
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER60
Description:
Directory: \drv282
Machine: rte28
Parameter Set: TPCC
Index: 590000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER6014010234
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER61
Description:
Directory: \drv292
Machine: rte29
Parameter Set: TPCC
Index: 600000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER6114029109
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER62
Description:
Directory: \drv302
Machine: rte30
Parameter Set: TPCC
Index: 610000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER6214050156
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER63
Description:
Directory: \drv312
Machine: rte31
Parameter Set: TPCC
Index: 620000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER6314065546
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER65
Description:
Directory: \drv013
Machine: rte01
Parameter Set: TPCC
Index: 640000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER6514096437
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER66
Description:
Directory: \drv023
Machine: rte02
Parameter Set: TPCC
Index: 650000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER6614110046
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER67
Description:
Directory: \drv033
Machine: rte03
Parameter Set: TPCC
Index: 660000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER6714125062
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER68
Description:
Directory: \drv043
Machine: rte04
Parameter Set: TPCC
Index: 670000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER6814148140
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER69
Description:
Directory: \drv053
Machine: rte05
Parameter Set: TPCC
Index: 680000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER6914161640
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER70
Description:
Directory: \drv063
Machine: rte06
Parameter Set: TPCC
Index: 690000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER70290512640
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER71
Description:
Directory: \drv073
Machine: rte07
Parameter Set: TPCC
Index: 700000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER71290626421
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1

Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER72
Description:
Directory: \drv083
Machine: rte08
Parameter Set: TPCC
Index: 710000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER72290740359
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER73
Description:
Directory: \drv093
Machine: rte09
Parameter Set: TPCC
Index: 720000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER731488449468
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER74
Description:
Directory: \drv103
Machine: rte10
Parameter Set: TPCC
Index: 730000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER741488487703
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER75
Description:
Directory: \drv113
Machine: rte11
Parameter Set: TPCC
Index: 740000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER751488545640
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER76
Description:
Directory: \drv123
Machine: rte12
Parameter Set: TPCC
Index: 750000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER761488595437
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER77
Description:
Directory: \drv133
Machine: rte13
Parameter Set: TPCC
Index: 760000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER771488620203
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER78
Description:
Directory: \drv143
Machine: rte14
Parameter Set: TPCC
Index: 770000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER781488640812
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER79
Description:
Directory: \drv153
Machine: rte15
Parameter Set: TPCC
Index: 780000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER791488715875
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER80
Description:
Directory: \drv163
Machine: rte16
Parameter Set: TPCC
Index: 790000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER801488754562
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER81
Description:
Directory: \drv173
Machine: rte17
Parameter Set: TPCC
Index: 800000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER811488783031
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER82
Description:
Directory: \drv183
Machine: rte18
Parameter Set: TPCC
Index: 810000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER821488905375
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER83
Description:
Directory: \drv193
Machine: rte19
Parameter Set: TPCC
Index: 820000000

Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER831488931828
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER84
Description:
Directory: \drv203
Machine: rte20
Parameter Set: TPCC
Index: 830000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER841488962687
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER85
Description:
Directory: \drv213
Machine: rte21
Parameter Set: TPCC
Index: 840000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER851489004500
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER86
Description:
Directory: \drv223
Machine: rte22
Parameter Set: TPCC
Index: 850000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER861489029921
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER87
Description:
Directory: \drv233
Machine: rte23
Parameter Set: TPCC
Index: 860000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER871489054218
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER88
Description:
Directory: \drv244
Machine: rte24
Parameter Set: TPCC
Index: 870000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER8889436296
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER89
Description:
Directory: \drv253
Machine: rte25
Parameter Set: TPCC
Index: 880000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER8989488765
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER90
Description:
Directory: \drv263
Machine: rte26
Parameter Set: TPCC
Index: 890000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER9089514468
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER91
Description:
Directory: \drv273
Machine: rte27
Parameter Set: TPCC
Index: 900000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER9189550140
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER92
Description:
Directory: \drv283
Machine: rte28
Parameter Set: TPCC
Index: 910000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER9289589828
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER93
Description:
Directory: \drv293
Machine: rte29
Parameter Set: TPCC
Index: 920000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER9389607781
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER94
Description:
Directory: \drv303
Machine: rte30
Parameter Set: TPCC
Index: 930000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER9489634390
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER95
Description:
Directory: \drv313
Machine: rte31
Parameter Set: TPCC
Index: 940000000
Seed: 1423
Configured Users: 4510
Pipe Name: DRIVER9589674656
Connect Rate: 1000
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0

Number of User groups: 93

Driver Engine: DRIVER01
IIS Server: acl01
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1 - 452
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER02
IIS Server: acl02
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 453 - 904
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER03
IIS Server: acl03
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 905 - 1356
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER04
IIS Server: acl04
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1357 - 1808
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER05
IIS Server: acl05
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1809 - 2260
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER06
IIS Server: acl06
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2261 - 2712
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER07
IIS Server: acl07
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2713 - 3164
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER08
IIS Server: acl08
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3165 - 3616
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER09
IIS Server: acl09
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3617 - 4068
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER10
IIS Server: acl10
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4069 - 4520
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER11
IIS Server: acl11
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4521 - 4972
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER12
IIS Server: acl12
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4973 - 5424
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER13
IIS Server: acl13
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5425 - 5876
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER14
IIS Server: acl14
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5877 - 6328
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER15
IIS Server: acl15
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 6329 - 6780
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER16
IIS Server: acl16
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 6781 - 7232
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER17
IIS Server: acl17
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 7233 - 7684
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER18
IIS Server: acl18
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 7685 - 8136
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER19
IIS Server: acl19
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 8137 - 8588
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER20
IIS Server: acl20
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 8589 - 9040
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER21
IIS Server: acl21
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 9041 - 9492
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER22
IIS Server: acl22
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 9493 - 9944
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER23
IIS Server: acl23
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 9945 - 10396
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER24
IIS Server: acl24
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 10397 - 10848
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER25
IIS Server: acl25
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 10849 - 11300
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER26
IIS Server: acl26
SQL Server: asama
Database: tpcc

User: sa
Protocol: HTML
w_id Range: 11301 - 11752
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER27
IIS Server: acl27
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 11753 - 12204
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER28
IIS Server: acl28
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 12205 - 12656
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER29
IIS Server: acl29
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 12657 - 13108
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER30
IIS Server: acl30
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 13109 - 13560
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER31
IIS Server: acl31
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 13561 - 14012
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER33
IIS Server: acl012
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 14013 - 14464
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER34
IIS Server: acl022
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 14465 - 14916
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER35
IIS Server: acl032
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 14917 - 15368
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER36
IIS Server: acl042
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 15369 - 15820
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER37
IIS Server: acl052
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 15821 - 16272
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER38
IIS Server: acl062
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 16273 - 16724
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER39
IIS Server: acl072
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 16725 - 17176
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER40
IIS Server: acl082
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 17177 - 17628
w_id Min Warehouse: 1

w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER41
IIS Server: acl092
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 17629 - 18080
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER42
IIS Server: acl102
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 18081 - 18532
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER43
IIS Server: acl112
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 18533 - 18984
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER44
IIS Server: acl122
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 18985 - 19436
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER45
IIS Server: acl132
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 19437 - 19888
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER46
IIS Server: acl142
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 19889 - 20340
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER47
IIS Server: acl152
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 20341 - 20792
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER48
IIS Server: acl162
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 20793 - 21244
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER49
IIS Server: acl172
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 21245 - 21696
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER50
IIS Server: acl182
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 21697 - 22148
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER51
IIS Server: acl192
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 22149 - 22600
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER52
IIS Server: acl202
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 22601 - 23052
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER53
IIS Server: acl212
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 23053 - 23504
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1

Scale Down: No

Driver Engine: DRIVER54
IIS Server: acl222
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 23505 - 23956
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER55
IIS Server: acl232
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 23957 - 24408
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER56
IIS Server: acl242
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 24409 - 24860
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER57
IIS Server: acl252
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 24861 - 25312
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER58
IIS Server: acl262
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 25313 - 25764
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4520
District id: 1
Scale Down: No

Driver Engine: DRIVER59
IIS Server: acl272
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 25765 - 26215
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER60
IIS Server: acl282
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 26216 - 26666
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER61
IIS Server: acl292
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 26667 - 27117
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER62
IIS Server: acl302
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 27118 - 27568
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER63
IIS Server: acl312
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 27569 - 28019
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER65
IIS Server: acl013
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 28020 - 28470
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER66
IIS Server: acl023
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 28471 - 28921
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER67
IIS Server: acl033
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 28922 - 29372
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER68
IIS Server: acl043

SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 29373 - 29823
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER69
IIS Server: acl053
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 29824 - 30274
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER70
IIS Server: acl063
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 30275 - 30725
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER71
IIS Server: acl073
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 30726 - 31176
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER72
IIS Server: acl083
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 31177 - 31627
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER73
IIS Server: acl093
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 31628 - 32078
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER74
IIS Server: acl103
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 32079 - 32529
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER75
IIS Server: acl113
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 32530 - 32980
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER76
IIS Server: acl123
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 32981 - 33431
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER77
IIS Server: acl133
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 33432 - 33882
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER78
IIS Server: acl143
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 33883 - 34333
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER79
IIS Server: acl153
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 34334 - 34784
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER80
IIS Server: acl163
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 34785 - 35235
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER81
IIS Server: acl173
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML

w_id Range: 35236 - 35686
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER82
IIS Server: acl183
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 35687 - 36137
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER83
IIS Server: acl193
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 36138 - 36588
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER84
IIS Server: acl203
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 36589 - 37039
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER85
IIS Server: acl213
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 37040 - 37490
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER86
IIS Server: acl223
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 37491 - 37941
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER87
IIS Server: acl233
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 37942 - 38392
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER88
IIS Server: acl243
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 38393 - 38843
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER89
IIS Server: acl253
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 38844 - 39294
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER90
IIS Server: acl263
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 39295 - 39745
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER91
IIS Server: acl273
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 39746 - 40196
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER92
IIS Server: acl283
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 40197 - 40647
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER93
IIS Server: acl293
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 40648 - 41098
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER94
IIS Server: acl303
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 41099 - 41549
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal

User Count: 4510
District id: 1
Scale Down: No

Driver Engine: DRIVER95
IIS Server: acl313
SQL Server: asama
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 41550 - 42000
w_id Min Warehouse: 1
w_id Max Warehouse: 42000
Scale: Normal
User Count: 4510
District id: 1
Scale Down: No

Number of Parameter Sets: 3

~Default
Default Parameter Set

	Txn Weight	Think Time	Key Time	RT Delay	RT Menu	Fence	Delay
New Order	10.00	12.05	18.01	0.10	0.10	5.00	
Payment	10.00	12.05	3.01	0.10	0.10	5.00	
Delivery	1.00	5.05	2.01	0.10	0.10	5.00	
Stock Level	1.00	5.05	2.01	0.10	0.10	20.00	
Order Status	1.00	10.05	2.01	0.10	0.10	5.00	

batch

	Txn Weight	Think Time	Key Time	RT Delay	RT Menu	Fence	Delay
New Order	44.92	0.00	0.00	0.00	0.00	0.00	
Payment	43.02	0.00	0.00	0.00	0.00	0.00	
Delivery	4.02	0.00	0.00	0.00	0.00	0.00	
Stock Level	4.02	0.00	0.00	0.00	0.00	0.00	
Order Status	4.02	0.00	0.00	0.00	0.00	0.00	

TPCC

	Txn Weight	Think Time	Key Time	RT Delay	RT Menu	Fence	Delay
New Order	44.92	12.07	18.01	0.10	0.10	5.00	
Payment	43.01	12.07	3.01	0.10	0.10	5.00	
Delivery	4.02	5.07	2.01	0.10	0.10	5.00	
Stock Level	4.02	5.07	2.01	0.10	0.10	20.00	
Order Status	4.03	10.07	2.01	0.10	0.10	5.00	

<Server Configuration>

Services

Following services were activated during the measurement.

Automatic Updates
COM+ Event System
Computer Browser
Cryptographic Services
DHCP Client
Distributed File System
Distributed Link Tracking Client
Distributed Transaction Coordinator
DNS Client
Error Reporting Service
Event Log
Help and Support
IPSEC Services
Logical Disk Manager
Microsoft Search
Network Connections
Network Location Awareness (NLA)
NT LM Security Support Provider
Plug and Play
Print Spooler
Protected Storage
Remote Procedure Call (RPC)
Remote Registry
Secondary Logon
Security Accounts Manager
Server
Shell Hardware Detection
System Event Notification
Task Scheduler
TCP/IP NetBIOS Helper
Terminal Services
Windows Management Instrumentation
Windows Time
Wireless Configuration
Workstation

Network Adapter Tunings

We bind each client/server connection (i.e., 8 QLA2350 VIA NICs)
onto the distinct CPU sets exclusively.
To do this, apply the following settings by using the SQLServer's "Server Network Utility".

Enabled protocols: "VIA" only
Vendor: Select "QLogic"
Listen Info: Enter the following string
"0:1433[0xf],1:1433[0xf0],2:1433[0xf00],3:1433[0xf000],4:1433[0xf0000],5:1433[0xf00000],6:1433[0x7f000000],7:1433[0x80000000]"

Registry Tunings

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\I/O System]
"CountOperations"=dword:00000000

Boot loader settings

OsLoadOptions = /MAXPROCSPERCLUSTER=1 /CONFIGFLAG=8224

CONFIGFLAG=8224 : Interrupt logging disable
MAXPROCSPERCLUSTER=1 : distribute interrupts to all CPUs equally.

User Rights Assignment

The Group Policy Editor was used to modify an entry under "Computer Configuration" -> "Windows Settings" -> "Security Settings"
-> "Local Policies" -> "User Rights Assignment".

The right of "Lock pages in memory" was assigned to the Administrator so
that SQL Server could use large amounts of physical memory.

System Information

System Information report written at: 04/20/03 17:02:15
System Name: ASAMA
[System Summary]

Item	Value
OS Name	Microsoft(R) Windows(R) .NET Server 2003, Datacenter Edition
Version	5.2.3718 Build 3718
OS Manufacturer	Microsoft Corporation
System Name	ASAMA
System Manufacturer	NEC
System Model	server u32000
System Type	Itanium (TM) -based System
Processor	ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1500 Mhz

SMBIOS Version 2.3
Windows Directory C:\WINDOWS
System Directory C:\WINDOWS\system32
Boot Device \Device\HarddiskVolume1
Locale United States
Hardware Abstraction Layer Version = "5.2.3718.0 (dnsrv.021114-1947)"
User Name ASAMA\Administrator
Time Zone Tokyo Standard Time
Total Physical Memory 32.12 MB
Available Physical Memory 507.00 GB
Total Virtual Memory 1,021.54 GB
Available Virtual Memory 1,016.04 GB
Page File Space 509.57 GB
Page File C:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

Resource Device
I/O Port 0x00001F00-0x00001FFF PCI bus
I/O Port 0x00001F00-0x00001FFF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00000000-0x000003AF PCI bus
I/O Port 0x00000000-0x000003AF Direct memory access controller

I/O Port 0x00002600-0x000026FF PCI bus
I/O Port 0x00002600-0x000026FF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00001B00-0x00001BFF PCI bus
I/O Port 0x00001B00-0x00001BFF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x000003C0-0x000003DF PCI bus
I/O Port 0x000003C0-0x000003DF Standard VGA Graphics Adapter

I/O Port 0x00002200-0x000022FF PCI bus
I/O Port 0x00002200-0x000022FF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00002000-0x000020FF PCI bus
I/O Port 0x00002000-0x000020FF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00001700-0x000017FF PCI bus
I/O Port 0x00001700-0x000017FF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00001300-0x000014FF PCI bus
I/O Port 0x00001300-0x000014FF Adaptec SCSI Card 39160 - Ultra160 SCSI

I/O Port 0x00001A00-0x00001AFF PCI bus
I/O Port 0x00001A00-0x00001AFF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00002100-0x000021FF PCI bus
I/O Port 0x00002100-0x000021FF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00001600-0x000016FF PCI bus
I/O Port 0x00001600-0x000016FF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00001D00-0x00001DFF PCI bus
I/O Port 0x00001D00-0x00001DFF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00002400-0x000024FF PCI bus
I/O Port 0x00002400-0x000024FF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00001900-0x000019FF PCI bus
I/O Port 0x00001900-0x000019FF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00002800-0x0000FFFF PCI bus
I/O Port 0x00002800-0x0000FFFF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00001500-0x000015FF PCI bus
I/O Port 0x00001500-0x000015FF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00002700-0x000027FF PCI bus
I/O Port 0x00002700-0x000027FF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00001C00-0x00001CFF PCI bus
I/O Port 0x00001C00-0x00001CFF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00002300-0x000023FF PCI bus
I/O Port 0x00002300-0x000023FF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00001800-0x000018FF PCI bus
I/O Port 0x00001800-0x000018FF QLogic QLA23xx PCI Fibre Channel Adapter

Memory Address 0xA0000-0xBFFFF PCI bus
Memory Address 0xA0000-0xBFFFF Standard VGA Graphics Adapter

I/O Port 0x00001200-0x000012FF PCI bus
I/O Port 0x00001200-0x000012FF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x000003B0-0x000003BB PCI bus
I/O Port 0x000003B0-0x000003BB Standard VGA Graphics Adapter

I/O Port 0x00001E00-0x00001EFF PCI bus
I/O Port 0x00001E00-0x00001EFF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00002500-0x000025FF PCI bus
I/O Port 0x00002500-0x000025FF QLogic QLA23xx PCI Fibre Channel Adapter

I/O Port 0x00000D00-0x000011FF PCI bus
I/O Port 0x00000D00-0x000011FF Extended IO Bus

[DMA]

Resource Device Status
Channel 4 Direct memory access controller OK

[Forced Hardware]

Device PNP Device ID

[I/O]

Resource Device Status
0x00000000-0x000003AF PCI bus OK
0x00000000-0x000003AF Direct memory access controller OK
0x000003B0-0x000003BB PCI bus OK
0x000003B0-0x000003BB Standard VGA Graphics Adapter OK
0x000003BC-0x000003BF PCI bus OK
0x000003C0-0x000003DF PCI bus OK
0x000003C0-0x000003DF Standard VGA Graphics Adapter OK
0x000003E0-0x00000CF7 PCI bus OK
0x00000D00-0x000011FF PCI bus OK
0x00000D00-0x000011FF Extended IO Bus OK
0x00000080-0x0000008F Direct memory access controller OK
0x000000C0-0x000000DF Direct memory access controller OK
0x00000020-0x00000021 Programmable interrupt controller OK
0x000000A0-0x000000A1 Programmable interrupt controller OK
0x00000040-0x00000043 System timer OK
0x00000061-0x00000061 System speaker OK
0x00000070-0x00000071 System CMOS/real time clock OK
0x000000F0-0x000000FF Numeric data processor OK
0x000000B2-0x000000B3 Generic Bus OK
0x00000E00-0x00000E3F Generic Bus OK
0x00000092-0x00000092 Motherboard resources OK
0x000004D0-0x000004D1 Motherboard resources OK
0x00000D80-0x00000D80 Motherboard resources OK
0x00000D81-0x00000D81 Motherboard resources OK
0x00000CA0-0x00000CA1 Motherboard resources OK
0x00000CA4-0x00000CA7 Motherboard resources OK
0x0000002E-0x0000002F Motherboard resources OK
0x00000060-0x00000060 Standard 101/102-Key or

Microsoft Natural PS/2 Keyboard OK
0x00000064-0x00000064 Standard 101/102-Key or
Microsoft Natural PS/2 Keyboard OK
0x000003F8-0x000003FF Communications Port (COM1)
OK
0x00000C00-0x00000C03 Extended IO Bus OK
0x00000F00-0x00000F7F Extended IO Bus OK
0x00000F80-0x00000FFF Extended IO Bus OK
0x00000CA2-0x00000CA2 OK
0x00000CA3-0x00000CA3 OK
0x00001120-0x0000112F Intel(r) IA64 Bus Master IDE
Controller OK
0x000001F0-0x000001F7 Primary IDE Channel OK
0x000003F6-0x000003F6 Primary IDE Channel OK
0x00001100-0x0000111F Intel(r) 82372FB PCI to USB
Universal Host Controller OK
0x00001000-0x000010FF Standard VGA Graphics Adapter
OK
0x000002F8-0x000002FF Communications Port (COM2)
OK
0x00001200-0x000012FF PCI bus OK
0x00001200-0x000012FF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00001300-0x000014FF PCI bus OK
0x00001300-0x000014FF Adaptec SCSI Card 39160 -
Ultra160 SCSI OK
0x00001400-0x000014FF Adaptec SCSI Card 39160 -
Ultra160 SCSI OK
0x00001500-0x000015FF PCI bus OK
0x00001500-0x000015FF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00001600-0x000016FF PCI bus OK
0x00001600-0x000016FF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00001700-0x000017FF PCI bus OK
0x00001700-0x000017FF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00001800-0x000018FF PCI bus OK
0x00001800-0x000018FF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00001900-0x000019FF PCI bus OK
0x00001900-0x000019FF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00001A00-0x00001AFF PCI bus OK
0x00001A00-0x00001AFF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00001B00-0x00001BFF PCI bus OK
0x00001B00-0x00001BFF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00001C00-0x00001CFF PCI bus OK
0x00001C00-0x00001CFF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00001D00-0x00001DFF PCI bus OK
0x00001D00-0x00001DFF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00001E00-0x00001EFF PCI bus OK
0x00001E00-0x00001EFF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00001F00-0x00001FFF PCI bus OK
0x00001F00-0x00001FFF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00002000-0x000020FF PCI bus OK
0x00002000-0x000020FF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00002100-0x000021FF PCI bus OK
0x00002100-0x000021FF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00002200-0x000022FF PCI bus OK
0x00002200-0x000022FF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00002300-0x000023FF PCI bus OK
0x00002300-0x000023FF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00002400-0x000024FF PCI bus OK
0x00002400-0x000024FF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00002500-0x000025FF PCI bus OK
0x00002500-0x000025FF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00002600-0x000026FF PCI bus OK
0x00002600-0x000026FF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00002700-0x000027FF PCI bus OK
0x00002700-0x000027FF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0x00002800-0x0000FFFF PCI bus OK
0x00002800-0x0000FFFF QLogic QLA23xx PCI Fibre
Channel Adapter OK

[IRQs]

Resource Device Status
IRQ 9 Microsoft ACPI-Compliant System OK
IRQ 0 System timer OK
IRQ 8 System CMOS/real time clock OK
IRQ 13 Numeric data processor OK
IRQ 1 Standard 101/102-Key or Microsoft Natural PS/2
Keyboard OK
IRQ 12 PS/2 Compatible Mouse OK
IRQ 4 Communications Port (COM1) OK
IRQ 11 OK
IRQ 14 Primary IDE Channel OK
IRQ 19 Intel(r) 82372FB PCI to USB Universal Host Controller
OK
IRQ 245 Communications Port (COM2) OK
IRQ 64 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 68 Adaptec SCSI Card 39160 - Ultra160 SCSI OK
IRQ 67 Adaptec SCSI Card 39160 - Ultra160 SCSI OK
IRQ 69 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 73 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 78 Intel(R) PRO/1000 F Server Adapter OK
IRQ 84 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 86 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 90 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 91 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 95 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 100 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 150 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 152 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 156 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 157 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 161 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 166 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 172 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 174 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 178 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 179 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 183 QLogic QLA23xx PCI Fibre Channel Adapter OK
IRQ 188 QLogic QLA23xx PCI Fibre Channel Adapter OK

[Memory]

Resource Device Status
0xA0000-0xBFFFF PCI bus OK
0xA0000-0xBFFFF Standard VGA Graphics Adapter OK
0xC0000-0xC3FFF PCI bus OK
0xC4000-0xC7FFF PCI bus OK
0xC8000-0xCBFFF PCI bus OK
0xCC000-0xCFFFF PCI bus OK
0xD0000-0xD3FFF PCI bus OK
0xD4000-0xD7FFF PCI bus OK
0xD8000-0xDBFFF PCI bus OK
0xDC000-0xDFFFF PCI bus OK
0xE0000-0xE3FFF PCI bus OK
0xE4000-0xE7FFF PCI bus OK
0xE8000-0xEBFFF PCI bus OK
0xEC000-0xEFFFF PCI bus OK
0xF0000-0xFFFFF PCI bus OK
0xFBC00000-0xFDFFFFFF PCI bus OK
0xFC000000-0xFCFFFFFF Standard VGA Graphics Adapter
OK
0xFDFFF000-0xFDFFFFFF Standard VGA Graphics Adapter
OK
0xFB800000-0xFBBFFFFF PCI bus OK
0xFBBFF000-0xFBBFFFFF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0xFB400000-0xFB7FFFFF PCI bus OK
0xFB7FF000-0xFB7FFFFF Adaptec SCSI Card 39160 -
Ultra160 SCSI OK
0xFB7FE000-0xFB7FEFFF Adaptec SCSI Card 39160 -
Ultra160 SCSI OK
0xFB000000-0xFB3FFFFF PCI bus OK
0xFB3FF000-0xFB3FFFFF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0xFAC00000-0xFAFFFFFF PCI bus OK
0xFAFFF000-0xFAFFFFFF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0xFA800000-0xFABFFFFF PCI bus OK
0xFABC0000-0xFABDFFFF Intel(R) PRO/1000 F Server
Adapter OK
0xFABF0000-0xFABFFFFF Intel(R) PRO/1000 F Server
Adapter OK
0xFA400000-0xFA7FFFFF PCI bus OK
0xFA7FF000-0xFA7FFFFF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0xFA000000-0xFA3FFFFF PCI bus OK
0xFA3FF000-0xFA3FFFFF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0xF9C00000-0xF9FFFFFF PCI bus OK
0xF9FFF000-0xF9FFFFFF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0xF9800000-0xF9BFFFFF PCI bus OK
0xF9BFF000-0xF9BFFFFF QLogic QLA23xx PCI Fibre
Channel Adapter OK
0xF9400000-0xF97FFFFF PCI bus OK
0xF97FF000-0xF97FFFFF QLogic QLA23xx PCI Fibre

Channel Adapter OK
0xF9000000-0xF93FFFFF PCI bus OK
0xF93FF000-0xF93FFFFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0xF8C00000-0xF8FFFFFF PCI bus OK
0xF8FFF000-0xF8FFFFFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0xF8800000-0xF8BFFFFF PCI bus OK
0xF8BFF000-0xF8BFFFFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0xF8400000-0xF87FFFFF PCI bus OK
0xF87FF000-0xF87FFFFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0xF8000000-0xF83FFFFF PCI bus OK
0xF83FF000-0xF83FFFFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0xF7C00000-0xF7FFFFFF PCI bus OK
0xF7FFF000-0xF7FFFFFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0xF7800000-0xF7BFFFFF PCI bus OK
0xF7BFF000-0xF7BFFFFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0xF7400000-0xF77FFFFF PCI bus OK
0xF77FF000-0xF77FFFFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0xF7000000-0xF73FFFFF PCI bus OK
0xF73FF000-0xF73FFFFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0xF6C00000-0xF6FFFFFF PCI bus OK
0xF6FFF000-0xF6FFFFFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0xF6800000-0xF6BFFFFF PCI bus OK
0xF6BFF000-0xF6BFFFFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0xF6400000-0xF67FFFFF PCI bus OK
0xF67FF000-0xF67FFFFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0xE0000000-0xF63FFFFF PCI bus OK
0xF63FF000-0xF63FFFFF QLogic QLA23xx PCI Fibre Channel Adapter OK

[Components]

[Multimedia]

[Audio Codecs]

CODEC Manufacturer Description Status File Version Size Creation Date
c:\windows\system32\msadp32.acm Microsoft Corporation OK C:\WINDOWS\system32\MSADP32.ACM 5.2.3718.0 (dnsrv.021114-1947) 49.00 KB (50,176 bytes) 11/18/2002 9:00 PM
c:\windows\system32\msgsm32.acm Microsoft Corporation OK C:\WINDOWS\system32\MSGSM32.ACM 5.2.3718.0 (dnsrv.021114-1947) 66.50 KB (68,096 bytes) 11/18/2002 9:00 PM
c:\windows\system32\msg711.acm Microsoft Corporation OK C:\WINDOWS\system32\MSG711.ACM 5.2.3718.0 (dnsrv.021114-1947) 33.00 KB (33,792 bytes) 11/18/2002 9:00 PM
c:\windows\system32\imaadp32.acm Microsoft Corporation OK C:\WINDOWS\system32\IMAADP32.ACM 5.2.3718.0 (dnsrv.021114-1947) 55.00 KB (56,320 bytes) 11/18/2002 9:00 PM
c:\windows\system32\tssoft32.acm DSP GROUP, INC. OK C:\WINDOWS\system32\TSSOFT32.ACM 1.01 29.00 KB (29,696 bytes) 11/18/2002 9:00 PM

[Video Codecs]

CODEC Manufacturer Description Status File Version Size Creation Date
c:\windows\system32\msvidc32.dll Microsoft Corporation OK C:\WINDOWS\system32\MSVIDC32.DLL 5.2.3718.0 (dnsrv.021114-1947) 67.00 KB (68,608 bytes) 11/18/2002 9:00 PM
c:\windows\system32\msrle32.dll Microsoft Corporation OK C:\WINDOWS\system32\MSRLE32.DLL 5.2.3718.0 (dnsrv.021114-1947) 24.50 KB (25,088 bytes) 11/18/2002 9:00 PM

[CD-ROM]

Item Value
Drive D:
Description CD-ROM Drive
Media Loaded Yes
Media Type CD-ROM
Name _NEC DVD-ROM DV-5800A
Manufacturer (Standard CD-ROM drives)
Status OK
Transfer Rate Not Available
SCSI Target ID 0
PNP Device ID IDE\CDROM_NEC_DVD-ROM_DV-5800A_____1.42____\5&8F26A51&0&0.0.0
Driver c:\windows\system32\drivers\cdrom.sys (5.2.3718.0 (dnsrv.021114-1947), 140.50 KB (143,872 bytes), 11/18/2002 9:00 PM)

[Sound Device]

Item Value

[Display]

Item Value
Name Standard VGA Graphics Adapter
PNP Device ID PCI\VEN_1002&DEV_4752&SUBSYS_80EB1033&REV_27\3&267A616A&0&20
Adapter Type ATI MACH64, (Standard display types) compatible
Adapter Description Standard VGA Graphics Adapter
Adapter RAM 4.00 MB (4,194,304 bytes)
Installed Drivers vga.dll,framebuf.dll,vga256,vga64k
Driver Version 5.2.3718.0
INF File display.inf (vga section)
Color Planes 1
Color Table Entries 65536
Resolution 1024 x 768 x 1 hertz
Bits/Pixel 16
Memory Address 0xFC000000-0xFCFFFFFF
I/O Port 0x00001000-0x000010FF
Memory Address 0xFDFFF000-0xFDFFFFFF
I/O Port 0x000003B0-0x000003BB
I/O Port 0x000003C0-0x000003DF
Memory Address 0xA0000-0xBFFFF
Driver c:\windows\system32\drivers\vgapnp.sys (5.2.3718.0 (dnsrv.021114-1947), 67.38 KB (68,992 bytes), 4/6/2003 4:53 AM)

[Infrared]

Item Value

[Input]

[Keyboard]

Item Value
Description Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
Name Enhanced (101- or 102-key)
Layout 00000409
PNP Device ID ACPI\PNP0303\2
Number of Function Keys 12
I/O Port 0x00000060-0x00000060
I/O Port 0x00000064-0x00000064
IRQ Channel IRQ 1
Driver c:\windows\system32\drivers\i8042prt.sys (5.2.3718.0 (dnsrv.021114-1947), 134.25 KB (137,472 bytes), 11/18/2002 9:00 PM)

[Pointing Device]

Item Value
Hardware Type PS/2 Compatible Mouse
Number of Buttons 2
Status OK
PNP Device ID ACPI\PNP0F13\2
Power Management Supported No
Double Click Threshold 6
Handedness Right Handed Operation
IRQ Channel IRQ 12
Driver c:\windows\system32\drivers\i8042prt.sys (5.2.3718.0 (dnsrv.021114-1947), 134.25 KB (137,472 bytes), 11/18/2002 9:00 PM)

[Modem]

Item Value

[Network]

[Adapter]

Item Value

Name [00000001] RAS Async Adapter
Adapter Type Not Available
Product Type RAS Async Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 4/20/2003 4:30 PM
Index 1
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000002] WAN Miniport (L2TP)
Adapter Type Not Available
Product Type WAN Miniport (L2TP)
Installed Yes
PNP Device ID ROOT\MS_L2TPMINIPORT\0000
Last Reset 4/20/2003 4:30 PM
Index 2
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000003] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPTP)
Installed Yes
PNP Device ID ROOT\MS_PPTPMINIPORT\0000
Last Reset 4/20/2003 4:30 PM
Index 3
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30

Name [00000004] WAN Miniport (PPPOE)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPPOE)
Installed Yes
PNP Device ID ROOT\MS_PPPOEMINIPORT\0000
Last Reset 4/20/2003 4:30 PM
Index 4
Service Name RasPppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30

Name [00000005] Direct Parallel
Adapter Type Not Available
Product Type Direct Parallel
Installed Yes
PNP Device ID ROOT\MS_PTIMINIPORT\0000
Last Reset 4/20/2003 4:30 PM
Index 5
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000006] WAN Miniport (IP)
Adapter Type Not Available
Product Type WAN Miniport (IP)
Installed Yes
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 4/20/2003 4:30 PM
Index 6
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000007] Intel(R) PRO/1000 F Server Adapter
Adapter Type Ethernet 802.3
Product Type Intel(R) PRO/1000 F Server Adapter
Installed Yes
PNP Device ID
PCI\VEN_8086&DEV_1001&SUBSYS_10038086&REV_02\3&20FEA912&0&10
Last Reset 4/20/2003 4:30 PM
Index 7
Service Name E1000
IP Address 10.1.2.222
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:03:47:00:40:A8
Memory Address 0xFABC0000-0xFABDFFFF
Memory Address 0xFABF0000-0xFABFFFFF
IRQ Channel IRQ 78
Driver c:\windows\system32\drivers\e1000645.sys (6.3.6.3 built by: WinDDK, 357.00 KB (365,568 bytes), 4/5/2003 9:06 PM)

[Protocol]

Item Value
Name MSAFD Tcpip [TCP/IP]
Connectionless Service No
Guarantees Delivery Yes
Guarantees Sequencing Yes
Maximum Address Size 16 bytes
Maximum Message Size 0 bytes
Message Oriented No
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data Yes
Supports Graceful Closing Yes
Supports Guaranteed Bandwidth No
Supports Multicasting No

Name MSAFD Tcpip [UDP/IP]
Connectionless Service Yes
Guarantees Delivery No
Guarantees Sequencing No
Maximum Address Size 16 bytes
Maximum Message Size 63.93 KB (65,467 bytes)
Message Oriented Yes
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting Yes

Name RSVP UDP Service Provider
Connectionless Service Yes
Guarantees Delivery No
Guarantees Sequencing No
Maximum Address Size 16 bytes
Maximum Message Size 63.93 KB (65,467 bytes)
Message Oriented Yes
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption Yes
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting Yes

Name RSVP TCP Service Provider
Connectionless Service No
Guarantees Delivery Yes

Guarantees Sequencing Yes
Maximum Address Size 16 bytes
Maximum Message Size 0 bytes
Message Oriented No
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption Yes
Supports Expedited Data Yes
Supports Graceful Closing Yes
Supports Guaranteed Bandwidth No
Supports Multicasting No

[WinSock]

Item Value
File c:\windows\system32\wsock32.dll
Size 23.00 KB (23,552 bytes)
Version 5.2.3718.0 (dnsrv.021114-1947)

[Ports]

[Serial]

Item Value
Name Communications Port (COM2)
Status OK
PNP Device ID ACPI\PNP0501\1
Maximum Input Buffer Size 0
Maximum Output Buffer Size No
Settable Baud Rate Yes
Settable Data Bits Yes
Settable Flow Control Yes
Settable Parity Yes
Settable Parity Check Yes
Settable Stop Bits Yes
Settable RLSD Yes
Supports RLSD Yes
Supports 16 Bit Mode No
Supports Special Characters No
Baud Rate 9600
Bits/Byte 8
Stop Bits 1
Parity None
Busy No
Abort Read/Write on Error No
Binary Mode Enabled Yes
Continue XMit on XOff No
CTS Outflow Control No
Discard NULL Bytes No
DSR Outflow Control 0
DSR Sensitivity 0
DTR Flow Control Type Enable
EOF Character 0
Error Replace Character 0
Error Replacement Enabled No
Event Character 0
Parity Check Enabled No
RTS Flow Control Type Enable
XOff Character 19
XOffXMit Threshold 512
XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control 0
I/O Port 0x000002F8-0x000002FF
IRQ Channel IRQ 245
Driver c:\windows\system32\drivers\serial.sys (5.2.3718.0 (dnsrv.021114-1947), 172.25 KB (176,384 bytes), 11/18/2002 9:00 PM)

Name Communications Port (COM1)
Status OK
PNP Device ID ACPI\PNP0501\2
Maximum Input Buffer Size 0
Maximum Output Buffer Size No
Settable Baud Rate Yes
Settable Data Bits Yes
Settable Flow Control Yes
Settable Parity Yes
Settable Parity Check Yes
Settable Stop Bits Yes
Settable RLSD Yes
Supports RLSD Yes
Supports 16 Bit Mode No
Supports Special Characters No
Baud Rate 9600
Bits/Byte 8
Stop Bits 1
Parity None
Busy No
Abort Read/Write on Error No
Binary Mode Enabled Yes
Continue XMit on XOff No
CTS Outflow Control No
Discard NULL Bytes No
DSR Outflow Control 0
DSR Sensitivity 0
DTR Flow Control Type Enable
EOF Character 0
Error Replace Character 0
Error Replacement Enabled No
Event Character 0
Parity Check Enabled No
RTS Flow Control Type Enable
XOff Character 19
XOffXMit Threshold 512
XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control 0
I/O Port 0x000003F8-0x000003FF
IRQ Channel IRQ 4
Driver c:\windows\system32\drivers\serial.sys (5.2.3718.0 (dnsrv.021114-1947), 172.25 KB (176,384 bytes), 11/18/2002 9:00 PM)

[Parallel]

Item Value

[Storage]

[Drives]

Item Value
Drive C:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 33.41 GB (35,878,670,336 bytes)
Free Space 15.33 GB (16,459,542,528 bytes)
Volume Name
Volume Serial Number D0436208

Drive D:
Description CD-ROM Disc

Drive F:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 418.91 GB (449,799,434,240 bytes)
Free Space 418.83 GB (449,718,087,680 bytes)
Volume Name b026
Volume Serial Number 5C1822E4

Drive G:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 418.91 GB (449,799,434,240 bytes)
Free Space 312.95 GB (336,022,433,792 bytes)
Volume Name b029
Volume Serial Number 107186E8

Drive H:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available

Drive I:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available

Drive J:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available

Volume Name Not Available
Volume Serial Number Not Available

Drive K:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 408.92 GB (439,073,669,120 bytes)
Free Space 408.84 GB (438,992,654,336 bytes)
Volume Name b030
Volume Serial Number C89155C4

Drive L:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available

Drive Z:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 32.82 GB (35,237,097,472 bytes)
Free Space 32.69 GB (35,100,479,488 bytes)
Volume Name junction
Volume Serial Number B8E08F74

[Disks]

Item Value
Description Disk drive
Manufacturer (Standard disk drives)
Model HITACHI DK32DJ-36MC SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 23
SCSI Target ID 15
Sectors/Track 63
Size 33.87 GB (36,372,188,160 bytes)
Total Cylinders 4,422
Total Sectors 71,039,430
Total Tracks 1,127,610
Tracks/Cylinder 255
Partition Disk #76, Partition #0
Partition Size 345.12 MB (361,880,064 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #76, Partition #1
Partition Size 33.41 GB (35,878,671,360 bytes)
Partition Starting Offset 493,516,800 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 19
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #58, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #58, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #58, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 19
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #59, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #59, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #59, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 19
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #60, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #60, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #60, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 19
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #61, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #61, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #61, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 19
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #62, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #62, Partition #1

Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #62, Partition #2
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 59,057,510,400 bytes
Partition Disk #62, Partition #3
Partition Size 408.92 GB (439,073,671,680 bytes)
Partition Starting Offset 96,638,814,720 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 19
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #63, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #63, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #63, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 12
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #40, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #40, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #40, Partition #2
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 59,057,510,400 bytes
Partition Disk #40, Partition #3
Partition Size 408.92 GB (439,073,671,680 bytes)
Partition Starting Offset 96,638,814,720 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 12
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #41, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #41, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #41, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 12
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #42, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #42, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #42, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 12
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #43, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #43, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #43, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 12
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #44, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #44, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #44, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 12
SCSI Target ID 1
Sectors/Track 63

Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #45, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #45, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #45, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 7
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #10, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #10, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #10, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 7
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #11, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #11, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #11, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 7
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #12, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #12, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #12, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 7
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #13, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #13, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #13, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 7
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #14, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #14, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #14, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 7
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #15, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #15, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #15, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0

SCSI Port 4
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #4, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #4, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #4, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 4
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #5, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #5, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #5, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 4
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #6, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #6, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #6, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 4
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #7, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #7, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #7, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 4
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #8, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #8, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #8, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 4
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #9, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #9, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #9, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 15
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #46, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #46, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #46, Partition #2
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 59,057,510,400 bytes
Partition Disk #46, Partition #3
Partition Size 408.92 GB (439,073,671,680 bytes)
Partition Starting Offset 96,638,814,720 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device

Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 15
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #47, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #47, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #47, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 15
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #48, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #48, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #48, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 15
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #49, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #49, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #49, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 15
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #50, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #50, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #50, Partition #2
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 59,057,510,400 bytes
Partition Disk #50, Partition #3
Partition Size 408.92 GB (439,073,671,680 bytes)
Partition Starting Offset 96,638,814,720 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 15
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #51, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #51, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #51, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 10
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #28, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #28, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #28, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 10
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #29, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #29, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #29, Partition #2

Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 59,057,510,400 bytes
Partition Disk #29, Partition #3
Partition Size 408.92 GB (439,073,671,680 bytes)
Partition Starting Offset 96,638,814,720 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 10
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #30, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #30, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #30, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 10
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #31, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #31, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #31, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 10
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #32, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #32, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #32, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 10
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #33, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #33, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #33, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 3
SCSI Target ID 0
Sectors/Track 63
Size 232.83 GB (249,999,160,320 bytes)
Total Cylinders 30,394
Total Sectors 488,279,610
Total Tracks 7,750,470
Tracks/Cylinder 255
Partition Disk #0, Partition #0
Partition Size 200.01 GB (214,753,803,264 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #0, Partition #1
Partition Size 32.82 GB (35,237,099,520 bytes)
Partition Starting Offset 214,753,835,520 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 3
SCSI Target ID 0
Sectors/Track 63
Size 232.83 GB (249,999,160,320 bytes)
Total Cylinders 30,394
Total Sectors 488,279,610
Total Tracks 7,750,470
Tracks/Cylinder 255
Partition Disk #1, Partition #0
Partition Size 200.01 GB (214,753,803,264 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #1, Partition #1
Partition Size 32.82 GB (35,237,099,520 bytes)
Partition Starting Offset 214,753,835,520 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 3
SCSI Target ID 0
Sectors/Track 63
Size 232.83 GB (249,999,160,320 bytes)
Total Cylinders 30,394
Total Sectors 488,279,610
Total Tracks 7,750,470
Tracks/Cylinder 255
Partition Disk #2, Partition #0
Partition Size 200.01 GB (214,753,803,264 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #2, Partition #1
Partition Size 32.82 GB (35,237,099,520 bytes)
Partition Starting Offset 214,753,835,520 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus 0
SCSI Logical Unit 3
SCSI Port 3
SCSI Target ID 0
Sectors/Track 63
Size 232.83 GB (249,999,160,320 bytes)
Total Cylinders 30,394
Total Sectors 488,279,610
Total Tracks 7,750,470
Tracks/Cylinder 255
Partition Disk #3, Partition #0
Partition Size 200.01 GB (214,753,803,264 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #3, Partition #1
Partition Size 32.82 GB (35,237,099,520 bytes)
Partition Starting Offset 214,753,835,520 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 21
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #70, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #70, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #70, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 21
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #71, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #71, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #71, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 21
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #72, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #72, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #72, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 21
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #73, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #73, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #73, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 21
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #74, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #74, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #74, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 21
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #75, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #75, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #75, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 16
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #52, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #52, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #52, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 16
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #53, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #53, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #53, Partition #2
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 59,057,510,400 bytes
Partition Disk #53, Partition #3
Partition Size 408.92 GB (439,073,671,680 bytes)
Partition Starting Offset 96,638,814,720 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 16
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #54, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #54, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #54, Partition #2
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 59,057,510,400 bytes
Partition Disk #54, Partition #3
Partition Size 408.92 GB (439,073,671,680 bytes)
Partition Starting Offset 96,638,814,720 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 16
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #55, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #55, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #55, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 16
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #56, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #56, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #56, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 16
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #57, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #57, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #57, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 9
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #22, Partition #0

Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #22, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #22, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 9
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #23, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #23, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #23, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 9
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #24, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #24, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #24, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 9
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #25, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #25, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #25, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 9
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #26, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #26, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #26, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 9
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #27, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #27, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #27, Partition #2
Partition Size 418.91 GB (449,799,436,800 bytes)
Partition Starting Offset 85,913,049,600 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 20
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #64, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #64, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #64, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 20
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450

Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #65, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #65, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #65, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 20
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #66, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #66, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #66, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 20
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #67, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #67, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #67, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 20
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #68, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #68, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #68, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 20
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #69, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #69, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #69, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 11
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #34, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #34, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #34, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 11
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #35, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #35, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #35, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 11
SCSI Target ID 0
Sectors/Track 63

Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #36, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #36, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #36, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 11
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #37, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #37, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #37, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 11
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #38, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #38, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #38, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 11
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #39, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #39, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #39, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 8
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #16, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #16, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #16, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 8
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #17, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #17, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #17, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 8
SCSI Target ID 0
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #18, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #18, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #18, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0

SCSI Port 8
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #19, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #19, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #19, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 8
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #20, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #20, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #20, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 8
SCSI Target ID 1
Sectors/Track 63
Size 498.92 GB (535,712,486,400 bytes)
Total Cylinders 65,130
Total Sectors 1,046,313,450
Total Tracks 16,608,150
Tracks/Cylinder 255
Partition Disk #21, Partition #0
Partition Size 20.00 GB (21,476,173,824 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #21, Partition #1
Partition Size 35.00 GB (37,581,304,320 bytes)
Partition Starting Offset 21,476,206,080 bytes
Partition Disk #21, Partition #2
Partition Size 443.92 GB (476,654,976,000 bytes)
Partition Starting Offset 59,057,510,400 bytes

[SCSI]

Item Value
Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&1070020&0&10
I/O Port 0x00001200-0x000012FF
Memory Address 0xFBBFF000-0xFBBFFFFF
IRQ Channel IRQ 64
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name Adaptec SCSI Card 39160 - Ultra160 SCSI
Manufacturer Adaptec
Status OK
PNP Device ID
PCI\VEN_9005&DEV_00C0&SUBSYS_F6209005&REV_01\3&29E81982&0&10
I/O Port 0x00001300-0x000014FF
Memory Address 0xFB7FF000-0xFB7FFFFF
IRQ Channel IRQ 68
Driver c:\windows\system32\drivers\adpu160m.sys (RTC_XP07 (lab01_n(storbuild).010917-1859), 319.88 KB (327,552 bytes), 11/18/2002 9:00 PM)

Name Adaptec SCSI Card 39160 - Ultra160 SCSI
Manufacturer Adaptec
Status OK
PNP Device ID
PCI\VEN_9005&DEV_00C0&SUBSYS_F6209005&REV_01\3&29E81982&0&11
I/O Port 0x00001400-0x000014FF
Memory Address 0xFB7FE000-0xFB7FEFFF
IRQ Channel IRQ 67
Driver c:\windows\system32\drivers\adpu160m.sys (RTC_XP07 (lab01_n(storbuild).010917-1859), 319.88 KB (327,552 bytes), 11/18/2002 9:00 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&172E68DD&0&10
I/O Port 0x00001500-0x000015FF
Memory Address 0xFB3FF000-0xFB3FFFFF
IRQ Channel IRQ 69
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&474B838&0&10
I/O Port 0x00001600-0x000016FF
Memory Address 0xFAFFF000-0xFAFFFFFF
IRQ Channel IRQ 73
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&AD74055&0&10
I/O Port 0x00001700-0x000017FF
Memory Address 0xFA7FF000-0xFA7FFFFF
IRQ Channel IRQ 84
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&23C0707C&0&10
I/O Port 0x00001800-0x000018FF
Memory Address 0xFA3FF000-0xFA3FFFFF
IRQ Channel IRQ 86
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&1106BFD7&0&10
I/O Port 0x00001900-0x000019FF
Memory Address 0xF9FFF000-0xF9FFFFFF
IRQ Channel IRQ 90
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&1B2F0CE&0&10
I/O Port 0x00001A00-0x00001AFF
Memory Address 0xF9BFF000-0xF9BFFFFF
IRQ Channel IRQ 91
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter

Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&146CA173&0&10
I/O Port 0x00001B00-0x00001BFF
Memory Address 0xF97FF000-0xF97FFFFF
IRQ Channel IRQ 95
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&39E002BD&0&10
I/O Port 0x00001C00-0x00001CFF
Memory Address 0xF93FF000-0xF93FFFFF
IRQ Channel IRQ 100
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&1D521019&0&10
I/O Port 0x00001D00-0x00001DFF
Memory Address 0xF8FFF000-0xF8FFFFFF
IRQ Channel IRQ 150
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&300BC0BE&0&10
I/O Port 0x00001E00-0x00001EFF
Memory Address 0xF8BFF000-0xF8BFFFFF
IRQ Channel IRQ 152
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&72AA75C&0&10
I/O Port 0x00001F00-0x00001FFF
Memory Address 0xF87FF000-0xF87FFFFF
IRQ Channel IRQ 156
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&19E45801&0&10
I/O Port 0x00002000-0x000020FF
Memory Address 0xF83FF000-0xF83FFFFF
IRQ Channel IRQ 157
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&2C9E08A6&0&10
I/O Port 0x00002100-0x000021FF
Memory Address 0xF7FFF000-0xF7FFFFFF
IRQ Channel IRQ 161
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&36B90202&0&10
I/O Port 0x00002200-0x000022FF
Memory Address 0xF7BFF000-0xF7BFFFFF
IRQ Channel IRQ 166
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&1145A0B8&0&10
I/O Port 0x00002300-0x000023FF
Memory Address 0xF77FF000-0xF77FFFFF
IRQ Channel IRQ 172
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&D525062&0&10
I/O Port 0x00002400-0x000024FF
Memory Address 0xF73FF000-0xF73FFFFF
IRQ Channel IRQ 174
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&200C0107&0&10
I/O Port 0x00002500-0x000025FF
Memory Address 0xF6FFF000-0xF6FFFFFF
IRQ Channel IRQ 178
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&32C5B1AC&0&10
I/O Port 0x00002600-0x000026FF
Memory Address 0xF6BFF000-0xF6BFFFFF
IRQ Channel IRQ 179
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&7B03F9A&0&10
I/O Port 0x00002700-0x000027FF
Memory Address 0xF67FF000-0xF67FFFFF
IRQ Channel IRQ 183
Driver c:\windows\system32\drivers\ql2300.sys (8.2.0 Beta 10 (W64 VI), 676.38 KB (692,608 bytes), 4/5/2003 8:35 PM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&1DD7A857&0&10
I/O Port 0x00002800-0x0000FFFF
Memory Address 0xF63FF000-0xF63FFFFF
IRQ Channel IRQ 188

[IDE]

Item Value
Name Intel(r) IA64 Bus Master IDE Controller
Manufacturer Intel
Status OK
PNP Device ID
PCI\VEN_8086&DEV_7601&SUBSYS_01061033&REV_01\3&267A616A&0&11
I/O Port 0x00001120-0x0000112F
Driver c:\windows\system32\drivers\intelide.sys (5.2.3718.0 (dnsrv.021114-1947), 8.75 KB (8,960 bytes), 11/18/2002 9:00 PM)

Name Primary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCIIDE\IDECHANNEL\4&31FC9C45&0&0
I/O Port 0x000001F0-0x000001F7
I/O Port 0x000003F6-0x000003F6
IRQ Channel IRQ 14
Driver c:\windows\system32\drivers\atapi.sys (5.2.3718.0

(dnsrv.021114-1947), 282.13 KB (288,896 bytes), 11/18/2002 9:00 PM)

[Printing]

Name Driver Port Name Server Name

[Problem Devices]

Device PNP Device ID Error Code
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&267A616A&0&08 The drivers for this device are not installed.
Not Available ACPI\NEC4171\0 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FD&SUBSYS_00FD1033&REV_00\3&267A616A&0&70 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&1070020&0&08 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&172E68DD&0&08 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&E44F86D&0&08 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&33B859B7&0&08 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&23C0707C&0&08 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&1B2F0CE&0&08 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&27265218&0&08 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&A985F74&0&08 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&300BC0BE&0&08 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&19E45801&0&08 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&3BCEF44&0&08 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&23FF515D&0&08 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&D525062&0&08 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&32C5B1AC&0&08 The drivers for this device are not installed.
System Interrupt Controller
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&309158FC&0&08 The drivers for this device are not installed.

[USB]

Device PNP Device ID
Intel(r) 82372FB PCI to USB Universal Host Controller
PCI\VEN_8086&DEV_7602&SUBSYS_01061033&REV_01\3&267A616A&0&12
USB Root Hub USB\ROOT_HUB\4&5C3BD33&0

[Software Environment]

[System Drivers]

Name Description File Type Started Start Mode
State Status Error Control Accept
Pause Accept Stop
abiosdsk Abiosdsk Not Available Kernel Driver
No Disabled Stopped OK Ignore
No No
acpi Microsoft ACPI Driver
c:\windows\system32\drivers\acpi.sys Kernel
Driver Yes Boot Running OK Normal
No Yes
acpiec ACPIEC c:\windows\system32\drivers\acpiec.sys
Kernel Driver No Disabled Stopped
OK Normal No No
adpu160m adpu160m c:\windows\system32\drivers\adpu160m.sys
Kernel Driver Yes Boot Running
OK Normal No Yes
adpu320 adpu320 Not Available Kernel Driver
No Disabled Stopped OK Normal
No No
afcnt afcnt Not Available Kernel Driver
No Disabled Stopped OK Normal
No No
afd AFD Networking Support Environment
c:\windows\system32\drivers\afd.sys Kernel
Driver Yes Auto Running OK Normal
No Yes
aic78u2 aic78u2 Not Available Kernel Driver
No Disabled Stopped OK Normal
No No
aic78xx aic78xx Not Available Kernel Driver
No Disabled Stopped OK Normal
No No
aliide AliIde Not Available Kernel Driver
No Disabled Stopped OK Normal
No No
asyncmac RAS Asynchronous Media Driver
c:\windows\system32\drivers\asyncmac.sys Kernel
Driver No Manual Stopped OK Normal
No No
atapi Standard IDE/ESDI Hard Disk Controller
c:\windows\system32\drivers\atapi.sys Kernel
Driver Yes Boot Running OK Normal
No Yes
atdisk Atdisk Not Available Kernel Driver
No Disabled Stopped OK Ignore
No No
atmarpc ATM ARP Client Protocol
c:\windows\system32\drivers\atmarpc.sys Kernel
Driver No Manual Stopped OK Normal
No No
audstub Audio Stub Driver
c:\windows\system32\drivers\audstub.sys Kernel
Driver Yes Manual Running OK Normal
No Yes
beep Beep c:\windows\system32\drivers\beep.sys
Kernel Driver Yes System Running
OK Normal No Yes
cbidf2k cbidf2k c:\windows\system32\drivers\cbidf2k.sys
Kernel Driver No Disabled Stopped
OK Normal No No
cdfs Cdfs c:\windows\system32\drivers\cdfs.sys
File System Driver Yes Disabled Running
OK Normal No Yes
cdrom CD-ROM Driver
c:\windows\system32\drivers\cdrom.sys Kernel
Driver Yes System Running OK Normal
No Yes
changer Changer Not Available Kernel Driver
No System Stopped OK Ignore
No No
clusdisk Cluster Disk Driver
c:\windows\system32\drivers\clusdisk.sys Kernel
Driver No Disabled Stopped OK Normal
No No
cmdide CmdIde Not Available Kernel Driver
No Disabled Stopped OK Normal
No No
cpqarry2 cpqarry2 Not Available Kernel Driver
No Disabled Stopped OK Normal
No No
cpqcissm cpqcissm Not Available Kernel Driver
No Disabled Stopped OK Normal
No No
cpqfcalm cpqfcalm Not Available Kernel Driver
No Disabled Stopped OK Normal
No No
crcdisk CRC Disk Filter Driver
c:\windows\system32\drivers\crcdisk.sys Kernel
Driver Yes Boot Running OK Normal
No Yes
dfsdriver DfsDriver c:\windows\system32\drivers\dfs.sys
File System Driver Yes Boot Running
OK Normal No Yes
disk Disk Driver c:\windows\system32\drivers\disk.sys
Kernel Driver Yes Boot Running
OK Normal No Yes
dmboot dmboot c:\windows\system32\drivers\dmboot.sys
Kernel Driver No Disabled Stopped
OK Normal No No

```
dmio    Logical Disk Manager Driver
        c:\windows\system32\drivers\dmio.sys        Kernel
Driver  Yes     Boot    Running OK      Normal
        No      Yes
dmload  dmload  c:\windows\system32\drivers\dmload.sys
        Kernel Driver           Yes     Boot    Running
        OK      Normal  No      Yes
e1000   Intel(R) PRO/1000 Device Driver
        c:\windows\system32\drivers\e1000645.sys    Kernel
Driver  Yes     Manual  Running OK      Normal
        No      Yes
em      em      c:\windows\system32\drivers\em.sys
        Kernel Driver           No      Manual  Stopped
        OK      Normal  No      No
fastfat Fastfat c:\windows\system32\drivers\fastfat.sys
        File System Driver      Yes     Disabled        Running
        OK      Normal  No      Yes
fdc     Fdc     c:\windows\system32\drivers\fdc.sys
        Kernel Driver           No      System  Stopped
        OK      Ignore  No      No
fips    Fips    c:\windows\system32\drivers\fips.sys
        Kernel Driver           Yes     System  Running
        OK      Normal  No      Yes
flpydisk        Flpydisk        c:\windows\system32\drivers\flpydisk.sys
        Kernel Driver           No      System  Stopped
        OK      Ignore  No      No
ftdisk  Volume Manager Driver
        c:\windows\system32\drivers\ftdisk.sys      Kernel
Driver  Yes     Boot    Running OK      Normal
        No      Yes
gpc     Generic Packet Classifier
        c:\windows\system32\drivers\msgpc.sys       Kernel
Driver  Yes     Manual  Running OK      Normal
        No      Yes
hpn     hpn     Not Available           Kernel Driver
        No      Disabled        Stopped OK      Normal
        No      No
http    HTTP    c:\windows\system32\drivers\http.sys
        Kernel Driver           No      Manual  Stopped
        OK      Normal  No      No
i2omgmt i2omgmt Not Available           Kernel Driver
        No      System  Stopped OK      Normal
        No      No
i8042prt        i8042 Keyboard and PS/2 Mouse Port Driver
        c:\windows\system32\drivers\i8042prt.sys    Kernel
Driver  Yes     System  Running OK      Normal
        No      Yes
imapi   CD-Burning Filter Driver
        c:\windows\system32\drivers\imapi.sys       Kernel
Driver  No      System  Stopped OK      Normal
        No      No
intelide        IntelIde        c:\windows\system32\drivers\intelide.sys
        Kernel Driver           Yes     Boot    Running
        OK      Normal  No      Yes
ipfilterdriver          IP Traffic Filter Driver
        c:\windows\system32\drivers\ipfltdrv.sys    Kernel
Driver  No      Manual  Stopped OK      Normal
        No      No
ipinip  IP in IP Tunnel Driver
        c:\windows\system32\drivers\ipinip.sys      Kernel
Driver  No      Manual  Stopped OK      Normal
        No      No
ipnat   IP Network Address Translator
        c:\windows\system32\drivers\ipnat.sys       Kernel
Driver  No      Manual  Stopped OK      Normal
        No      No
ipsec   IPSEC driver
        c:\windows\system32\drivers\ipsec.sys       Kernel
Driver  Yes     System  Running OK      Normal
        No      Yes
isapnp  PnP ISA/EISA Bus Driver
        c:\windows\system32\drivers\isapnp.sys      Kernel
Driver  Yes     Boot    Running OK      Critical
        No      Yes
kbdclass        Keyboard Class Driver
        c:\windows\system32\drivers\kbdclass.sys    Kernel
Driver  Yes     System  Running OK      Normal
        No      Yes
ksecdd  KSecDD  c:\windows\system32\drivers\ksecdd.sys
        Kernel Driver           Yes     Boot    Running
        OK      Normal  No      Yes
lp6nds35        lp6nds35        Not Available           Kernel Driver
        No      Disabled        Stopped OK      Normal
        No      No
mnmdd   mnmdd   Not Available           Kernel Driver
        No      System  Stopped OK      Ignore
        No      No
modem   Modem   c:\windows\system32\drivers\modem.sys
        Kernel Driver           No      Manual  Stopped
        OK      Ignore  No      No
mouclass        Mouse Class Driver
        c:\windows\system32\drivers\mouclass.sys    Kernel
Driver  Yes     System  Running OK      Normal
        No      Yes
mountmgr        Mount Point Manager
        c:\windows\system32\drivers\mountmgr.sys    Kernel
Driver  Yes     Boot    Running OK      Normal
        No      Yes
mraid35x        mraid35x        Not Available           Kernel Driver
        No      Disabled        Stopped OK      Normal
        No      No
mrxdav  WebDav Client Redirector
        c:\windows\system32\drivers\mrxdav.sys      File
System Driver   No      Manual  Stopped OK
        Normal  No      No
mrxsmb  MRXSMB  c:\windows\system32\drivers\mrxsmb.sys
        File System Driver      Yes     System  Running
        OK      Normal  No      Yes
msfs    Msfs    c:\windows\system32\drivers\msfs.sys
        File System Driver      Yes     System  Running
        OK      Normal  No      Yes
mup     Mup     c:\windows\system32\drivers\mup.sys
        File System Driver      Yes     Boot    Running
        OK      Normal  No      Yes
ndis    NDIS System Driver
        c:\windows\system32\drivers\ndis.sys        Kernel
Driver  Yes     Boot    Running OK      Normal
        No      Yes
ndistapi        Remote Access NDIS TAPI Driver
        c:\windows\system32\drivers\ndistapi.sys    Kernel
Driver  Yes     Manual  Running OK      Normal
        No      Yes
ndisuio NDIS Usermode I/O Protocol
        c:\windows\system32\drivers\ndisuio.sys     Kernel
Driver  Yes     Manual  Running OK      Normal
        No      Yes
ndiswan Remote Access NDIS WAN Driver
        c:\windows\system32\drivers\ndiswan.sys     Kernel
Driver  Yes     Manual  Running OK      Normal
        No      Yes
ndproxy NDIS Proxy
        c:\windows\system32\drivers\ndproxy.sys     Kernel
Driver  Yes     Manual  Running OK      Normal
        No      Yes
netbios NetBIOS Interface
        c:\windows\system32\drivers\netbios.sys     File
System Driver   Yes     System  Running OK
        Normal  No      Yes
netbt   NetBios over Tcpip
        c:\windows\system32\drivers\netbt.sys       Kernel
Driver  Yes     System  Running OK      Normal
        No      Yes
nfrd960 nfrd960 Not Available           Kernel Driver
        No      Disabled        Stopped OK      Normal
        No      No
npfs    Npfs    c:\windows\system32\drivers\npfs.sys
        File System Driver      Yes     System  Running
        OK      Normal  No      Yes
ntfs    Ntfs    c:\windows\system32\drivers\ntfs.sys
        File System Driver      Yes     Disabled        Running
        OK      Normal  No      Yes
null    Null    c:\windows\system32\drivers\null.sys
        Kernel Driver           Yes     System  Running
        OK      Normal  No      Yes
partmgr Partition Manager
        c:\windows\system32\drivers\partmgr.sys     Kernel
Driver  Yes     Boot    Running OK      Normal
        No      Yes
pci     PCI Bus Driver
        c:\windows\system32\drivers\pci.sys         Kernel
Driver  Yes     Boot    Running OK      Critical
        No      Yes
pciide  PCIIde  Not Available           Kernel Driver
        No      Disabled        Stopped OK      Normal
        No      No
pcmcia  Pcmcia  c:\windows\system32\drivers\pcmcia.sys
        Kernel Driver           No      Disabled        Stopped
        OK      Normal  No      No
pdcomp  PDCOMP  Not Available           Kernel Driver
        No      Manual  Stopped OK      Ignore
        No      No
pdframe PDFRAME Not Available           Kernel Driver
        No      Manual  Stopped OK      Ignore
        No      No
pdreli  PDRELI  Not Available           Kernel Driver
        No      Manual  Stopped OK      Ignore
        No      No
pdrframe        PDRFRAME        Not Available           Kernel
Driver  No      Manual  Stopped OK      Ignore
        No      No
pptpminiport    WAN Miniport (PPTP)
        c:\windows\system32\drivers\raspptp.sys     Kernel
Driver  Yes     Manual  Running OK      Normal
        No      Yes
processor       Processor Driver
        c:\windows\system32\drivers\processr.sys    Kernel
```

```
Driver          Yes         Manual      Running     OK          Normal
                No          Yes
ptilink         Direct Parallel Link Driver
                c:\windows\system32\drivers\ptilink.sys         Kernel
Driver          Yes         Manual      Running     OK          Normal
                No          Yes
ql1080          ql1080      Not Available           Kernel Driver
                No          Disabled    Stopped     OK          Normal
                No          No
ql10wnt         Ql10wnt     Not Available           Kernel Driver
                No          Disabled    Stopped     OK          Normal
                No          No
ql12160         ql12160     Not Available           Kernel Driver
                No          Disabled    Stopped     OK          Normal
                No          No
ql1240          ql1240      Not Available           Kernel Driver
                No          Disabled    Stopped     OK          Normal
                No          No
ql1280          ql1280      Not Available           Kernel Driver
                No          Disabled    Stopped     OK          Normal
                No          No
ql2100          ql2100      Not Available           Kernel Driver
                No          Disabled    Stopped     OK          Normal
                No          No
ql2200          ql2200      Not Available           Kernel Driver
                No          Disabled    Stopped     OK          Normal
                No          No
ql2300          ql2300      c:\windows\system32\drivers\ql2300.sys
                Kernel Driver           Yes         Boot        Running
                OK          Normal      No          Yes
qldirect        qldirect    c:\windows\system32\drivers\qldirect.sys
                Kernel Driver           Yes         Boot        Running
                OK          Normal      No          Yes
qlvika          qlvika      c:\windows\system32\drivers\qlvika.sys
                Kernel Driver           Yes         Auto        Running
                OK          Normal      No          Yes
rasacd          Remote Access Auto Connection Driver
                c:\windows\system32\drivers\rasacd.sys          Kernel
Driver          Yes         System      Running     OK          Normal
                No          Yes
rasl2tp         WAN Miniport (L2TP)
                c:\windows\system32\drivers\rasl2tp.sys         Kernel
Driver          Yes         Manual      Running     OK          Normal
                No          Yes
raspppoe        Remote Access PPPOE Driver
                c:\windows\system32\drivers\raspppoe.sys        Kernel
Driver          Yes         Manual      Running     OK          Normal
                No          Yes
raspti          Direct Parallel
                c:\windows\system32\drivers\raspti.sys          Kernel
Driver          Yes         Manual      Running     OK          Normal
                No          Yes
rdbss           Rdbss       c:\windows\system32\drivers\rdbss.sys
                File System Driver      Yes         System      Running
                OK          Normal      No          Yes
rdpcdd          RDPCDD      c:\windows\system32\drivers\rdpcdd.sys
                Kernel Driver           Yes         System      Running
                OK          Ignore      No          Yes
rdpdr           Terminal Server Device Redirector Driver
                c:\windows\system32\drivers\rdpdr.sys           Kernel
Driver          Yes         Manual      Running     OK          Normal
                No          Yes
rdpwd           RDPWD       c:\windows\system32\drivers\rdpwd.sys
                Kernel Driver           No          Manual      Stopped
                OK          Ignore      No          No
redbook         Digital CD Audio Playback Filter Driver
                c:\windows\system32\drivers\redbook.sys         Kernel
Driver          Yes         System      Running     OK          Normal
                No          Yes
serenum         Serenum Filter Driver
                c:\windows\system32\drivers\serenum.sys         Kernel
Driver          Yes         Manual      Running     OK          Normal
                No          Yes
serial          Serial port driver
                c:\windows\system32\drivers\serial.sys          Kernel
Driver          Yes         System      Running     OK          Ignore
                No          Yes
sfloppy         Sfloppy     c:\windows\system32\drivers\sfloppy.sys
                Kernel Driver           No          System      Stopped
                OK          Ignore      No          No
simbad          Simbad      Not Available           Kernel Driver
                No          Disabled    Stopped     OK          Normal
                No          No
srv             Srv         c:\windows\system32\drivers\srv.sys
                File System Driver      Yes         Manual      Running
                OK          Normal      No          Yes
swenum          Software Bus Driver
                c:\windows\system32\drivers\swenum.sys          Kernel
Driver          Yes         Manual      Running     OK          Normal
                No          Yes
symc8xx         symc8xx     Not Available           Kernel Driver
                No          Disabled    Stopped     OK          Normal
                No          No
symmpi          symmpi      Not Available           Kernel Driver
                No          Disabled    Stopped     OK          Normal
                No          No
sym_hi          sym_hi      Not Available           Kernel Driver
                No          Disabled    Stopped     OK          Normal
                No          No
sym_u3          sym_u3      Not Available           Kernel Driver
                No          Disabled    Stopped     OK          Normal
                No          No
tbs             tbs         \??\c:\windows\system32\drivers\tbs.sys
                Kernel Driver           No          Manual      Stopped
                OK          Normal      No          No
tcpip           TCP/IP Protocol Driver
                c:\windows\system32\drivers\tcpip.sys           Kernel
Driver          Yes         System      Running     OK          Normal
                No          Yes
tdpipe          TDPIPE      c:\windows\system32\drivers\tdpipe.sys
                Kernel Driver           No          Manual      Stopped
                OK          Ignore      No          No
tdtcp           TDTCP       c:\windows\system32\drivers\tdtcp.sys
                Kernel Driver           No          Manual      Stopped
                OK          Ignore      No          No
termdd          Terminal Device Driver
                c:\windows\system32\drivers\termdd.sys          Kernel
Driver          Yes         System      Running     OK          Normal
                No          Yes
toside          TosIde      Not Available           Kernel Driver
                No          Disabled    Stopped     OK          Normal
                No          No
udfs            Udfs        c:\windows\system32\drivers\udfs.sys
                File System Driver      No          Disabled    Stopped
                OK          Normal      No          No
usbhub          USB2 Enabled Hub
                c:\windows\system32\drivers\usbhub.sys          Kernel
Driver          Yes         Manual      Running     OK          Normal
                No          Yes
usbuhci         Microsoft USB Universal Host Controller Miniport Driver
                c:\windows\system32\drivers\usbuhci.sys         Kernel
Driver          Yes         Manual      Running     OK          Normal
                No          Yes
vga             vga         c:\windows\system32\drivers\vgapnp.sys
                Kernel Driver           Yes         Manual      Running
                OK          Ignore      No          Yes
vgasave         VGA Display Controller.
                c:\windows\system32\drivers\vga.sys             Kernel
Driver          No          System      Stopped     OK          Ignore
                No          No
viaide          ViaIde      Not Available           Kernel Driver
                No          Disabled    Stopped     OK          Normal
                No          No
volsnap         Storage volumes
                c:\windows\system32\drivers\volsnap.sys         Kernel
Driver          Yes         Boot        Running     OK          Normal
                No          Yes
wanarp          Remote Access IP ARP Driver
                c:\windows\system32\drivers\wanarp.sys          Kernel
Driver          Yes         Manual      Running     OK          Normal
                No          Yes
wdica           WDICA       Not Available           Kernel Driver
                No          Manual      Stopped     OK          Ignore
                No          No
wlbs            Network Load Balancing
                c:\windows\system32\drivers\wlbs.sys            Kernel
Driver          No          Manual      Stopped     OK          Normal
                No          No

[Signed Drivers]

Device Name             Signed      Device Class            Driver
Version     Driver Date Manufacturer            INF Name    Driver
Name        Device ID
Not Available           Not Available           Not Available
            Not Available           Not Available           Not
Available   Not Available           Not Available
            HTREE\ROOT\0
ACPI IA64-based PC      Yes         COMPUTER                5.2.3718.0
            10/1/2002   (Standard computers)    hal.inf     Not
Available   ROOT\ACPI_HAL\0000
Microsoft ACPI-Compliant System     Yes         SYSTEM      5.2.3718.0
            10/1/2002   Microsoft   acpi.inf    Not Available
            ACPI_HAL\PNP0C08\0
Processor   Yes         PROCESSOR               5.2.3718.0  10/1/2002
            (Standard processor types)          cpu.inf     Not
Available   ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\_0
Processor   Yes         PROCESSOR               5.2.3718.0  10/1/2002
            (Standard processor types)          cpu.inf     Not
Available   ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\_1
Processor   Yes         PROCESSOR               5.2.3718.0  10/1/2002
            (Standard processor types)          cpu.inf     Not
Available   ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\_2
```

```
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\_3
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\_4
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\_5
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\_6
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\_7
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\_8
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\_9
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\10
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\11
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\12
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\13
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\14
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\15
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\16
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\17
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\18
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\19
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\20
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\21
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\22
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\23
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\24
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\25
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\26
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\27
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\28
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\29
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\30
Processor Yes PROCESSOR 5.2.3718.0 10/1/2002
(Standard processor types) cpu.inf Not
Available ACPI\GENUINEINTEL_-
_IA64_FAMILY_31_MODEL_1\31
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\0
System Interrupt Controller Not Available
UNKNOWN Not Available Not
Available Not Available Not Available Not
Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_
00\3&267A616A&0&08
Intel 82372FB PCI to ISA Bridge Yes SYSTEM 5.2.3718.0
10/1/2002 Intel machine.inf Not
Available
PCI\VEN_8086&DEV_7600&SUBSYS_00000000&REV_
01\3&267A616A&0&10
Direct memory access controller Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available ACPI\PNP0200\4&22C34DB0&0
Programmable interrupt controller Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available ACPI\PNP0000\4&22C34DB0&0
System timer Yes SYSTEM 5.2.3718.0 10/1/2002
(Standard system devices) machine.inf
Not Available ACPI\PNP0100\4&22C34DB0&0
System speaker Yes SYSTEM 5.2.3718.0 10/1/2002
(Standard system devices) machine.inf
Not Available ACPI\PNP0800\4&22C34DB0&0
System CMOS/real time clock Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available ACPI\PNP0B00\4&22C34DB0&0
Numeric data processor Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available ACPI\PNP0C04\4&22C34DB0&0
Generic Bus Yes SYSTEM 5.2.3718.0 10/1/2002
(Standard system devices) machine.inf
Not Available ACPI\PNP0A05\4&22C34DB0&0
Extended IO Bus Yes SYSTEM 5.2.3718.0 10/1/2002
(Standard system devices) machine.inf
Not Available ACPI\PNP0A06\1
Motherboard resources Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available ACPI\PNP0C02\1
Extended IO Bus Yes SYSTEM 5.2.3718.0 10/1/2002
(Standard system devices) machine.inf
Not Available ACPI\PNP0A06\3
Motherboard resources Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available ACPI\PNP0C02\3
Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
Yes KEYBOARD 5.2.3718.0 10/1/2002
(Standard keyboards) keyboard.inf Not
Available ACPI\PNP0303\2
PS/2 Compatible Mouse Yes MOUSE 5.2.3718.0
10/1/2002 Microsoft msmouse.inf Not
Available ACPI\PNP0F13\2
Communications Port Yes PORTS 5.2.3718.0 10/1/2002
(Standard port types) msports.inf Not Available
ACPI\PNP0501\2
Extended IO Bus Yes SYSTEM 5.2.3718.0 10/1/2002
(Standard system devices) machine.inf
Not Available ACPI\PNP0A06\4
Not Available Not Available Not Available
Not Available Not Available Not
Available Not Available Not Available
ACPI\NEC4171\0
Intel(r) IA64 Bus Master IDE Controller Yes HDC
5.2.3718.0 10/1/2002 Intel mshdc.inf Not
Available
PCI\VEN_8086&DEV_7601&SUBSYS_01061033&REV_
01\3&267A616A&0&11
Primary IDE Channel Yes HDC 5.2.3718.0 10/1/2002
```

(Standard IDE ATA/ATAPI controllers) mshdc.inf
Not Available
PCIIDE\IDECHANNEL\4&31FC9C45&0&0
CD-ROM Drive Yes CDROM 5.2.3718.0 10/1/2002
(Standard CD-ROM drives) cdrom.inf Not
Available IDE\CDROM_NEC_DVD-ROM_DV-
5800A_____1.42____\5&8F26A51&0&0.0.0
Intel(r) 82372FB PCI to USB Universal Host Controller Yes
USB 5.2.3718.0 10/1/2002 Intel usbport.inf
Not Available
PCI\VEN_8086&DEV_7602&SUBSYS_01061033&REV_
01\3&267A616A&0&12
USB Root Hub Yes USB 5.2.3718.0 10/1/2002
(Standard USB Host Controller) usbport.inf Not
Available USB\ROOT_HUB\4&5C3BD33&0
Standard VGA Graphics Adapter Yes DISPLAY 5.2.3718.0
10/1/2002 (Standard display types) display.inf
Not Available
PCI\VEN_1002&DEV_4752&SUBSYS_80EB1033&REV_
27\3&267A616A&0&20
Default Monitor Yes MONITOR 5.1.2001.0 6/6/2001
(Standard monitor types) monitor.inf Not
Available
DISPLAY\DEFAULT_MONITOR\4&20D43725&0&123456
78&00&04
System Interrupt Controller Not Available
UNKNOWN Not Available Not
Available Not Available Not Available Not
Available
PCI\VEN_1033&DEV_00FD&SUBSYS_00FD1033&REV
_00\3&267A616A&0&70
PCI standard host CPU bridge Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available
PCI\VEN_1033&DEV_00F9&SUBSYS_00000000&REV_
02\3&267A616A&0&78
PCI standard host CPU bridge Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available
PCI\VEN_1033&DEV_00F9&SUBSYS_00000000&REV_
02\3&267A616A&0&80
PCI standard host CPU bridge Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available
PCI\VEN_1033&DEV_00F9&SUBSYS_00000000&REV_
02\3&267A616A&0&88
PCI standard host CPU bridge Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available
PCI\VEN_1033&DEV_00F9&SUBSYS_00000000&REV_
02\3&267A616A&0&90
PCI standard host CPU bridge Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available
PCI\VEN_1033&DEV_00F9&SUBSYS_00000000&REV_
02\3&267A616A&0&98
PCI standard host CPU bridge Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available
PCI\VEN_1033&DEV_00F9&SUBSYS_00000000&REV_
02\3&267A616A&0&A0
PCI standard host CPU bridge Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available
PCI\VEN_1033&DEV_00F9&SUBSYS_00000000&REV_
02\3&267A616A&0&A8
PCI standard host CPU bridge Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available
PCI\VEN_1033&DEV_00F9&SUBSYS_00000000&REV_
02\3&267A616A&0&B0
PCI standard host CPU bridge Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available
PCI\VEN_1033&DEV_00F9&SUBSYS_00000000&REV_
02\3&267A616A&0&B8
Communications Port Yes PORTS 5.2.3718.0 10/1/2002
(Standard port types) msports.inf Not Available
ACPI\PNP0501\1
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\1
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\2
System Interrupt Controller Not Available Not
Available Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_
00\3&1070020&0&08
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_
02\3&1070020&0&10
Qlogic processor device Yes SYSTEM 5.2.3718.0
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_
DEVICE&REV_\4&37D8710D&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\3
Adaptec SCSI Card 39160 - Ultra160 SCSI Yes
SCSIADAPTER 5.2.3718.0 10/1/2002 Adaptec
pnpscsi.inf Not Available
PCI\VEN_9005&DEV_00C0&SUBSYS_F6209005&REV_
01\3&29E81982&0&10
Adaptec SCSI Card 39160 - Ultra160 SCSI Yes
SCSIADAPTER 5.2.3718.0 10/1/2002 Adaptec
pnpscsi.inf Not Available
PCI\VEN_9005&DEV_00C0&SUBSYS_F6209005&REV_
01\3&29E81982&0&11
Qlogic processor device Yes SYSTEM 5.2.3718.0
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_GEM359&R
EV_1.06\4&1AF8FFB&0&040
Qlogic processor device Yes SYSTEM 5.2.3718.0
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_GEM359&R
EV_1.06\4&1AF8FFB&0&050
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_HITACHI&PROD_DK32DJ-
36MC&REV_G2G2\4&1AF8FFB&0&0F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\4
System Interrupt Controller Not Available Not
Available Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_
00\3&172E68DD&0&08
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_
02\3&172E68DD&0&10
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&7C3A82C&0&000
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&7C3A82C&0&001
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&7C3A82C&0&002
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&7C3A82C&0&003
Qlogic processor device Yes SYSTEM 5.2.3718.0
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_
DEVICE&REV_\4&7C3A82C&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\5
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_
02\3&474B838&0&10
Qlogic processor device Yes SYSTEM 5.2.3718.0
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_
DEVICE&REV_\4&2091DFBB&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\6
System Interrupt Controller Not Available Not
Available Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_
00\3&E44F86D&0&08
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\7
Intel(R) PRO/1000 F Server Adapter Yes NET
6.3.6.3 10/1/2002 Intel nete1000.inf
Not Available
PCI\VEN_8086&DEV_1001&SUBSYS_10038086&REV_
02\3&20FEA912&0&10
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard

system devices) machine.inf Not Available
ACPI\PNP0A03\8
System Interrupt Controller Not Available Not
Available Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_
00\3&33B859B7&0&08
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\9
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_
02\3&AD74055&0&10
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&348718C2&0&000
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&348718C2&0&001
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&348718C2&0&002
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&348718C2&0&010
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&348718C2&0&011
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&348718C2&0&012
Qlogic processor device Yes SYSTEM 5.2.3718.0
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_
DEVICE&REV_\4&348718C2&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\A
System Interrupt Controller Not Available Not
Available Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_
00\3&23C0707C&0&08
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_
02\3&23C0707C&0&10
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&B5C9978&0&000
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&B5C9978&0&001
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&B5C9978&0&002
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&B5C9978&0&010
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&B5C9978&0&011
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&B5C9978&0&012
Qlogic processor device Yes SYSTEM 5.2.3718.0
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_
DEVICE&REV_\4&B5C9978&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\B
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_
02\3&1106BFD7&0&10
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2300\4&35C513CA&0&000
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2300\4&35C513CA&0&001
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2300\4&35C513CA&0&002
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2300\4&35C513CA&0&010
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2300\4&35C513CA&0&011
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2300\4&35C513CA&0&012
Qlogic processor device Yes SYSTEM 5.2.3718.0
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_
DEVICE&REV_\4&35C513CA&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\C
System Interrupt Controller Not Available Not
Available Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_
00\3&1B2F0CE&0&08
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_
02\3&1B2F0CE&0&10
Qlogic processor device Yes SYSTEM 5.2.3718.0
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_
DEVICE&REV_\4&2D905A77&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\D
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_
02\3&146CA173&0&10
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&4BB1442&0&000
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&4BB1442&0&001
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&4BB1442&0&002
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&4BB1442&0&010
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&4BB1442&0&011
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002
(Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_
2200\4&4BB1442&0&012
Qlogic processor device Yes SYSTEM 5.2.3718.0
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_
DEVICE&REV_\4&4BB1442&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\E
System Interrupt Controller Not Available Not
Available Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_
00\3&27265218&0&08
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\F
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic

oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&39E002BD&0&10
Qlogic processor device Yes SYSTEM 5.2.3718.0 10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\4&3AE9C922&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\10
System Interrupt Controller Not Available Not Available Not Available Not Available Not Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&A985F74&0&08
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\11
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&1D521019&0&10
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&10813903&0&000
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&10813903&0&001
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&10813903&0&002
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&10813903&0&010
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&10813903&0&011
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&10813903&0&012
Qlogic processor device Yes SYSTEM 5.2.3718.0 10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\4&10813903&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\12
System Interrupt Controller Not Available Not Available Not Available Not Available Not Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&300BC0BE&0&08
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&300BC0BE&0&10
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&9DBB11B&0&000
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&9DBB11B&0&001
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&9DBB11B&0&002
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&9DBB11B&0&010
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&9DBB11B&0&011
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&9DBB11B&0&012
Qlogic processor device Yes SYSTEM 5.2.3718.0 10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\4&9DBB11B&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\13
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&72AA75C&0&10
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\4&14A4FC75&0&000
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\4&14A4FC75&0&001
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\4&14A4FC75&0&002
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\4&14A4FC75&0&010
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\4&14A4FC75&0&011
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\4&14A4FC75&0&012
Qlogic processor device Yes SYSTEM 5.2.3718.0 10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\4&14A4FC75&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\14
System Interrupt Controller Not Available Not Available Not Available Not Available Not Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&19E45801&0&08
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&19E45801&0&10
Qlogic processor device Yes SYSTEM 5.2.3718.0 10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\4&27451C9E&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\15
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&2C9E08A6&0&10
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&AB800EB&0&000
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&AB800EB&0&001
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&AB800EB&0&002
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&AB800EB&0&010
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&AB800EB&0&011
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&AB800EB&0&012
Qlogic processor device Yes SYSTEM 5.2.3718.0 10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\4&AB800EB&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\16
System Interrupt Controller Not Available Not Available Not Available Not

Available Not Available Not Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&3BCEF44&0&08
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\17
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&36B90202&0&10
Qlogic processor device Yes SYSTEM 5.2.3718.0 10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\4&293F92DE&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\18
System Interrupt Controller Not Available Not Available Not Available Not Available Not Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&23FF515D&0&08
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\19
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&1145A0B8&0&10
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&1D24E420&0&000
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&1D24E420&0&001
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&1D24E420&0&002
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&1D24E420&0&010
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&1D24E420&0&011
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&1D24E420&0&012
Qlogic processor device Yes SYSTEM 5.2.3718.0 10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\4&1D24E420&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\1A
System Interrupt Controller Not Available Not Available Not Available Not Available Not Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&D525062&0&08
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&D525062&0&10
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&731EE2F&0&000
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&731EE2F&0&001
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&731EE2F&0&002
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&731EE2F&0&010
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&731EE2F&0&011
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&731EE2F&0&012
Qlogic processor device Yes SYSTEM 5.2.3718.0 10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\4&731EE2F&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\1B
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&200C0107&0&10
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\4&9606CB9&0&000
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\4&9606CB9&0&001
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\4&9606CB9&0&002
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\4&9606CB9&0&010
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\4&9606CB9&0&011
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\4&9606CB9&0&012
Qlogic processor device Yes SYSTEM 5.2.3718.0 10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\4&9606CB9&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\1C
System Interrupt Controller Not Available Not Available Not Available Not Available Not Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_00\3&32C5B1AC&0&08
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&32C5B1AC&0&10
Qlogic processor device Yes SYSTEM 5.2.3718.0 10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\4&380107DF&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard system devices) machine.inf Not Available
ACPI\PNP0A03\1D
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\3&7B03F9A&0&10
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&37249EF2&0&000
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&37249EF2&0&001
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&37249EF2&0&002
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&37249EF2&0&010
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&37249EF2&0&011
Disk drive Yes DISKDRIVE 5.2.3718.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2200\4&37249EF2&0&012
Qlogic processor device Yes SYSTEM 5.2.3718.0

10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_
DEVICE&REV_\4&37249EF2&0&07F0
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\1E
System Interrupt Controller Not Available Not
Available Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_1033&DEV_00FE&SUBSYS_00FE1033&REV_
00\3&309158FC&0&08
PCI bus Yes SYSTEM 5.2.3718.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A03\1F
QLogic QLA23xx PCI Fibre Channel Adapter No
SCSIADAPTER 8.2.0.0 10/22/2002 QLogic
oem0.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_
02\3&1DD7A857&0&10
Qlogic processor device Yes SYSTEM 5.2.3718.0
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_
DEVICE&REV_\4&178F0609&0&07F0
ACPI Thermal Zone Yes SYSTEM 5.2.3718.0 10/1/2002
(Standard system devices) machine.inf
Not Available ACPI\THERMALZONE\PROC
ACPI Thermal Zone Yes SYSTEM 5.2.3718.0 10/1/2002
(Standard system devices) machine.inf
Not Available ACPI\THERMALZONE\CELL
ACPI Thermal Zone Yes SYSTEM 5.2.3718.0 10/1/2002
(Standard system devices) machine.inf
Not Available ACPI\THERMALZONE\XBC_
ACPI Thermal Zone Yes SYSTEM 5.2.3718.0 10/1/2002
(Standard system devices) machine.inf
Not Available ACPI\THERMALZONE\PCXA
ACPI Fixed Feature Button Yes SYSTEM 5.2.3718.0
10/1/2002 (Standard system devices) machine.inf
Not Available
ACPI\FIXEDBUTTON\2&DABA3FF&0
Logical Disk Manager Yes SYSTEM 5.2.3718.0 10/1/2002
(Standard system devices) machine.inf
Not Available ROOT\DMIO\0000
Volume Manager Yes SYSTEM 5.2.3718.0 10/1/2002
(Standard system devices) machine.inf
Not Available ROOT\FTDISK\0000
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4
9679D00-A773-01C2-507B-9E5F8078F531}
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4
9B6F580-A773-01C2-F1B3-12714F758821}
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3
15596A0-FBAA-01C2-A1F4-04622FD5EC6D}
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35F4OFFSET7E00LENGTH320052FC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35F4OFFSET3200537A00LENGTH8344B7800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35F7OFFSET7E00LENGTH320052FC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35F7OFFSET3200537A00LENGTH8344B7800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35F1OFFSET7E00LENGTH320052FC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35F1OFFSET3200537A00LENGTH8344B7800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35F0OFFSET7E00LENGTH320052FC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35F0OFFSET3200537A00LENGTH8344B7800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FD5OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FD5OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FD5OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FD4OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FD4OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FD4OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFC2OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFC2OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFC2OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FD9OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FD9OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FD9OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35F9OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35F9OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35F9OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FDBOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FDBOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FDBOFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35E5OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35E5OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35E5OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35E4OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35E4OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35E4OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002

Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35E7OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35E7OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
35E7OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE171F
0A8BOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE171F
0A8BOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE171F
0A8BOFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFC0OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFC0OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFC0OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE30C
DC405OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE30C
DC405OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE30C
DC405OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CCCOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CCCOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CCCOFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFC7OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFC7OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFC7OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CCEOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CCEOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CCEOFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CCFOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CCFOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CCFOFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC8OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC8OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC8OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC9OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC9OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC9OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
206COFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
206COFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
206COFFSETDC01A1800LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
206COFFSET16801F4A00LENGTH663AD48A00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9560
A285OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9560
A285OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9560
A285OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE8E12
1BE9OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE8E12
1BE9OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE8E12
1BE9OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREA049
72FCOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREA049
72FCOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREA049
72FCOFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2063OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2063OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2063OFFSETDC01A1800LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF

2063OFFSET16801F4A00LENGTH663AD48A00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREA049
72FEOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREA049
72FEOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREA049
72FEOFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFC8OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFC8OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFC8OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9B49
2B5BOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9B49
2B5BOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9B49
2B5BOFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFCFOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFCFOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFCFOFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9560
A28DOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9560
A28DOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9560
A28DOFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2060OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2060OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2060OFFSETDC01A1800LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2060OFFSET16801F4A00LENGTH663AD48A00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FA2OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FA2OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FA2OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FA6OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FA6OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FA6OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE30C
DC40FOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE30C
DC40FOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE30C
DC40FOFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFCEOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFCEOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFCEOFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
3403OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
3403OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
3403OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
3402OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
3402OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
3402OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFCCOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFCCOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFCCOFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CCAOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CCAOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CCAOFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CCBOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CCBOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CCBOFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002

Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CD4OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CD4OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CD4OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CD5OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CD5OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CD5OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CD6OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CD6OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CD6OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CD7OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CD7OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CD7OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFC5OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFC5OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFC5OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2062OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2062OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2062OFFSETDC01A1800LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2062OFFSET16801F4A00LENGTH663AD48A00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2061OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2061OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2061OFFSETDC01A1800LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2061OFFSET16801F4A00LENGTH663AD48A00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE30C
DC401OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE30C
DC401OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE30C
DC401OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFCBOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFCBOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFCBOFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFCAOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFCAOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEE3
CFFCAOFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2068OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2068OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2068OFFSETDC01A1800LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
2068OFFSET16801F4A00LENGTH663AD48A00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FCBOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FCBOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FCBOFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FCAOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FCAOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FCAOFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FCCOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FCCOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FCCOFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE476A
4315OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE476A

4315OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE476A
4315OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FB1OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FB1OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FB1OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FDEOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FDEOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FDEOFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
206AOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
206AOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
206AOFFSETDC01A1800LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURED2FF
206AOFFSET16801F4A00LENGTH663AD48A00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE4E4F
F72OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE4E4F
F72OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE4E4F
F72OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE3CA3
8C3FOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE3CA3
8C3FOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE3CA3
8C3FOFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREA049
72FDOFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREA049
72FDOFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREA049
72FDOFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FC6OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FC6OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FC6OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC5OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC5OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC5OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC6OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC6OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC6OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC7OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC7OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC7OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC0OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC0OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC0OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC2OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC2OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC2OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC3OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC3OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE11D1
1CC3OFFSETDC01A1800LENGTH6EFAD9BC00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE76FC
E9F5OFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE76FC
E9F5OFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE76FC
E9F5OFFSET1400D11A00LENGTH68BA22BA00
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE476A
431COFFSET7E00LENGTH500146800
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002
Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE476A
431COFFSET50014E600LENGTH8C0053200
Generic volume Yes VOLUME 5.2.3718.0 10/1/2002

```
        Microsoft   volume.inf  Not Available
        STORAGE\VOLUME\1&30A96598&0&SIGNATURE476A
431COFFSET1400D11A00LENGTH68BA22BA00
Generic volume      Yes         VOLUME   5.2.3718.0  10/1/2002
        Microsoft   volume.inf  Not Available
        STORAGE\VOLUME\1&30A96598&0&SIGNATURE476A
431FOFFSET7E00LENGTH500146800
Generic volume      Yes         VOLUME   5.2.3718.0  10/1/2002
        Microsoft   volume.inf  Not Available
        STORAGE\VOLUME\1&30A96598&0&SIGNATURE476A
431FOFFSET50014E600LENGTH8C0053200
Generic volume      Yes         VOLUME   5.2.3718.0  10/1/2002
        Microsoft   volume.inf  Not Available
        STORAGE\VOLUME\1&30A96598&0&SIGNATURE476A
431FOFFSET1400D11A00LENGTH68BA22BA00
Generic volume      Yes         VOLUME   5.2.3718.0  10/1/2002
        Microsoft   volume.inf  Not Available
        STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
3415OFFSET7E00LENGTH500146800
Generic volume      Yes         VOLUME   5.2.3718.0  10/1/2002
        Microsoft   volume.inf  Not Available
        STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
3415OFFSET50014E600LENGTH8C0053200
Generic volume      Yes         VOLUME   5.2.3718.0  10/1/2002
        Microsoft   volume.inf  Not Available
        STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
3415OFFSET1400D11A00LENGTH68BA22BA00
Generic volume      Yes         VOLUME   5.2.3718.0  10/1/2002
        Microsoft   volume.inf  Not Available
        STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
3417OFFSET7E00LENGTH500146800
Generic volume      Yes         VOLUME   5.2.3718.0  10/1/2002
        Microsoft   volume.inf  Not Available
        STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
3417OFFSET50014E600LENGTH8C0053200
Generic volume      Yes         VOLUME   5.2.3718.0  10/1/2002
        Microsoft   volume.inf  Not Available
        STORAGE\VOLUME\1&30A96598&0&SIGNATUREE64F
3417OFFSET1400D11A00LENGTH68BA22BA00
Generic volume      Yes         VOLUME   5.2.3718.0  10/1/2002
        Microsoft   volume.inf  Not Available
        STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FD3OFFSET7E00LENGTH500146800
Generic volume      Yes         VOLUME   5.2.3718.0  10/1/2002
        Microsoft   volume.inf  Not Available
        STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FD3OFFSET50014E600LENGTH8C0053200
Generic volume      Yes         VOLUME   5.2.3718.0  10/1/2002
        Microsoft   volume.inf  Not Available
        STORAGE\VOLUME\1&30A96598&0&SIGNATURE65BF
2FD3OFFSET1400D11A00LENGTH68BA22BA00
AFD Networking Support Environment      Not Available
        LEGACYDRIVER    Not Available       Not
Available   Not Available   Not Available   Not
Available   ROOT\LEGACY_AFD\0000
Beep        Not Available   LEGACYDRIVER    Not
Available   Not Available   Not Available   Not
Available   Not Available   ROOT\LEGACY_BEEP\0000
CRC Disk Filter Driver  Not Available   LEGACYDRIVER
        Not Available   Not Available   Not
Available   Not Available   Not Available
        ROOT\LEGACY_CRCDISK\0000
dmboot      Not Available   LEGACYDRIVER    Not
Available   Not Available   Not Available   Not
Available   Not Available   ROOT\LEGACY_DMBOOT\0000
dmload      Not Available   LEGACYDRIVER    Not
Available   Not Available   Not Available   Not
Available   Not Available   ROOT\LEGACY_DMLOAD\0000
Fips        Not Available   LEGACYDRIVER    Not
Available   Not Available   Not Available   Not
Available   Not Available   ROOT\LEGACY_FIPS\0000
Generic Packet Classifier       Not Available
        LEGACYDRIVER    Not Available       Not
Available   Not Available   Not Available   Not
Available   ROOT\LEGACY_GPC\0000
IPSEC driver        Not Available   LEGACYDRIVER
        Not Available   Not Available   Not
Available   Not Available   Not Available
        ROOT\LEGACY_IPSEC\0000
ksecdd      Not Available   LEGACYDRIVER    Not
Available   Not Available   Not Available   Not
Available   Not Available   ROOT\LEGACY_KSECDD\0000
mountmgr    Not Available   LEGACYDRIVER    Not
Available   Not Available   Not Available   Not
Available   Not Available
        ROOT\LEGACY_MOUNTMGR\0000
NDIS System Driver  Not Available   LEGACYDRIVER
        Not Available   Not Available   Not
Available   Not Available   Not Available
        ROOT\LEGACY_NDIS\0000
Remote Access NDIS TAPI Driver  Not Available
        LEGACYDRIVER    Not Available       Not
Available   Not Available   Not Available   Not
Available   ROOT\LEGACY_NDISTAPI\0000
NDIS Usermode I/O Protocol      Not Available
        LEGACYDRIVER    Not Available       Not
Available   Not Available   Not Available   Not
Available   ROOT\LEGACY_NDISUIO\0000
NDProxy     Not Available   LEGACYDRIVER    Not
Available   Not Available   Not Available   Not
Available   Not Available   ROOT\LEGACY_NDPROXY\0000
NetBios over Tcpip  Not Available   LEGACYDRIVER
        Not Available   Not Available   Not
Available   Not Available   Not Available
        ROOT\LEGACY_NETBT\0000
Null        Not Available   LEGACYDRIVER    Not
Available   Not Available   Not Available   Not
Available   Not Available   ROOT\LEGACY_NULL\0000
Partition Manager   Not Available   LEGACYDRIVER
        Not Available   Not Available   Not
Available   Not Available   Not Available
        ROOT\LEGACY_PARTMGR\0000
qldirect    Not Available   LEGACYDRIVER    Not
Available   Not Available   Not Available   Not
Available   Not Available   ROOT\LEGACY_QLDIRECT\0000
qlvika      Not Available   LEGACYDRIVER    Not
Available   Not Available   Not Available   Not
Available   Not Available   ROOT\LEGACY_QLVIKA\0000
Remote Access Auto Connection Driver    Not Available
        LEGACYDRIVER    Not Available       Not
Available   Not Available   Not Available   Not
Available   ROOT\LEGACY_RASACD\0000
RDPCDD      Not Available   LEGACYDRIVER    Not
Available   Not Available   Not Available   Not
Available   Not Available   ROOT\LEGACY_RDPCDD\0000
sacdrv      Not Available   LEGACYDRIVER    Not
Available   Not Available   Not Available   Not
Available   Not Available   ROOT\LEGACY_SACDRV\0000
Tbs         Not Available   LEGACYDRIVER    Not
Available   Not Available   Not Available   Not
Available   Not Available   ROOT\LEGACY_TBS\0000
TCP/IP Protocol Driver  Not Available   LEGACYDRIVER
        Not Available   Not Available   Not
Available   Not Available   Not Available
        ROOT\LEGACY_TCPIP\0000
volsnap     Not Available   LEGACYDRIVER    Not
Available   Not Available   Not Available   Not
Available   Not Available   ROOT\LEGACY_VOLSNAP\0000
Remote Access IP ARP Driver     Not Available
        LEGACYDRIVER    Not Available       Not
Available   Not Available   Not Available   Not
Available   ROOT\LEGACY_WANARP\0000
Audio Codecs        Yes     MEDIA   5.2.3718.0  10/1/2002
        (Standard system devices)   wave.inf    Not
Available   ROOT\MEDIA\MS_MMACM
Legacy Audio Drivers    Yes     MEDIA   5.2.3718.0  10/1/2002
        (Standard system devices)   wave.inf    Not
Available   ROOT\MEDIA\MS_MMDRV
Media Control Devices   Yes     MEDIA   5.2.3718.0  10/1/2002
        (Standard system devices)   wave.inf    Not
Available   ROOT\MEDIA\MS_MMMCI
Legacy Video Capture Devices    Yes     MEDIA   5.2.3718.0
        10/1/2002   (Standard system devices)   wave.inf
        Not Available   ROOT\MEDIA\MS_MMVCD
Video Codecs        Yes     MEDIA   5.2.3718.0  10/1/2002
        (Standard system devices)   wave.inf    Not
Available   ROOT\MEDIA\MS_MMVID
WAN Miniport (L2TP)     Yes     NET     5.2.3718.0  10/1/2002
        Microsoft   netrasa.inf Not Available
        ROOT\MS_L2TPMINIPORT\0000
WAN Miniport (IP)       Yes     NET     5.2.3718.0  10/1/2002
        Microsoft   netrasa.inf Not Available
        ROOT\MS_NDISWANIP\0000
WAN Miniport (PPPOE)        Yes     NET     5.2.3718.0
        10/1/2002   Microsoft   netrasa.inf Not Available
        ROOT\MS_PPPOEMINIPORT\0000
WAN Miniport (PPTP)     Yes     NET     5.2.3718.0  10/1/2002
        Microsoft   netrasa.inf Not Available
        ROOT\MS_PPTPMINIPORT\0000
Direct Parallel     Yes     NET     5.2.3718.0  10/1/2002
        Microsoft   netrasa.inf Not Available
        ROOT\MS_PTIMINIPORT\0000
Terminal Server Device Redirector   Yes     SYSTEM  5.2.3718.0
        10/1/2002   (Standard system devices)   machine.inf
        Not Available   ROOT\RDPDR\0000
Terminal Server Keyboard Driver     Yes     SYSTEM  5.2.3718.0
        10/1/2002   (Standard system devices)   machine.inf
        Not Available   ROOT\RDP_KBD\0000
Terminal Server Mouse Driver    Yes     SYSTEM  5.2.3718.0
        10/1/2002   (Standard system devices)   machine.inf
        Not Available   ROOT\RDP_MOU\0000
Plug and Play Software Device Enumerator    Yes     SYSTEM
        5.2.3718.0  10/1/2002   (Standard system devices)
        machine.inf     Not Available
        ROOT\SYSTEM\0000
```

[Environment Variables]

Variable Value User Name
ComSpec %SystemRoot%\system32\cmd.exe
<SYSTEM>
Path
%SystemRoot%\system32;%SystemRoot%;%SystemRoo
t%\System32\Wbem;C:\Program Files\Microsoft SQL
Server\80\Tools\Binn\ <SYSTEM>
windir %SystemRoot% <SYSTEM>
OS Windows_NT <SYSTEM>
PROCESSOR_ARCHITECTURE IA64 <SYSTEM>
PROCESSOR_LEVEL 31 <SYSTEM>
PROCESSOR_IDENTIFIER ia64 Family 31 Model 1 Stepping
5, GenuineIntel <SYSTEM>
PROCESSOR_REVISION 0105 <SYSTEM>
NUMBER_OF_PROCESSORS 32 <SYSTEM>
ClusterLog C:\WINDOWS\Cluster\cluster.log <SYSTEM>
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
<SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>
INTEL_LICENSE_FILE C:\Program Files (x86)\Common
Files\Intel\Licenses <SYSTEM>
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\SYSTEM
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\SYSTEM
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\LOCAL SERVICE
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\LOCAL SERVICE
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\NETWORK SERVICE
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\NETWORK SERVICE
TEMP %USERPROFILE%\Local Settings\Temp
ASAMA\Administrator
TMP %USERPROFILE%\Local Settings\Temp
ASAMA\Administrator

[Print Jobs]

Document Size Owner Notify Status Time
Submitted Start Time Until Time Elapsed Time Pages
Printed Job ID Priority Parameters Driver
Print Processor Host Print Queue Data Type
Name

[Network Connections]

Local Name Remote Name Type Status
User Name
B: \\bmaster\g$ Disk Persistent Connection

[Running Tasks]

Name Path Process ID Priority Min Working Set
Max Working Set Start Time Version Size
File Date
system idle process Not Available 0 0
Not Available Not Available Not
Available Not Available Not Available Not
Available
system Not Available 4 8 0
2826240 Not Available Not Available
Not Available Not Available
smss.exe c:\windows\system32\smss.exe 444 11
409600 2826240 4/20/2003 4:37 PM 5.2.3718.0
(dnsrv.021114-1947) 128.50 KB (131,584 bytes) 11/18/2002
9:00 PM
csrss.exe Not Available 500 13 Not
Available Not Available 4/20/2003 4:43 PM Not
Available Not Available Not Available
winlogon.exe c:\windows\system32\winlogon.exe
524 13 409600 2826240 4/20/2003
4:43 PM 5.2.3718.0 (dnsrv.021114-1947) 614.50 KB (629,248
bytes) 11/18/2002 9:00 PM
services.exe c:\windows\system32\services.exe 568
9 409600 2826240 4/20/2003 4:43 PM
5.2.3718.0 (dnsrv.021114-1947) 285.50 KB (292,352
bytes) 11/18/2002 9:00 PM
lsass.exe c:\windows\system32\lsass.exe 580 9
409600 2826240 4/20/2003 4:43 PM 5.2.3718.0
(dnsrv.021114-1947) 15.00 KB (15,360 bytes) 11/18/2002
9:00 PM
svchost.exe c:\windows\system32\svchost.exe 796
8 409600 2826240 4/20/2003 4:43 PM
5.2.3718.0 (dnsrv.021114-1947) 32.00 KB (32,768
bytes) 11/18/2002 9:00 PM
svchost.exe Not Available 872 8
Not Available Not Available 4/20/2003
4:43 PM Not Available Not Available Not
Available
svchost.exe Not Available 920 8
Not Available Not Available 4/20/2003
4:43 PM Not Available Not Available Not
Available
svchost.exe c:\windows\system32\svchost.exe 936
8 409600 2826240 4/20/2003 4:43 PM
5.2.3718.0 (dnsrv.021114-1947) 32.00 KB (32,768
bytes) 11/18/2002 9:00 PM
spoolsv.exe c:\windows\system32\spoolsv.exe 1100
8 409600 2826240 4/20/2003 4:43 PM
5.2.3718.0 (dnsrv.021114-1947) 154.00 KB (157,696
bytes) 11/18/2002 9:00 PM
msdtc.exe Not Available 1152 8 Not
Available Not Available 4/20/2003 4:43 PM Not
Available Not Available Not Available
svchost.exe c:\windows\system32\svchost.exe 1620
8 409600 2826240 4/20/2003 4:43 PM
5.2.3718.0 (dnsrv.021114-1947) 32.00 KB (32,768
bytes) 11/18/2002 9:00 PM
svchost.exe Not Available 1984 8
Not Available Not Available 4/20/2003
4:43 PM Not Available Not Available Not
Available
mssearch.exe c:\program files\common
files\system\mssearch\bin\mssearch.exe 2032 8
409600 2826240 4/20/2003 4:43 PM
9.107.8320.0 460.00 KB (471,040 bytes)
4/9/2003 2:54 AM
dfssvc.exe c:\windows\system32\dfssvc.exe 276 8
409600 2826240 4/20/2003 4:43 PM 5.2.3718.0
(dnsrv.021114-1947) 430.00 KB (440,320 bytes) 11/18/2002
9:00 PM
svchost.exe c:\windows\system32\svchost.exe 380
8 409600 2826240 4/20/2003 4:43 PM
5.2.3718.0 (dnsrv.021114-1947) 32.00 KB (32,768
bytes) 11/18/2002 9:00 PM
explorer.exe c:\windows\explorer.exe 1148
8 409600 2826240 4/20/2003 4:44 PM
6.00.3718.0 (dnsrv.021114-1947) 1.62 MB (1,701,376
bytes) 11/18/2002 9:00 PM
ctfmon.exe c:\windows\system32\ctfmon.exe 1992 8
409600 2826240 4/20/2003 4:44 PM 5.2.3718.0
(dnsrv.021114-1947) 31.50 KB (32,256 bytes) 11/18/2002
9:00 PM
sqlmangr.exe c:\program files\microsoft sql
server\80\tools\binn\sqlmangr.exe 2056 8 409600
2826240 4/20/2003 4:44 PM 2000.080.0760.00
225.00 KB (230,400 bytes) 2/7/2003 6:48 AM
ctfmon.exe c:\windows\syswow64\ctfmon.exe 2068 8
409600 2826240 4/20/2003 4:44 PM 5.2.3718.0
(dnsrv.021114-1947) 13.50 KB (13,824 bytes) 11/18/2002
9:00 PM
wmiprvse.exe Not Available 2140 8
Not Available Not Available 4/20/2003
4:44 PM Not Available Not Available Not
Available
wmiprvse.exe Not Available 3140 8
Not Available Not Available 4/20/2003
4:53 PM Not Available Not Available Not
Available
wmiprvse.exe Not Available 3200 8
Not Available Not Available 4/20/2003
4:54 PM Not Available Not Available Not
Available
helpctr.exe c:\windows\pchealth\helpctr\binaries\helpctr.exe
3312 8 409600 2826240 4/20/2003
4:57 PM 5.2.3718.0 (dnsrv.021114-1947) 1.97 MB (2,067,456
bytes) 4/5/2003 8:05 PM
helpsvc.exe
c:\windows\pchealth\helpctr\binaries\helpsvc.exe
3380 8 409600 2826240 4/20/2003
4:57 PM 5.2.3718.0 (dnsrv.021114-1947) 2.18 MB (2,288,640
bytes) 4/5/2003 8:05 PM

[Loaded Modules]

Name Version Size File Date Manufacturer
Path
smss 5.2.3718.0 (dnsrv.021114-1947) 128.50 KB (131,584
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\smss.exe
ntdll 5.2.3718.0 (dnsrv.021114-1947) 1.45 MB (1,523,712
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\ntdll.dll
winlogon 5.2.3718.0 (dnsrv.021114-1947) 614.50 KB (629,248
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\winlogon.exe
kernel32 5.2.3718.0 (dnsrv.021114-1947) 1.76 MB (1,848,320
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\kernel32.dll
msvcrt 7.0.3718.0 (dnsrv.021114-1947) 873.50 KB (894,464

bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\msvcrt.dll
advapi32 5.2.3718.0 (dnsrv.021114-1947) 1.30 MB (1,358,336 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\advapi32.dll
rpcrt4 5.2.3718.0 (dnsrv.021114-1947) 2.01 MB (2,112,000 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\rpcrt4.dll
user32 5.2.3718.0 (dnsrv.021114-1947) 1.31 MB (1,372,672 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\user32.dll
gdi32 5.2.3718.0 (dnsrv.021114-1947) 783.50 KB (802,304 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\gdi32.dll
userenv 5.2.3718.0 (dnsrv.021114-1947) 1.46 MB (1,535,488 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\userenv.dll
nddeapi 5.2.3718.0 (dnsrv.021114-1947) 39.50 KB (40,448 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\nddeapi.dll
crypt32 5.131.3718.0 (dnsrv.021114-1947) 1.50 MB (1,574,400 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\crypt32.dll
msasn1 5.2.3718.0 (dnsrv.021114-1947) 153.50 KB (157,184 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\msasn1.dll
secur32 5.2.3718.0 (dnsrv.021114-1947) 166.00 KB (169,984 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\secur32.dll
winsta 5.2.3718.0 (dnsrv.021114-1947) 138.50 KB (141,824 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\winsta.dll
netapi32 5.2.3718.0 (dnsrv.021114-1947) 832.00 KB (851,968 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\netapi32.dll
profmap 5.2.3718.0 (dnsrv.021114-1947) 55.50 KB (56,832 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\profmap.dll
regapi 5.2.3718.0 (dnsrv.021114-1947) 124.00 KB (126,976 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\regapi.dll
ws2_32 5.2.3718.0 (dnsrv.021114-1947) 228.50 KB (233,984 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\ws2_32.dll
ws2help 5.2.3718.0 (dnsrv.021114-1947) 49.50 KB (50,688 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\ws2help.dll
msgina 5.2.3718.0 (dnsrv.021114-1947) 1.91 MB (2,007,040 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\msgina.dll
shsvcs 6.00.3718.0 (dnsrv.021114-1947) 321.50 KB (329,216 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\shsvcs.dll
shlwapi 6.00.3718.0 (dnsrv.021114-1947) 720.00 KB (737,280 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\shlwapi.dll
sfc 5.2.3718.0 (dnsrv.021114-1947) 7.50 KB (7,680 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\sfc.dll
sfc_os 5.2.3718.0 (dnsrv.021114-1947) 257.00 KB (263,168 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\sfc_os.dll
wintrust 5.131.3718.0 (dnsrv.021114-1947) 452.00 KB (462,848 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\wintrust.dll
ole32 5.2.3718.0 (dnsrv.021114-1947) 3.38 MB (3,547,648 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\ole32.dll
imagehlp 5.2.3718.0 (dnsrv.021114-1947) 128.00 KB (131,072 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\imagehlp.dll
comctl32 6.0 (dnsrv.021114-1947) 2.18 MB (2,285,056 bytes) 4/6/2003 4:42 AM Microsoft Corporation c:\windows\winsxs\ia64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.100.0_x-ww_b3722bab\comctl32.dll
winscard 5.2.3718.0 (dnsrv.021114-1947) 291.50 KB (298,496 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\winscard.dll
wtsapi32 5.2.3718.0 (dnsrv.021114-1947) 47.00 KB (48,128 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\wtsapi32.dll
version 5.2.3718.0 (dnsrv.021114-1947) 44.00 KB (45,056 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\version.dll
sxs 5.2.3718.0 (dnsrv.021114-1947) 1.77 MB (1,860,096 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\sxs.dll
shell32 6.00.3718.0 (dnsrv.021114-1947) 12.35 MB (12,951,040 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\shell32.dll
setupapi 5.2.3718.0 (dnsrv.021114-1947) 1.90 MB (1,990,656 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\setupapi.dll
wldap32 5.2.3718.0 (dnsrv.021114-1947) 402.00 KB (411,648 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\wldap32.dll
cscdll 5.2.3718.0 (dnsrv.021114-1947) 210.50 KB (215,552 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\cscdll.dll
wlnotify 5.2.3718.0 (dnsrv.021114-1947) 218.00 KB (223,232 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\wlnotify.dll
winmm 5.2.3718.0 (dnsrv.021114-1947) 404.00 KB (413,696 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\winmm.dll
winspool 5.2.3718.0 (dnsrv.021114-1947) 399.00 KB (408,576 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\winspool.drv
mpr 5.2.3718.0 (dnsrv.021114-1947) 163.00 KB (166,912 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\mpr.dll
comctl32 5.82 (dnsrv.021114-1947) 1.55 MB (1,621,504 bytes) 4/6/2003 4:41 AM Microsoft Corporation c:\windows\winsxs\ia64_microsoft.windows.common-controls_6595b64144ccf1df_5.82.0.0_x-ww_b9c4a0a5\comctl32.dll
uxtheme 6.00.3718.0 (dnsrv.021114-1947) 527.50 KB (540,160 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\uxtheme.dll
samlib 5.2.3718.0 (dnsrv.021114-1947) 106.00 KB (108,544 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\samlib.dll
cscui 5.2.3718.0 (dnsrv.021114-1947) 574.00 KB (587,776 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\cscui.dll
drprov 5.2.3718.0 (dnsrv.021114-1947) 26.50 KB (27,136 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\drprov.dll
ntlanman 5.2.3718.0 (dnsrv.021114-1947) 108.00 KB (110,592 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\ntlanman.dll
netui0 5.2.3718.0 (dnsrv.021114-1947) 181.50 KB (185,856 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\netui0.dll
netui1 5.2.3718.0 (dnsrv.021114-1947) 482.00 KB (493,568 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\netui1.dll
davclnt 5.2.3718.0 (dnsrv.021114-1947) 59.00 KB (60,416 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\davclnt.dll
mprui 5.2.3718.0 (dnsrv.021114-1947) 97.00 KB (99,328 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\mprui.dll
netui2 5.2.3718.0 (dnsrv.021114-1947) 761.50 KB (779,776 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\netui2.dll
comdlg32 6.00.3718.0 (dnsrv.021114-1947) 706.00 KB (722,944 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\comdlg32.dll
netmsg 5.2.3718.0 (dnsrv.021114-1947) 177.50 KB (181,760 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\netmsg.dll
oleaut32 5.2.3718.0 3.56 MB (3,730,432 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\oleaut32.dll
ntmarta 5.2.3718.0 (dnsrv.021114-1947) 343.50 KB (351,744 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\ntmarta.dll
clbcatq 2001.12.4648.0 (dnsrv.021114-1947) 1.23 MB (1,293,312 bytes) 4/5/2003 8:02 PM Microsoft Corporation c:\windows\system32\clbcatq.dll
comres 2001.12.4648.0 (dnsrv.021114-1947) 779.50 KB (798,208 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\comres.dll
services 5.2.3718.0 (dnsrv.021114-1947) 285.50 KB (292,352 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\services.exe
scesrv 5.2.3718.0 (dnsrv.021114-1947) 762.50 KB (780,800 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\scesrv.dll
authz 5.2.3718.0 (dnsrv.021114-1947) 201.50 KB (206,336 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\authz.dll
umpnpmgr 5.2.3718.0 (dnsrv.021114-1947) 317.00 KB (324,608 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\umpnpmgr.dll
ncobjapi 5.2.3718.0 (dnsrv.021114-1947) 118.50 KB (121,344 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\ncobjapi.dll
msvcp60 6.10.2240.8 941.50 KB (964,096 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\msvcp60.dll
eventlog 5.2.3718.0 (dnsrv.021114-1947) 155.50 KB (159,232 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\eventlog.dll
psapi 5.2.3718.0 (dnsrv.021114-1947) 48.00 KB (49,152 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\psapi.dll

lsass 5.2.3718.0 (dnsrv.021114-1947) 15.00 KB (15,360 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\lsass.exe
lsasrv 5.2.3718.0 (dnsrv.021114-1947) 1.94 MB (2,031,104 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\lsasrv.dll
samsrv 5.2.3718.0 (dnsrv.021114-1947) 1,016.00 KB (1,040,384 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\samsrv.dll
cryptdll 5.2.3718.0 (dnsrv.021114-1947) 58.50 KB (59,904 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\cryptdll.dll
dnsapi 5.2.3718.0 (dnsrv.021114-1947) 402.00 KB (411,648 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\dnsapi.dll
ntdsapi 5.2.3718.0 (dnsrv.021114-1947) 182.50 KB (186,880 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\ntdsapi.dll
msprivs 5.2.3718.0 (dnsrv.021114-1947) 45.00 KB (46,080 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\msprivs.dll
kerberos 5.2.3718.0 (dnsrv.021114-1947) 874.50 KB (895,488 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\kerberos.dll
msv1_0 5.2.3718.0 (dnsrv.021114-1947) 329.50 KB (337,408 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\msv1_0.dll
netlogon 5.2.3718.0 (dnsrv.021114-1947) 935.50 KB (957,952 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\netlogon.dll
w32time 5.2.3718.0 (dnsrv.021114-1947) 540.50 KB (553,472 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\w32time.dll
iphlpapi 5.2.3718.0 (dnsrv.021114-1947) 223.00 KB (228,352 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\iphlpapi.dll
schannel 5.2.3718.0 (dnsrv.021114-1947) 468.50 KB (479,744 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\schannel.dll
wdigest 5.2.3718.0 (dnsrv.021114-1947) 161.50 KB (165,376 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\wdigest.dll
rsaenh 5.2.3718.0 (dnsrv.021114-1947) 372.07 KB (381,000 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\rsaenh.dll
rassfm 5.2.3718.0 (dnsrv.021114-1947) 56.00 KB (57,344 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\rassfm.dll
kdcsvc 5.2.3718.0 (dnsrv.021114-1947) 570.00 KB (583,680 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\kdcsvc.dll
ntdsa 5.2.3718.0 (dnsrv.021114-1947) 3.69 MB (3,869,696 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\ntdsa.dll
ntdsatq 5.2.3718.0 (dnsrv.021114-1947) 79.50 KB (81,408 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\ntdsatq.dll
mswsock 5.2.3718.0 (dnsrv.021114-1947) 670.00 KB (686,080 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\mswsock.dll
esent 5.2.3718.0 (dnsrv.021114-1947) 2.48 MB (2,598,912 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\esent.dll
scecli 5.2.3718.0 (dnsrv.021114-1947) 468.00 KB (479,232 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\scecli.dll
wshtcpip 5.2.3718.0 (dnsrv.021114-1947) 38.00 KB (38,912 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\wshtcpip.dll
ipsecsvc 5.2.3718.0 (dnsrv.021114-1947) 410.50 KB (420,352 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\ipsecsvc.dll
oakley 5.2.3718.0 (dnsrv.021114-1947) 493.50 KB (505,344 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\oakley.dll
winipsec 5.2.3718.0 (dnsrv.021114-1947) 78.50 KB (80,384 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\winipsec.dll
pstorsvc 5.2.3718.0 (dnsrv.021114-1947) 57.50 KB (58,880 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\pstorsvc.dll
psbase 5.2.3718.0 (dnsrv.021114-1947) 172.00 KB (176,128 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\psbase.dll
dssenh 5.2.3718.0 (dnsrv.021114-1947) 319.07 KB (326,728 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\dssenh.dll
wlbsctrl 5.2.3718.0 (dnsrv.021114-1947) 194.50 KB (199,168 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\wlbsctrl.dll
svchost 5.2.3718.0 (dnsrv.021114-1947) 32.00 KB (32,768 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\svchost.exe
rpcss 5.2.3718.0 (dnsrv.021114-1947) 645.00 KB (660,480 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\rpcss.dll
wzcsvc 5.2.3718.0 (dnsrv.021114-1947) 608.50 KB (623,104 bytes) 11/15/2002 11:37 PM Microsoft Corporation c:\windows\system32\wzcsvc.dll
rtutils 5.2.3718.0 (dnsrv.021114-1947) 81.50 KB (83,456 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\rtutils.dll
wmi 5.2.3718.0 (dnsrv.021114-1947) 5.00 KB (5,120 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\wmi.dll
dhcpcsvc 5.2.3718.0 (dnsrv.021114-1947) 279.00 KB (285,696 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\dhcpcsvc.dll
rastls 5.2.3718.0 (dnsrv.021114-1947) 346.00 KB (354,304 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\rastls.dll
atl 3.00.2282 348.00 KB (356,352 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\atl.dll
cryptui 5.131.3718.0 (dnsrv.021114-1947) 1.04 MB (1,094,656 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\cryptui.dll
mprapi 5.2.3718.0 (dnsrv.021114-1947) 237.50 KB (243,200 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\mprapi.dll
activeds 5.2.3718.0 (dnsrv.021114-1947) 544.50 KB (557,568 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\activeds.dll
adsldpc 5.2.3718.0 (dnsrv.021114-1947) 312.00 KB (319,488 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\adsldpc.dll
credui 5.2.3718.0 (dnsrv.021114-1947) 288.00 KB (294,912 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\credui.dll
rasapi32 5.2.3718.0 (dnsrv.021114-1947) 585.00 KB (599,040 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\rasapi32.dll
rasman 5.2.3718.0 (dnsrv.021114-1947) 154.50 KB (158,208 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\rasman.dll
tapi32 5.2.3718.0 (dnsrv.021114-1947) 493.00 KB (504,832 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\tapi32.dll
raschap 5.2.3718.0 (dnsrv.021114-1947) 200.00 KB (204,800 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\raschap.dll
schedsvc 5.2.3718.0 (dnsrv.021114-1947) 519.50 KB (531,968 bytes) 4/5/2003 8:04 PM Microsoft Corporation c:\windows\system32\schedsvc.dll
msidle 6.00.3718.0 (dnsrv.021114-1947) 8.50 KB (8,704 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\msidle.dll
wkssvc 5.2.3718.0 (dnsrv.021114-1947) 304.00 KB (311,296 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\wkssvc.dll
wiarpc 5.2.3718.0 (dnsrv.021114-1947) 68.00 KB (69,632 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\wiarpc.dll
cryptsvc 5.2.3718.0 (dnsrv.021114-1947) 126.00 KB (129,024 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\cryptsvc.dll
certcli 5.2.3718.0 (dnsrv.021114-1947) 586.00 KB (600,064 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\certcli.dll
vssapi 5.2.3718.0 (dnsrv.021114-1947) 1.28 MB (1,337,344 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\vssapi.dll
dmserver 5.2.3718.0 (dnsrv.021114-1947) 45.00 KB (46,080 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\dmserver.dll
es 2001.12.4648.0 (dnsrv.021114-1947) 637.50 KB (652,800 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\es.dll
pchsvc 5.2.3718.0 (dnsrv.021114-1947) 94.50 KB (96,768 bytes) 4/5/2003 8:05 PM Microsoft Corporation c:\windows\pchealth\helpctr\binaries\pchsvc.dll
srvsvc 5.2.3718.0 (dnsrv.021114-1947) 187.50 KB (192,000 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\srvsvc.dll
comsvcs 2001.12.4648.0 (dnsrv.021114-1947) 2.96 MB (3,104,256 bytes) 4/5/2003 8:02 PM Microsoft Corporation c:\windows\system32\comsvcs.dll
seclogon 5.2.3718.0 (dnsrv.021114-1947) 41.50 KB (42,496 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\seclogon.dll
sens 5.2.3718.0 (dnsrv.021114-1947) 90.50 KB (92,672 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\sens.dll
trkwks 5.2.3718.0 (dnsrv.021114-1947) 245.50 KB (251,392 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\trkwks.dll
wmisvc 5.2.3718.0 (dnsrv.021114-1947) 408.00 KB (417,792

bytes) 4/5/2003 8:02 PM Microsoft Corporation c:\windows\system32\wbem\wmisvc.dll
wuauserv 5.4.3718.0 (dnsrv.021114-1947) 17.00 KB (17,408 bytes) 4/5/2003 8:03 PM Microsoft Corporation c:\windows\system32\wuauserv.dll
wuaueng 5.4.3718.0 (dnsrv.021114-1947) 495.50 KB (507,392 bytes) 4/5/2003 8:03 PM Microsoft Corporation c:\windows\system32\wuaueng.dll
advpack 6.00.3718.0 (dnsrv.021114-1947) 240.00 KB (245,760 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\advpack.dll
wininet 6.00.3718.0 (dnsrv.021114-1947) 1.42 MB (1,492,480 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\wininet.dll
browser 5.2.3718.0 (dnsrv.021114-1947) 187.00 KB (191,488 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\browser.dll
netrap 5.2.3718.0 (dnsrv.021114-1947) 30.00 KB (30,720 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\netrap.dll
netman 5.2.3718.0 (dnsrv.021114-1947) 590.00 KB (604,160 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\netman.dll
wzcsapi 5.2.3718.0 (dnsrv.021114-1947) 49.00 KB (50,176 bytes) 11/15/2002 11:37 PM Microsoft Corporation c:\windows\system32\wzcsapi.dll
netshell 5.2.3718.0 (dnsrv.021114-1947) 2.65 MB (2,779,136 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\netshell.dll
clusapi 5.2.3718.0 (dnsrv.021114-1947) 165.00 KB (168,960 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\clusapi.dll
hnetcfg 5.2.3718.0 (dnsrv.021114-1947) 764.00 KB (782,336 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\hnetcfg.dll
wbemcomn 5.2.3718.0 (dnsrv.021114-1947) 596.00 KB (610,304 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\wbem\wbemcomn.dll
wbemcore 5.2.3718.0 (dnsrv.021114-1947) 1.63 MB (1,708,544 bytes) 4/5/2003 8:02 PM Microsoft Corporation c:\windows\system32\wbem\wbemcore.dll
esscli 5.2.3718.0 (dnsrv.021114-1947) 911.00 KB (932,864 bytes) 4/5/2003 8:02 PM Microsoft Corporation c:\windows\system32\wbem\esscli.dll
fastprox 5.2.3718.0 (dnsrv.021114-1947) 1.50 MB (1,573,888 bytes) 4/5/2003 8:02 PM Microsoft Corporation c:\windows\system32\wbem\fastprox.dll
wbemsvc 5.2.3718.0 (dnsrv.021114-1947) 62.50 KB (64,000 bytes) 4/5/2003 8:02 PM Microsoft Corporation c:\windows\system32\wbem\wbemsvc.dll
wmiutils 5.2.3718.0 (dnsrv.021114-1947) 284.50 KB (291,328 bytes) 4/5/2003 8:02 PM Microsoft Corporation c:\windows\system32\wbem\wmiutils.dll
repdrvfs 5.2.3718.0 (dnsrv.021114-1947) 598.50 KB (612,864 bytes) 4/5/2003 8:02 PM Microsoft Corporation c:\windows\system32\wbem\repdrvfs.dll
wmiprvsd 5.2.3718.0 (dnsrv.021114-1947) 1.38 MB (1,442,816 bytes) 4/5/2003 8:02 PM Microsoft Corporation c:\windows\system32\wbem\wmiprvsd.dll
wbemess 5.2.3718.0 (dnsrv.021114-1947) 972.00 KB (995,328 bytes) 4/5/2003 8:02 PM Microsoft Corporation c:\windows\system32\wbem\wbemess.dll
ncprov 5.2.3718.0 (dnsrv.021114-1947) 133.50 KB (136,704 bytes) 4/5/2003 8:02 PM Microsoft Corporation c:\windows\system32\wbem\ncprov.dll
wbemcons 5.2.3718.0 (dnsrv.021114-1947) 237.00 KB (242,688 bytes) 4/5/2003 8:02 PM Microsoft Corporation c:\windows\system32\wbem\wbemcons.dll
rasdlg 5.2.3718.0 (dnsrv.021114-1947) 1.35 MB (1,417,216 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\rasdlg.dll
rasadhlp 5.2.3718.0 (dnsrv.021114-1947) 13.00 KB (13,312 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\rasadhlp.dll
winhttp 5.2.3718.0 (dnsrv.021114-1947) 876.50 KB (897,536 bytes) 4/6/2003 4:42 AM Microsoft Corporation c:\windows\winsxs\ia64_microsoft.windows.winhttp_6595b64144ccf1df_5.1.0.0_x-ww_0fbfffd6\winhttp.dll
spoolsv 5.2.3718.0 (dnsrv.021114-1947) 154.00 KB (157,696 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\spoolsv.exe
spoolss 5.2.3718.0 (dnsrv.021114-1947) 227.50 KB (232,960 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\spoolss.dll
localspl 5.2.3718.0 (dnsrv.021114-1947) 835.50 KB (855,552 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\localspl.dll
cnbjmon 5.2.3680.0 (Lab03_dev(skatari).020509-1043) 99.50 KB (101,888 bytes) 11/15/2002 11:36 PM Microsoft Corporation c:\windows\system32\cnbjmon.dll
pjlmon 5.2.3718.0 (dnsrv.021114-1947) 37.50 KB (38,400 bytes) 11/15/2002 11:36 PM Microsoft Corporation c:\windows\system32\pjlmon.dll
tcpmon 5.2.3718.0 (dnsrv.021114-1947) 132.50 KB (135,680 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\tcpmon.dll
mgmtapi 5.2.3718.0 (dnsrv.021114-1947) 40.50 KB (41,472 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\mgmtapi.dll
snmpapi 5.2.3718.0 (dnsrv.021114-1947) 52.00 KB (53,248 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\snmpapi.dll
wsnmp32 5.2.3718.0 (dnsrv.021114-1947) 111.50 KB (114,176 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\wsnmp32.dll
usbmon 5.2.3718.0 (dnsrv.021114-1947) 43.50 KB (44,544 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\usbmon.dll
winrnr 5.2.3718.0 (dnsrv.021114-1947) 38.00 KB (38,912 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\winrnr.dll
wshqos 5.2.3718.0 (dnsrv.021114-1947) 65.50 KB (67,072 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\wshqos.dll
win32spl 5.2.3718.0 (dnsrv.021114-1947) 268.50 KB (274,944 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\win32spl.dll
inetpp 5.2.3718.0 (dnsrv.021114-1947) 212.00 KB (217,088 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\inetpp.dll
icmp 5.2.3718.0 (dnsrv.021114-1947) 2.50 KB (2,560 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\icmp.dll
ersvc 5.2.3718.0 (dnsrv.021114-1947) 61.00 KB (62,464 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\ersvc.dll
mssearch 9.107.8320.0 460.00 KB (471,040 bytes) 4/9/2003 2:54 AM Microsoft Corporation c:\program files\common files\system\mssearch\bin\mssearch.exe
mssws 9.107.8320.0 27.00 KB (27,648 bytes) 4/9/2003 2:54 AM Microsoft Corporation c:\program files\common files\system\mssearch\bin\mssws.dll
mssrch 9.107.8320.0 6.50 MB (6,819,328 bytes) 4/9/2003 2:54 AM Microsoft Corporation c:\progra~1\common~1\system\mssearch\bin\mssrch.dll
security 5.2.3718.0 (dnsrv.021114-1947) 6.50 KB (6,656 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\security.dll
tquery 9.107.8320.0 5.85 MB (6,133,248 bytes) 4/9/2003 2:54 AM Microsoft Corporation c:\program files\common files\system\mssearch\bin\tquery.dll
propdefs 9.107.8320.0 888.50 KB (909,824 bytes) 4/9/2003 2:54 AM Microsoft Corporation c:\progra~1\common~1\system\mssearch\bin\propdefs.dll
srchidx 9.107.8320.0 2.26 MB (2,374,144 bytes) 4/9/2003 2:54 AM Microsoft Corporation c:\progra~1\common~1\system\mssearch\bin\srchidx.dll
iprop 5.2.3718.0 (dnsrv.021114-1947) 3.00 KB (3,072 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\iprop.dll
dfssvc 5.2.3718.0 (dnsrv.021114-1947) 430.00 KB (440,320 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\dfssvc.exe
resutils 5.2.3718.0 (dnsrv.021114-1947) 147.50 KB (151,040 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\resutils.dll
mfc42u 6.00.2282.0 3.34 MB (3,506,176 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\mfc42u.dll
wsock32 5.2.3718.0 (dnsrv.021114-1947) 23.00 KB (23,552 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\wsock32.dll
termsrv 5.2.3718.0 (dnsrv.021114-1947) 603.00 KB (617,472 bytes) 4/5/2003 8:02 PM Microsoft Corporation c:\windows\system32\termsrv.dll
icaapi 5.2.3718.0 (dnsrv.021114-1947) 26.50 KB (27,136 bytes) 4/5/2003 8:02 PM Microsoft Corporation c:\windows\system32\icaapi.dll
mstlsapi 5.2.3718.0 (dnsrv.021114-1947) 311.00 KB (318,464 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\mstlsapi.dll
explorer 6.00.3718.0 (dnsrv.021114-1947) 1.62 MB (1,701,376 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\explorer.exe
browseui 6.00.3718.0 (dnsrv.021114-1947) 2.42 MB (2,536,960 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\browseui.dll
shdocvw 6.00.3718.0 (dnsrv.021114-1947) 3.19 MB (3,348,992 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\shdocvw.dll
apphelp 5.2.3718.0 (dnsrv.021114-1947) 262.50 KB (268,800 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\apphelp.dll
themeui 6.00.3718.0 (dnsrv.021114-1947) 823.00 KB (842,752 bytes) 11/18/2002 9:00 PM Microsoft Corporation c:\windows\system32\themeui.dll

msimg32 5.2.3718.0 (dnsrv.021114-1947) 7.00 KB (7,168 bytes)
11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\msimg32.dll
msutb 5.2.3718.0 (dnsrv.021114-1947) 562.00 KB (575,488
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\msutb.dll
msctf 5.2.3718.0 (dnsrv.021114-1947) 924.50 KB (946,688
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\msctf.dll
linkinfo 5.2.3718.0 (dnsrv.021114-1947) 42.00 KB (43,008
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\linkinfo.dll
ntshrui 6.00.3718.0 (dnsrv.021114-1947) 233.00 KB (238,592
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\ntshrui.dll
urlmon 6.00.3718.0 (dnsrv.021114-1947) 1.18 MB (1,232,384
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\urlmon.dll
webcheck 6.00.3718.0 (dnsrv.021114-1947) 665.00 KB (680,960
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\webcheck.dll
stobject 5.2.3718.0 (dnsrv.021114-1947) 178.00 KB (182,272
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\stobject.dll
batmeter 6.00.3718.0 (dnsrv.021114-1947) 55.50 KB (56,832
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\batmeter.dll
powrprof 6.00.3718.0 (dnsrv.021114-1947) 36.00 KB (36,864
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\powrprof.dll
printui 5.2.3718.0 (dnsrv.021114-1947) 1.09 MB (1,143,296
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\printui.dll
cfgmgr32 5.2.3718.0 (dnsrv.021114-1947) 15.50 KB (15,872
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\cfgmgr32.dll
browselc 6.00.3718.0 (dnsrv.021114-1947) 61.00 KB (62,464
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\browselc.dll
shdoclc 6.00.3718.0 (dnsrv.021114-1947) 520.50 KB (532,992
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\shdoclc.dll
mstask 5.2.3718.0 (dnsrv.021114-1947) 603.50 KB (617,984
bytes) 4/5/2003 8:04 PM Microsoft Corporation
c:\windows\system32\mstask.dll
msxml3 8.40.9214.0 3.41 MB (3,580,416 bytes)
11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\msxml3.dll
ctfmon 5.2.3718.0 (dnsrv.021114-1947) 31.50 KB (32,256
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\ctfmon.exe
sqlmangr 2000.080.0760.00 225.00 KB (230,400 bytes)
2/7/2003 6:48 AM Microsoft Corporation c:\program
files\microsoft sql server\80\tools\binn\sqlmangr.exe
w95scm 2000.080.0760.00 133.50 KB (136,704 bytes)
2/7/2003 6:49 AM Microsoft Corporation c:\program
files\microsoft sql server\80\tools\binn\w95scm.dll
odbc32 3.525.1015.0 620.00 KB (634,880 bytes)
11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\odbc32.dll
sqlsvc 2000.080.0760.00 315.50 KB (323,072 bytes)
2/7/2003 6:49 AM Microsoft Corporation c:\program
files\microsoft sql server\80\tools\binn\sqlsvc.dll
odbcbcp 2000.085.1015.00 48.00 KB (49,152 bytes)
11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\odbcbcp.dll
sqlresld 2000.080.0760.00 28.50 KB (29,184 bytes)
2/7/2003 6:49 AM Microsoft Corporation c:\program
files\microsoft sql server\80\tools\binn\sqlresld.dll
odbcint 3.525.1015.0 88.00 KB (90,112 bytes)
11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\odbcint.dll
sqlsvc 2000.080.0760.00 2.00 KB (2,048 bytes) 2/6/2003
10:09 PM Microsoft Corporation c:\program files\microsoft sql
server\80\tools\binn\resources\1033\sqlsvc.rll
sqlmangr 2000.080.0760.00 75.50 KB (77,312 bytes)
2/6/2003 10:15 PM Microsoft Corporation c:\program
files\microsoft sql server\80\tools\binn\resources\1033\sqlmangr.rll
ctfmon 5.2.3718.0 (dnsrv.021114-1947) 13.50 KB (13,824
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\syswow64\ctfmon.exe
wow64 5.2.3718.0 (dnsrv.021114-1947) 438.50 KB (449,024
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\wow64.dll
wow64win 5.2.3718.0 (dnsrv.021114-1947) 514.50 KB (526,848
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\wow64win.dll
wow64cpu 5.2.3718.0 (dnsrv.021114-1947) 38.00 KB (38,912
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\wow64cpu.dll
helpctr 5.2.3718.0 (dnsrv.021114-1947) 1.97 MB (2,067,456
bytes) 4/5/2003 8:05 PM Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\helpctr.exe
hcappres 5.2.3718.0 (dnsrv.021114-1947) 6.00 KB (6,144 bytes)
4/5/2003 8:05 PM Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\hcappres.dll
itss 5.2.3718.0 (dnsrv.021114-1947) 349.00 KB (357,376
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\itss.dll
pchshell 5.2.3718.0 (dnsrv.021114-1947) 277.00 KB (283,648
bytes) 4/5/2003 8:05 PM Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\pchshell.dll
mlang 6.00.3718.0 (dnsrv.021114-1947) 799.00 KB (818,176
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\mlang.dll
mshtml 6.00.3718.0 (dnsrv.021114-1947) 7.82 MB (8,204,800
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\mshtml.dll
msimtf 5.2.3718.0 (dnsrv.021114-1947) 528.00 KB (540,672
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\msimtf.dll
jscript 5.6.0.8028 1.21 MB (1,268,736 bytes) 11/18/2002
9:00 PM Microsoft Corporation c:\windows\system32\jscript.dll
msls31 3.10.349.0 448.00 KB (458,752 bytes) 11/18/2002
9:00 PM Microsoft Corporation c:\windows\system32\msls31.dll
imm32 5.2.3718.0 (dnsrv.021114-1947) 307.50 KB (314,880
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\imm32.dll
mshtmled 6.00.3718.0 (dnsrv.021114-1947) 1.34 MB (1,408,512
bytes) 11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\mshtmled.dll
vbscript 5.6.0.8028 1.06 MB (1,110,016 bytes) 11/18/2002
9:00 PM Microsoft Corporation c:\windows\system32\vbscript.dll
mfc42 6.00.2282.0 3.36 MB (3,526,656 bytes)
11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\mfc42.dll
msinfo 5.2.3718.0 (dnsrv.021114-1947) 1.20 MB (1,257,984
bytes) 4/5/2003 8:05 PM Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\msinfo.dll
riched32 5.2.3718.0 (dnsrv.021114-1947) 5.00 KB (5,120 bytes)
11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\riched32.dll
riched20 5.31.23.1218 1.25 MB (1,313,280 bytes)
11/18/2002 9:00 PM Microsoft Corporation
c:\windows\system32\riched20.dll
wbemprox 5.2.3718.0 (dnsrv.021114-1947) 46.00 KB (47,104
bytes) 4/5/2003 8:02 PM Microsoft Corporation
c:\windows\system32\wbem\wbemprox.dll
helpsvc 5.2.3718.0 (dnsrv.021114-1947) 2.18 MB (2,288,640
bytes) 4/5/2003 8:05 PM Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\helpsvc.exe

[Services]

Display Name Name State Start Mode Service
Type Path Error Control Start NameTag ID
Alerter Alerter Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k localservice
Normal NT AUTHORITY\LocalService 0
Application Layer Gateway Service ALG Stopped
Manual Own Process
c:\windows\system32\alg.exe Normal NT
AUTHORITY\LocalService 0
Application Management AppMgmt Stopped Manual
Share Process c:\windows\system32\svchost.exe
-k netsvcs Normal LocalSystem 0
Windows Audio AudioSrv Stopped Disabled Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Background Intelligent Transfer Service BITS Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Computer Browser Browser Running Auto Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Indexing Service CiSvc Stopped Manual Share
Process c:\windows\system32\cisvc.exe Normal
LocalSystem 0
ClipBook ClipSrv Stopped Disabled Own Process
c:\windows\system32\clipsrv.exe Normal
LocalSystem 0
COM+ System Application COMSysApp Stopped
Manual Own Process
c:\windows\system32\dllhost.exe /processid:{02d4b3f1-
fd88-11d1-960d-00805fc79235} Normal LocalSystem
0
Cryptographic Services CryptSvc Running Auto
Share Process c:\windows\system32\svchost.exe
-k netsvcs Normal LocalSystem 0
Distributed File System Dfs Running Auto
Own Process c:\windows\system32\dfssvc.exe
Normal LocalSystem 0
DHCP Client Dhcp Running Auto Share
Process c:\windows\system32\svchost.exe -k networkservice
Normal NT AUTHORITY\NetworkService 0

```
Logical Disk Manager Administrative Service  dmadmin  Stopped
        Manual  Share Process
        c:\windows\system32\dmadmin.exe /com  Normal
        LocalSystem  0
Logical Disk Manager  dmserver  Running  Auto  Share
Process  c:\windows\system32\svchost.exe -k netsvcs  Normal
        LocalSystem  0
DNS Client  Dnscache  Running  Auto  Share Process
        c:\windows\system32\svchost.exe -k networkservice
        Normal  NT AUTHORITY\NetworkService  0
Error Reporting Service  ERSvc  Running  Auto
        Share Process  c:\windows\system32\svchost.exe
-k winerr  Ignore  LocalSystem  0
Event Log  Eventlog  Running  Auto  Share Process
        c:\windows\system32\services.exe  Normal
        LocalSystem  0
COM+ Event System  EventSystem  Running  Manual
        Share Process  c:\windows\system32\svchost.exe
-k netsvcs  Normal  LocalSystem  0
Help and Support  helpsvc  Running  Auto  Share
Process  c:\windows\system32\svchost.exe -k netsvcs  Normal
        LocalSystem  0
Human Interface Device Access  HidServ  Stopped  Disabled
        Share Process  c:\windows\system32\svchost.exe
-k netsvcs  Normal  LocalSystem  0
HTTP SSL  HTTPFilter  Stopped  Manual  Share Process
        c:\windows\system32\lsass.exe  Normal
        LocalSystem  0
IAS Jet Database Access  IASJet  Stopped  Manual
        Share Process  c:\windows\syswow64\svchost.exe
-k iasjet  Normal  LocalSystem  0
IMAPI CD-Burning COM Service  ImapiService  Stopped
        Disabled  Own Process
        c:\windows\system32\imapi.exe  Normal
        LocalSystem  0
Intersite Messaging  IsmServ  Stopped  Disabled  Own
Process  c:\windows\system32\ismserv.exe  Normal
        LocalSystem  0
Kerberos Key Distribution Center  kdc  Stopped  Disabled
        Share Process  c:\windows\system32\lsass.exe
        Normal  LocalSystem  0
Server  lanmanserver  Running  Auto  Share
Process  c:\windows\system32\svchost.exe -k netsvcs  Normal
        LocalSystem  0
Workstation  lanmanworkstation  Running  Auto
        Share Process  c:\windows\system32\svchost.exe
-k netsvcs  Normal  LocalSystem  0
License Logging  LicenseService  Stopped  Disabled
        Own Process  c:\windows\system32\llssrv.exe
        Normal  NT AUTHORITY\NetworkService  0
TCP/IP NetBIOS Helper  LmHosts  Running  Auto
        Share Process  c:\windows\system32\svchost.exe
-k localservice  Normal  NT AUTHORITY\LocalService
        0
Messenger  Messenger  Stopped  Disabled  Share Process
        c:\windows\system32\svchost.exe -k netsvcs  Normal
        LocalSystem  0
Distributed Transaction Coordinator  MSDTC  Running
        Auto  Own Process
        c:\windows\system32\msdtc.exe  Normal  NT
AUTHORITY\NetworkService  0
Windows Installer  MSIServer  Stopped  Manual  Share
Process  c:\windows\system32\msiexec.exe /v  Normal
        LocalSystem  0
Microsoft Search  MSSEARCH  Running  Auto
        Share Process  "c:\program files\common
files\system\mssearch\bin\mssearch.exe"  Normal
        LocalSystem  0
MSSQLSERVER  MSSQLSERVER  Stopped  Manual
        Own Process  c:\program files\microsoft sql
server\mssql\binn\sqlservr.exe -smssqlserver  Normal
        LocalSystem  0
MSSQLServerADHelper  MSSQLServerADHelper
        Stopped  Manual  Own Process  c:\program
files\microsoft sql server\80\tools\binn\sqladhlp.exe  Normal
        LocalSystem  0
MSSQLServerOLAPService  MSSQLServerOLAPService
        Stopped  Manual  Own Process  c:\program
files\microsoft analysis services\bin\msmdsrv.exe  Normal
        LocalSystem  0
Network DDE  NetDDE  Stopped  Disabled  Share
Process  c:\windows\system32\netdde.exe  Normal
        LocalSystem  0
Network DDE DSDM  NetDDEdsdm  Stopped  Disabled
        Share Process  c:\windows\system32\netdde.exe
        Normal  LocalSystem  0
Net Logon  Netlogon  Stopped  Manual  Share Process
        c:\windows\system32\lsass.exe  Normal
        LocalSystem  0
Network Connections  Netman  Running  Manual  Share
Process  c:\windows\system32\svchost.exe -k netsvcs  Normal
        LocalSystem  0
Network Location Awareness (NLA)  Nla  Running
        Manual  Share Process
        c:\windows\system32\svchost.exe -k netsvcs  Normal
        LocalSystem  0
File Replication  NtFrs  Stopped  Manual  Own
Process  c:\windows\system32\ntfrs.exe  Ignore
        LocalSystem  0
NT LM Security Support Provider  NtLmSsp  Running  Manual
        Share Process  c:\windows\system32\lsass.exe
        Normal  LocalSystem  0
Removable Storage  NtmsSvc  Stopped  Manual  Share
Process  c:\windows\system32\svchost.exe -k netsvcs  Normal
        LocalSystem  0
Plug and Play  PlugPlay  Running  Auto  Share
Process  c:\windows\system32\services.exe  Normal
        LocalSystem  0
IPSEC Services  PolicyAgent  Running  Auto
        Share Process  c:\windows\system32\lsass.exe
        Normal  LocalSystem  0
Protected Storage  ProtectedStorage  Running  Auto
        Share Process  c:\windows\system32\lsass.exe
        Normal  LocalSystem  0
Remote Access Auto Connection Manager  RasAuto  Stopped
        Manual  Share Process
        c:\windows\system32\svchost.exe -k netsvcs  Normal
        LocalSystem  0
Remote Access Connection Manager  RasMan  Stopped
        Manual  Share Process
        c:\windows\system32\svchost.exe -k netsvcs  Normal
        LocalSystem  0
Remote Desktop Help Session Manager  RDSessMgr
        Stopped  Manual  Own Process
        c:\windows\system32\sessmgr.exe  Normal
        LocalSystem  0
Routing and Remote Access  RemoteAccess  Stopped
        Disabled  Share Process
        c:\windows\system32\svchost.exe -k netsvcs  Normal
        LocalSystem  0
Remote Registry  RemoteRegistry  Running  Auto
        Share Process  c:\windows\system32\svchost.exe
-k regsvc  Normal  NT AUTHORITY\LocalService  0
Remote Procedure Call (RPC) Locator  RpcLocator
        Stopped  Manual  Own Process
        c:\windows\system32\locator.exe  Normal  NT
AUTHORITY\NetworkService  0
Remote Procedure Call (RPC)  RpcSs  Running  Auto
        Share Process  c:\windows\system32\svchost -k
rpcss  Normal  LocalSystem  0
Resultant Set of Policy Provider  RSoPProv  Stopped  Manual
        Share Process
        c:\windows\system32\rsopprov.exe  Normal
        LocalSystem  0
Special Administration Console Helper  sacsvr  Stopped
        Manual  Share Process
        c:\windows\system32\svchost.exe -k netsvcs  Normal
        LocalSystem  0
Security Accounts Manager  SamSs  Running  Auto
        Share Process  c:\windows\system32\lsass.exe
        Normal  LocalSystem  0
Smart Card  SCardSvr  Stopped  Manual  Share Process
        c:\windows\system32\scardsvr.exe  Ignore
        NT AUTHORITY\LocalService  0
Task Scheduler  Schedule  Running  Auto  Share
Process  c:\windows\system32\svchost.exe -k netsvcs  Normal
        LocalSystem  0
Secondary Logon  seclogon  Running  Auto  Share
Process  c:\windows\system32\svchost.exe -k netsvcs  Ignore
        LocalSystem  0
System Event Notification  SENS  Running  Auto
        Share Process  c:\windows\system32\svchost.exe
-k netsvcs  Normal  LocalSystem  0
Shell Hardware Detection  ShellHWDetection  Running
        Auto  Share Process
        c:\windows\system32\svchost.exe -k netsvcs  Ignore
        LocalSystem  0
Print Spooler  Spooler  Running  Auto  Own
Process  c:\windows\system32\spoolsv.exe  Normal
        LocalSystem  0
SQLSERVERAGENT  SQLSERVERAGENT  Stopped  Manual
        Own Process  c:\program files\microsoft sql
server\mssql\binn\sqlagent.exe -i mssqlserver  Normal
        LocalSystem  0
Windows Image Acquisition (WIA)  stisvc  Stopped  Disabled
        Share Process  c:\windows\system32\svchost.exe
-k imgsvc  Normal  NT AUTHORITY\LocalService  0
Microsoft Software Shadow Copy Provider  swprv  Stopped
        Manual  Own Process
        c:\windows\system32\svchost.exe -k swprv  Normal
        LocalSystem  0
Performance Logs and Alerts  SysmonLog  Stopped
        Manual  Own Process
        c:\windows\system32\smlogsvc.exe  Normal
        NT Authority\NetworkService  0
Telephony  TapiSrv  Stopped  Manual  Share Process
```

```
c:\windows\system32\svchost.exe -k tapisrv Normal
LocalSystem 0
Terminal Services TermService Running Manual
Share Process c:\windows\system32\svchost.exe
-k termsvcs Normal LocalSystem 0
Telnet TlntSvr Stopped Disabled Own Process
c:\windows\system32\tlntsvr.exe Normal NT
AUTHORITY\LocalService 0
Distributed Link Tracking Server TrkSvr Stopped Disabled
Share Process c:\windows\system32\svchost.exe
-k netsvcs Normal LocalSystem 0
Distributed Link Tracking Client TrkWks Running Auto
Share Process c:\windows\system32\svchost.exe
-k netsvcs Normal LocalSystem 0
Terminal Services Session Directory Tssdis Stopped
Disabled Own Process
c:\windows\system32\tssdis.exe Normal
LocalSystem 0
Upload Manager uploadmgr Stopped Disabled Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Uninterruptible Power Supply UPS Stopped Manual
Own Process c:\windows\system32\ups.exe
Normal LocalSystem 0
Virtual Disk Service vds Stopped Manual Own
Process c:\windows\system32\vds.exe Normal
LocalSystem 0
Volume Shadow Copy VSS Stopped Manual Own
Process c:\windows\system32\vssvc.exe Normal
LocalSystem 0
Windows Time W32Time Running Auto Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
WebClient WebClient Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k localservice
Normal NT AUTHORITY\LocalService 0
WinHTTP Web Proxy Auto-Discovery Service WinHttpAutoProxySvc
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k localservice
Normal NT AUTHORITY\LocalService 0
Windows Management Instrumentation winmgmt Running
Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs Ignore
LocalSystem 0
Windows Management Instrumentation Driver Extensions Wmi
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
WMI Performance Adapter WmiApSrv Stopped Manual
Own Process
c:\windows\system32\wbem\wmiapsrv.exe Normal
LocalSystem 0
Automatic Updates wuauserv Running Auto Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Wireless Configuration WZCSVC Running Auto Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
```

[Program Groups]

```
Group Name Name User Name
Accessories Default User:Accessories Default
User
Accessories\Accessibility Default
User:Accessories\Accessibility Default User
Accessories\Entertainment Default
User:Accessories\Entertainment Default User
Startup Default User:Startup Default User
Accessories All Users:Accessories All Users
Accessories\Accessibility All
Users:Accessories\Accessibility All Users
Accessories\Communications All
Users:Accessories\Communications All Users
Accessories\Entertainment All
Users:Accessories\Entertainment All Users
Accessories\System Tools All Users:Accessories\System
Tools All Users
Administrative Tools All Users:Administrative Tools All Users
Microsoft SQL Server All Users:Microsoft SQL Server All Users
Startup All Users:Startup All Users
Accessories NT AUTHORITY\SYSTEM:Accessories
NT AUTHORITY\SYSTEM
Accessories\Accessibility NT
AUTHORITY\SYSTEM:Accessories\Accessibility NT
AUTHORITY\SYSTEM
Accessories\Entertainment NT
AUTHORITY\SYSTEM:Accessories\Entertainment NT
AUTHORITY\SYSTEM
Startup NT AUTHORITY\SYSTEM:Startup NT
AUTHORITY\SYSTEM
Accessories ASAMA\Administrator:Accessories
ASAMA\Administrator
Accessories\Accessibility
ASAMA\Administrator:Accessories\Accessibility
ASAMA\Administrator
Accessories\Entertainment
ASAMA\Administrator:Accessories\Entertainment
ASAMA\Administrator
Administrative Tools ASAMA\Administrator:Administrative Tools
ASAMA\Administrator
QLogic Corporation ASAMA\Administrator:QLogic Corporation
ASAMA\Administrator
QLogic Corporation\SANblade Control VIX
ASAMA\Administrator:QLogic Corporation\SANblade
Control VIX ASAMA\Administrator
Startup ASAMA\Administrator:Startup ASAMA\Administrator
```

[Startup Programs]

```
Program Command User Name Location
desktop desktop.ini NT AUTHORITY\SYSTEM Startup
desktop desktop.ini ASAMA\Administrator Startup
ctfmon.exe c:\windows\system32\ctfmon.exe ASAMA\Administrator
HKU\S-1-5-21-3116100345-911277676-3070513564-
500\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
desktop desktop.ini .DEFAULT Startup
desktop desktop.ini All Users Common Startup
Service Manager
c:\progra~1\micros~1\80\tools\binn\sqlmangr.exe /n
All Users Common Startup
```

[OLE Registration]

```
Object Local Server
Sound (OLE2) sndrec32.exe
Media Clip mplay32.exe
Video Clip mplay32.exe /avi
MIDI Sequence mplay32.exe /mid
Sound Not Available
Media Clip Not Available
WordPad Document "%programfiles%\windows
nt\accessories\wordpad.exe"
Bitmap Image mspaint.exe
```

[Windows Error Reporting]

```
Time Type Details
```

[Internet Settings]

[Internet Explorer]

[Following are sub-categories of this main category]
[Summary]

```
Item Value
Version 6.0.3718.0
Build 63718
Application Path C:\Program Files\Internet Explorer
Language English (United States)
Active Printer Not Available

Cipher Strength 128-bit
Content Advisor Disabled
IEAK Install No
```

[File Versions]

```
File Version Size Date Path Company
actxprxy.dll 6.0.3718.0 230 KB 11/18/2002 9:00:00 PM
C:\WINDOWS\system32 Microsoft Corporation
advpack.dll 6.0.3718.0 240 KB 11/18/2002 9:00:00 PM
C:\WINDOWS\system32 Microsoft Corporation
asctrls.ocx 6.0.3718.0 219 KB 11/18/2002 9:00:00 PM
C:\WINDOWS\system32 Microsoft Corporation
browselc.dll 6.0.3718.0 61 KB 11/18/2002 9:00:00
PM C:\WINDOWS\system32 Microsoft Corporation
browseui.dll 6.0.3718.0 2,478 KB 11/18/2002 9:00:00
PM C:\WINDOWS\system32 Microsoft Corporation
cdfview.dll 6.0.3718.0 292 KB 11/18/2002 9:00:00 PM
C:\WINDOWS\system32 Microsoft Corporation
comctl32.dll 5.82.3718.0 1,584 KB 11/18/2002
9:00:00 PM C:\WINDOWS\system32 Microsoft
Corporation
dxtrans.dll 6.3.3718.0 562 KB 11/18/2002 9:00:00 PM
C:\WINDOWS\system32 Microsoft Corporation
dxtmsft.dll 6.3.3718.0 919 KB 11/18/2002 9:00:00 PM
C:\WINDOWS\system32 Microsoft Corporation
iecont.dll <File Missing> Not Available Not
Available Not Available Not Available
iecontlc.dll <File Missing> Not Available Not
Available Not Available Not Available
iedkcs32.dll 16.0.3718.0 666 KB 11/18/2002
```

9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
iepeers.dll 6.0.3718.0 652 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
iesetup.dll 6.0.3718.0 83 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
ieuinit.inf Not Available 19 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Not Available
iexplore.exe 6.0.3718.0 102 KB 11/18/2002 9:00:00 PM C:\Program Files\Internet Explorer Microsoft Corporation
imgutil.dll 6.0.3718.0 88 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
inetcpl.cpl 6.0.3718.0 574 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
inetcplc.dll 6.0.3718.0 108 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
inseng.dll 6.0.3718.0 212 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
mlang.dll 6.0.3718.0 799 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
msencode.dll <File Missing> Not Available Not Available Not Available Not Available
mshta.exe 6.0.3718.0 59 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
mshtml.dll 6.0.3718.0 8,013 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
mshtml.tlb 6.0.3718.0 1,319 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
mshtmled.dll 6.0.3718.0 1,376 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
mshtmler.dll 6.0.3718.0 56 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
msident.dll 6.0.3718.0 128 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
msidntld.dll 6.0.3718.0 14 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
msieftp.dll 6.0.3718.0 536 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
msrating.dll 6.0.3718.0 379 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
mstime.dll 6.0.3718.0 1,621 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
occache.dll 6.0.3718.0 201 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
proctexe.ocx <File Missing> Not Available Not Available Not Available Not Available
sendmail.dll 6.0.3718.0 97 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
shdoclc.dll 6.0.3718.0 521 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
shdocvw.dll 6.0.3718.0 3,271 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
shfolder.dll 6.0.3718.0 37 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
shlwapi.dll 6.0.3718.0 720 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
tdc.ocx 1.3.0.3130 177 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
url.dll 6.0.3718.0 45 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
urlmon.dll 6.0.3718.0 1,204 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
webcheck.dll 6.0.3718.0 665 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation
wininet.dll 6.0.3718.0 1,458 KB 11/18/2002 9:00:00 PM C:\WINDOWS\system32 Microsoft Corporation

[Connectivity]

Item Value
Connection PreferenceNever dial

LAN Settings

AutoConfigProxy Not Available
AutoProxyDetectMode Disabled
AutoConfigURL
Proxy Disabled
ProxyServer
ProxyOverride

[Cache]

[Following are sub-categories of this main category]
[Summary]

Item Value
Page Refresh Type Automatic
Temporary Internet Files Folder C:\Documents and Settings\NetworkService\Local Settings\Temporary Internet Files
Total Disk Space Not Available
Available Disk Space Not Available
Maximum Cache Size Not Available
Available Cache Size Not Available

[List of Objects]

Program File Status CodeBase
No cached object information available

[Content]

[Following are sub-categories of this main category]
[Summary]

Item Value
Content Advisor Disabled

[Personal Certificates]

Issued To Issued By Validity Signature Algorithm
No personal certificate information available

[Other People Certificates]

Issued To Issued By Validity Signature Algorithm
No other people certificate information available

[Publishers]

Name
No publisher information available

[Security]

Zone Security Level
My Computer Custom
Local intranet Medium-low
Trusted sites Low
Internet Medium
Restricted sites High

<Client Configuration>

COM+ Application Configuration

-- COM+ Settings (properties of component TPCC.ALLTxns)
-- for each 31 frontends

Transactions: not supported

Enable object pooling
- Minimum pool size: 21
- Maximum pool size: 21
- Creation timeout (ms): 1200000

Enable object construction
- Constructor string: "dummy string (do not remove)"

Enable just in time activation
Component supports events and statistics

Concurrency: required

TPCC Application Registry

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPCC]
"Path"="C:\\Inetpub\\wwwroot\\"
"NumberOfDeliveryThreads"=dword:00000004
"MaxConnections"=dword:00004e20
"MaxPendingDeliveries"=dword:00000bb8
"DB_Protocol"="DBLIB"
"TxnMonitor"="COM"
"DbServer"="asama"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"=""
"COM_SinglePool"="YES"

InetInfo Registry

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:000000fa

"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,00,56,00,43,00
,00,00,53,00,\
  4d,00,54,00,50,00,53,00,56,00,43,00,00,00,00,00
"PoolThreadLimit"=dword:000007fc
"ThreadTimeout"=dword:00015180

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Inet
Info\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000842
"Last Help"=dword:00000843
"First Counter"=dword:00000802
"First Help"=dword:00000803
"Library Validation
Code"=hex:08,19,a2,cd,58,b6,c2,01,10,25,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,c3,bb,02,47,d4,c0,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000

WWW Service Registry

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3
SVC]
"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,00,54,0
0,5c,00,53,00,\

79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00,6e,00,65,00,
74,00,73,\

00,72,00,76,00,5c,00,69,00,6e,00,65,00,74,00,69,00,6e,00,66,00,6f,
00,2e,00,\
  65,00,78,00,65,00,00,00
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):49,00,49,00,53,00,41,00,44,00,4d,00,49,
00,4e,00,00,00,\
  00,00
"DependOnGroup"=hex(7):00,00
"ObjectName"="LocalSystem"
"Description"="Provides Web connectivity and administration through
the Internet Information Services snap-in."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3
SVC\ASP]
"NOTE"="This is for backward compatibility only."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3
SVC\ASP\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3
SVC\Parameters]
"MajorVersion"=dword:00000005
"MinorVersion"=dword:00000000
"InstallPath"="C:\\WINNT\\System32\\inetsrv"
"CertMapList"="C:\\WINNT\\System32\\inetsrv\\iiscrmap.dll"
"AccessDeniedMessage"="Error: Access is Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\\WINNT\\System32\\LogFiles"
"AcceptExOutstanding"=dword:000003e8

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3
SVC\Parameters\ADCLaunch]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3
SVC\Parameters\ADCLaunch\AdvancedDataFactory]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3
SVC\Parameters\ADCLaunch\RDSServer.DataFactory]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3
SVC\Parameters\Script Map]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3
SVC\Parameters\Virtual Roots]
"/"="c:\\inetpub\\wwwroot,,205"
"/Scripts"="c:\\inetpub\\scripts,,204"
"/IISHelp"="c:\\winnt\\help\\iishelp,,201"
"/IISAdmin"="C:\\WINNT\\System32\\inetsrv\\iisadmin,,201"
"/IISSamples"="c:\\inetpub\\iissamples,,201"
"/MSADC"="c:\\program files\\common files\\system\\msadc,,205"
"/_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\W3
SVC\Performance]
"Library"="w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"Last Counter"=dword:000008e6
"Last Help"=dword:000008e7
"First Counter"=dword:00000844
"First Help"=dword:00000845
"Library Validation
Code"=hex:58,49,e1,d1,58,b6,c2,01,10,3d,00,00,00,00,00,00
"WbemAdapFileTime"=hex:00,c3,bb,02,47,d4,c0,01
"WbemAdapFileSize"=dword:00001d10
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3
SVC\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,
00,00,00,02,\

00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,0
1,00,00,\

00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,
00,00,00,\

05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00
,00,05,\

20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01,02,00,01,01,
00,00,00,\

00,00,05,0b,00,00,00,00,20,02,00,00,00,00,00,1c,00,fd,01,02,00,01,02,00,
00,00,00,\

00,05,20,00,00,00,23,02,00,00,72,00,73,00,01,01,00,00,00,00,00,05,
12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3
SVC\Enum]
"0"="Root\\LEGACY_W3SVC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

System Information

System Information report written at: 04/20/2003 10:19:34 AM
[System Information]

[Following are sub-categories of this main category]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Server
Version	5.0.2195 Service Pack 2 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	ACL01
System Manufacturer	NEC
System Model	Express5800/120Re-2 [N8100-817]
System Type	X86-based PC
Processor	x86 Family 15 Model 2 Stepping 4 GenuineIntel ~1795 Mhz
Processor	x86 Family 15 Model 2 Stepping 4 GenuineIntel ~1795 Mhz
Processor	x86 Family 15 Model 2 Stepping 4 GenuineIntel ~1795 Mhz
Processor	x86 Family 15 Model 2 Stepping 4 GenuineIntel ~1795 Mhz
BIOS Version	FRU Ver 0.01
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition1
Locale	United States
User Name	ACL01\Administrator
Total Physical Memory	1,048,044 KB
Available Physical Memory	826,792 KB
Total Virtual Memory	2,389,968 KB
Available Virtual Memory	2,021,760 KB
Page File Space	1,341,924 KB
Page File	C:\pagefile.sys

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

Resource	Device

No conflicted/shared resources

[DMA]

Channel	Device	Status
4	Direct memory access controller	OK
2	Standard floppy disk controller	OK

[Forced Hardware]

Device	PNP Device ID
No Forced Hardware	

[I/O]

Address Range	Device	Status
0x0000-0x0CF7	PCI bus	OK
0x0000-0x0CF7	Direct memory access controller	OK
0x0D00-0xFFFF	PCI bus	OK
0x2000-0x3FFF	PCI standard PCI-to-PCI bridge	OK
0x2000-0x3FFF	PCI standard PCI-to-PCI bridge	OK
0x2000-0x3FFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0x3000-0x3FFF	PCI standard PCI-to-PCI bridge	OK
0x3000-0x3FFF	Adaptec AIC-7899 Ultra160/m PCI SCSI Card	OK
0x3400-0x34FF	Adaptec AIC-7899 Ultra160/m PCI SCSI Card	OK
0x2480-0x24BF	Intel(R) 82546EB Based Dual Port Network Connection	OK
0x2440-0x247F	Intel(R) 82546EB Based Dual Port Network Connection #2	OK
0x2400-0x243F	Intel(R) PRO/100 S Server Adapter	OK
0x4020-0x403F	Standard Universal PCI to USB Host Controller	OK
0x4000-0x401F	Standard Universal PCI to USB Host Controller	OK
0x1000-0x10FF	ATI Technologies Inc. RAGE XL PCI	OK
0x03B0-0x03BB	ATI Technologies Inc. RAGE XL PCI	OK
0x03C0-0x03DF	ATI Technologies Inc. RAGE XL PCI	OK
0x0A79-0x0A79	ISAPNP Read Data Port	OK
0x0279-0x0279	ISAPNP Read Data Port	OK
0x0274-0x0277	ISAPNP Read Data Port	OK
0x04D0-0x04D1	Motherboard resources	OK
0x0010-0x001F	Motherboard resources	OK
0x040B-0x040B	Motherboard resources	OK
0x04D6-0x04D6	Motherboard resources	OK
0x0C14-0x0C14	Motherboard resources	OK
0x0C49-0x0C49	Motherboard resources	OK
0x0C52-0x0C52	Motherboard resources	OK
0x0C6C-0x0C6C	Motherboard resources	OK
0x0C6F-0x0C6F	Motherboard resources	OK
0x0F50-0x0F57	Motherboard resources	OK
0x0C00-0x0C01	Motherboard resources	OK
0x0C98-0x0C98	Motherboard resources	OK
0x0CD6-0x0CD7	Motherboard resources	OK
0x002E-0x002F	Motherboard resources	OK
0x0530-0x0531	Motherboard resources	OK
0x0500-0x051F	Motherboard resources	OK
0x0540-0x055F	Motherboard resources	OK
0x0580-0x058F	Motherboard resources	OK
0x0580-0x058F	PCI Device	OK
0x0020-0x0021	Programmable interrupt controller	OK
0x00A0-0x00A1	Programmable interrupt controller	OK
0x0081-0x0083	Direct memory access controller	OK
0x0087-0x0087	Direct memory access controller	OK
0x0089-0x008B	Direct memory access controller	OK
0x008F-0x008F	Direct memory access controller	OK
0x00C0-0x00DF	Direct memory access controller	OK
0x0040-0x0043	System timer	OK
0x0070-0x0071	System CMOS/real time clock	OK
0x0061-0x0061	System speaker	OK
0x00F0-0x00FF	Numeric data processor	OK
0x0060-0x0060	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard	OK
0x0064-0x0064	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard	OK
0x0CA6-0x0CA6	Microsoft ACPI-Compliant Embedded Controller	OK
0x0CA7-0x0CA7	Microsoft ACPI-Compliant Embedded Controller	OK
0x03F2-0x03F3	Standard floppy disk controller	OK
0x03F4-0x03F5	Standard floppy disk controller	OK
0x03F7-0x03F7	Standard floppy disk controller	OK
0x03F8-0x03FF	Communications Port (COM1)	OK
0x02F8-0x02FF	Communications Port (COM2)	OK
0x03A0-0x03AF	Standard Dual Channel PCI IDE Controller	OK
0x01F0-0x01F7	Primary IDE Channel	OK
0x03F6-0x03F6	Primary IDE Channel	OK
0x0170-0x0177	Secondary IDE Channel	OK
0x0376-0x0376	Secondary IDE Channel	OK

[IRQs]

IRQ Number	Device
9	Microsoft ACPI-Compliant System
50	Adaptec AIC-7899 Ultra160/m PCI SCSI Card
49	Adaptec AIC-7899 Ultra160/m PCI SCSI Card
30	Intel(R) 82546EB Based Dual Port Network Connection
31	Intel(R) 82546EB Based Dual Port Network Connection #2
27	Intel(R) PRO/100 S Server Adapter
28	QLogic QLA23xx PCI Fibre Channel Adapter
16	Standard Universal PCI to USB Host Controller
19	Standard Universal PCI to USB Host Controller
17	ATI Technologies Inc. RAGE XL PCI
8	System CMOS/real time clock
13	Numeric data processor
12	Logitech PS/2 Port Mouse
1	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
6	Standard floppy disk controller
4	Communications Port (COM1)
3	Communications Port (COM2)
15	Secondary IDE Channel

[Memory]

Range	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	ATI Technologies Inc. RAGE XL PCI	OK
0x40000000-0xFFFFFFFF	PCI bus	OK
0xFE800000-0xFEBFFFFF	PCI standard PCI-to-PCI bridge	OK
0xFE800000-0xFEBFFFFF	PCI standard PCI-to-PCI bridge	OK
0xFEBE0000-0xFEBE0FFF	System Interrupt Controller	OK
0xFEA00000-0xFEAFFFFF	PCI standard PCI-to-PCI bridge	OK
0xFEAE0000-0xFEAE0FFF	Adaptec AIC-7899 Ultra160/m PCI SCSI Card	OK
0xFEAF0000-0xFEAF0FFF	Adaptec AIC-7899 Ultra160/m PCI SCSI Card	OK
0xFEBF0000-0xFEBF0FFF	System Interrupt Controller	OK
0xFE960000-0xFE97FFFF	Intel(R) 82546EB Based Dual Port Network Connection	OK
0xFE980000-0xFE99FFFF	Intel(R) 82546EB Based Dual Port Network Connection #2	OK
0xFE9E0000-0xFE9E0FFF	Intel(R) PRO/100 S Server Adapter	OK
0xFE9A0000-0xFE9BFFFF	Intel(R) PRO/100 S Server Adapter	OK
0xFE9F0000-0xFE9F0FFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0xFD000000-0xFDFFFFFF	ATI Technologies Inc. RAGE XL PCI	OK
0xFE7F0000-0xFE7F0FFF	ATI Technologies Inc. RAGE XL PCI	OK
0xFFC00000-0xFFFFFFFF	Intel(r) 82802 Firmware Hub Device	OK
0xFFBFFC00-0xFFBFFFFF	Standard Dual Channel PCI IDE Controller	OK

[Components]

[Following are sub-categories of this main category]

[Multimedia]

[Following are sub-categories of this main category]

[Audio Codecs]

Codec	Manufacturer	Description	Status
File	Version	Size	Creation Date
c:\winnt\system32\tssoft32.acm	DSP GROUP, INC.		

```
OK C:\WINNT\System32\TSSOFT32.ACM
1.01 9.27 KB (9,488 bytes) 12/8/1999 5:00:00 AM
c:\winnt\system32\msadp32.acm Microsoft
Corporation OK
C:\WINNT\System32\MSADP32.ACM
5.00.2134.1 14.77 KB (15,120 bytes)
12/8/1999 5:00:00 AM
c:\winnt\system32\imaadp32.acm Microsoft
Corporation OK
C:\WINNT\System32\IMAADP32.ACM
5.00.2134.1 16.27 KB (16,656 bytes)
12/8/1999 5:00:00 AM
c:\winnt\system32\msgsm32.acm Microsoft
Corporation OK
C:\WINNT\System32\MSGSM32.ACM
5.00.2134.1 22.27 KB (22,800 bytes)
12/8/1999 5:00:00 AM
c:\winnt\system32\lhacm.acm Microsoft Corporation
OK C:\WINNT\System32\LHACM.ACM
4.4.3385 33.27 KB (34,064 bytes) 1/7/2003
2:30:42 PM
c:\winnt\system32\msg723.acm Microsoft Corporation
OK C:\WINNT\System32\MSG723.ACM
4.4.3385 106.77 KB (109,328 bytes) 1/7/2003
2:30:41 PM
c:\winnt\system32\iac25_32.ax Intel Corporation Indeo®
audio software OK
C:\WINNT\System32\IAC25_32.AX 2.05.53
195.00 KB (199,680 bytes) 12/8/1999 5:00:00 AM
c:\winnt\system32\msg711.acm Microsoft Corporation
OK C:\WINNT\System32\MSG711.ACM
5.00.2134.1 10.27 KB (10,512 bytes)
12/8/1999 5:00:00 AM

[Video Codecs]

Codec Manufacturer Description Status
File Version Size Creation Date
c:\winnt\system32\ir50_32.dll Intel Corporation Indeo®
video 5.10 OK C:\WINNT\System32\IR50_32.DLL
R.5.10.15.2.55 737.50 KB (755,200 bytes)
12/8/1999 5:00:00 AM
c:\winnt\system32\msh261.drv Microsoft Corporation
OK C:\WINNT\System32\MSH261.DRV
4.4.3385 163.77 KB (167,696 bytes) 1/7/2003
2:30:42 PM
c:\winnt\system32\msh263.drv Microsoft Corporation
OK C:\WINNT\System32\MSH263.DRV
4.4.3385 252.27 KB (258,320 bytes) 1/7/2003
2:30:06 PM
c:\winnt\system32\iccvid.dll Radius Inc.
OK C:\WINNT\System32\ICCVID.DLL
1.10.0.6 108.00 KB (110,592 bytes) 12/8/1999
5:00:00 AM
c:\winnt\system32\ir32_32.dll Intel(R) Corporation
OK C:\WINNT\System32\IR32_32.DLL
Not Available 194.50 KB (199,168 bytes)
12/8/1999 5:00:00 AM
c:\winnt\system32\msrle32.dll Microsoft Corporation
OK C:\WINNT\System32\MSRLE32.DLL
5.00.2134.1 10.77 KB (11,024 bytes)
12/8/1999 5:00:00 AM
c:\winnt\system32\msvidc32.dll Microsoft Corporation
OK C:\WINNT\System32\MSVIDC32.DLL
5.00.2134.1 27.27 KB (27,920 bytes)
12/8/1999 5:00:00 AM

[CD-ROM]

Item Value
Drive D:
Description CD-ROM Drive
Media Loaded False
Media Type CD-ROM
Name MITSUMI CD-ROM SR243T
Manufacturer (Standard CD-ROM drives)
Status OK
Transfer Rate Not Available
SCSI Target ID 0
PNP Device ID IDE\CDROMMITSUMI_CD-
ROM_SR243T_____L02J____\5&2C4DDEAA&0&
0.0.0

[Sound Device]

Item Value
No sound devices

[Display]

Item Value
Name ATI Technologies Inc. RAGE XL PCI
PNP Device ID
PCI\VEN_1002&DEV_4752&SUBSYS_81B41033&REV_
27\4&27A7C225&0&60F0
Adapter Type ATI RAGE XL PCI, ATI Technologies Inc.
compatible
Adapter Description ATI Technologies Inc. RAGE XL PCI
Adapter RAM 8.00 MB (8,388,608 bytes)
Installed Drivers atidrab.dll
Driver Version 5.00.2179.1
INF File display.inf (atirage3 section)
Color Planes 1
Color Table Entries 65536
Resolution 1024 x 768 x 60 hertz
Bits/Pixel 16

[Infrared]

Item Value
No infrared devices

[Input]

[ Following are sub-categories of this main category ]

[Keyboard]

Item Value
Description Standard 101/102-Key or Microsoft Natural
PS/2 Keyboard
Name Enhanced (101- or 102-key)
Layout 00000409
PNP Device ID ACPI\PNP0303\4&1E30281&0
NumberOfFunctionKeys 12

[Pointing Device]

Item Value
Hardware Type Logitech PS/2 Port Mouse
Number of Buttons 3
Status OK
PNP Device ID ACPI\PNP0F12\4&1E30281&0
Power Management Supported False
Double Click Threshold 6
Handedness Right Handed Operation

[Modem]

Item Value
No modems

[Network]

[ Following are sub-categories of this main category ]

[Adapter]

Item Value
Name [00000000] RAS Async Adapter
Adapter Type Not Available
Product Name RAS Async Adapter
Installed True
PNP Device ID Not Available
Last Reset Not Available
Index 0
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Not Available

Name [00000001] WAN Miniport (L2TP)
Adapter Type Not Available
Product Name WAN Miniport (L2TP)
Installed True
PNP Device ID ROOT\MS_L2TPMINIPORT\0000
Last Reset Not Available
Index 1
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
```

DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Rasl2tp
Driver c:\winnt\system32\drivers\rasl2tp.sys (50800, 5.00.2179.1)

Name [00000002] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Name WAN Miniport (PPTP)
Installed True
PNP Device ID ROOT\MS_PPTPMINIPORT\0000
Last Reset Not Available
Index 2
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30
Service Name PptpMiniport
Driver c:\winnt\system32\drivers\raspptp.sys (47856, 5.00.2160.1)

Name [00000003] Direct Parallel
Adapter Type Not Available
Product Name Direct Parallel
Installed True
PNP Device ID ROOT\MS_PTIMINIPORT\0000
Last Reset Not Available
Index 3
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name Raspti
Driver c:\winnt\system32\drivers\raspti.sys (16880, 5.00.2146.1)

Name [00000004] WAN Miniport (IP)
Adapter Type Not Available
Product Name WAN Miniport (IP)
Installed True
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset Not Available
Index 4
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Service Name NdisWan
Driver c:\winnt\system32\drivers\ndiswan.sys (90096, 5.00.2195.2779)

Name [00000005] Intel(R) 82546EB Based Dual Port Network Connection
Adapter Type Ethernet 802.3
Product Name Intel(R) 82546EB Based Dual Port Network Connection
Installed True
PNP Device ID PCI\VEN_8086&DEV_1010&SUBSYS_81B41033&REV_01\5&890422F&0&38F818
Last Reset Not Available
Index 5
Service Name E1000
IP Address 10.10.201.250
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:07:E9:04:75:0C
Service Name E1000
IRQ Number 30
I/O Port 0x2480-0x24BF
Driver c:\winnt\system32\drivers\e1000nt5.sys (100544, 6.2.11.0)

Name [00000006] Intel(R) 82546EB Based Dual Port Network Connection
Adapter Type Ethernet 802.3
Product Name Intel(R) 82546EB Based Dual Port Network Connection
Installed True
PNP Device ID PCI\VEN_8086&DEV_1010&SUBSYS_81B41033&REV_01\5&890422F&0&39F818
Last Reset Not Available
Index 6
Service Name E1000
IP Address 10.10.1.250
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:07:E9:04:75:0D
Service Name E1000
IRQ Number 31
I/O Port 0x2440-0x247F
Driver c:\winnt\system32\drivers\e1000nt5.sys (100544, 6.2.11.0)

Name [00000007] Intel(R) PRO/100 S Server Adapter
Adapter Type Ethernet 802.3
Product Name Intel(R) PRO/100 S Server Adapter
Installed True
PNP Device ID PCI\VEN_8086&DEV_1229&SUBSYS_10408086&REV_0C\5&890422F&0&48F818
Last Reset Not Available
Index 7
Service Name E100B
IP Address 10.10.101.250
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:02:B3:BE:1A:FF
Service Name E100B
IRQ Number 27
I/O Port 0x2400-0x243F
Driver c:\winnt\system32\drivers\e100bnt5.sys (139536, 6.01.03.0000)

[Protocol]

Item Value
Name MSAFD Tcpip [TCP/IP]
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 16 bytes
MaximumMessageSize 0 bytes
MessageOriented False
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData True
SupportsGracefulClosing True
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD Tcpip [UDP/IP]
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 16 bytes
MaximumMessageSize 65467 bytes
MessageOriented True
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting True

Name RSVP UDP Service Provider
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 16 bytes
MaximumMessageSize 65467 bytes
MessageOriented True
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption True
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticastingTrue

Name RSVP TCP Service Provider
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 16 bytes
MaximumMessageSize 0 bytes
MessageOriented False
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption True
SupportsExpeditedData True
SupportsGracefulClosing True
SupportsGuaranteedBandwidth False
SupportsMulticastingFalse

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{6B533E53-BE70-47E1-8481-9653BDC3133D}] SEQPACKET 4
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticastingFalse

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{6B533E53-BE70-47E1-8481-9653BDC3133D}] DATAGRAM 4
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticastingFalse

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{265D4C13-27E9-4B28-8476-0218D64AD72B}] SEQPACKET 3
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticastingFalse

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{265D4C13-27E9-4B28-8476-0218D64AD72B}] DATAGRAM 3
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticastingFalse

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{7B596BF9-8493-448A-B6AD-DFD2F9817090}] SEQPACKET 0
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticastingFalse

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{7B596BF9-8493-448A-B6AD-DFD2F9817090}] DATAGRAM 0
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticastingFalse

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{13D219AB-B3F7-40AF-B851-F480C01EB495}] SEQPACKET 1
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticastingFalse

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{13D219AB-B3F7-40AF-B851-F480C01EB495}] DATAGRAM 1
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False

SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticastingFalse

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{DE342350-8743-4F17-A539-668DC75FD474}] SEQPACKET 2
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticastingFalse

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{DE342350-8743-4F17-A539-668DC75FD474}] DATAGRAM 2
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticastingFalse

[WinSock]

Item Value
File c:\winnt\system32\winsock.dll
Version 3.10
Size 2.80 KB (2,864 bytes)

File c:\winnt\system32\wsock32.dll
Version 5.00.2195.2871
Size 21.27 KB (21,776 bytes)

[Ports]

[Following are sub-categories of this main category]

[Serial]

Item Value
Name COM1
Status OK
PNP Device ID ACPI\PNP0501\1
Maximum Input Buffer Size 0
Maximum Output Buffer Size False
Settable Baud Rate True
Settable Data Bits True
Settable Flow Control True
Settable Parity True
Settable Parity Check True
Settable Stop Bits True
Settable RLSD True
Supports RLSD True
Supports 16 Bit Mode False
Supports Special Characters False
Baud Rate 9600
Bits/Byte 8
Stop Bits 1
Parity None
Busy 0
Abort Read/Write on Error 0
Binary Mode Enabled -1
Continue XMit on XOff 0
CTS Outflow Control 0
Discard NULL Bytes 0
DSR Outflow Control 0
DSR Sensitivity 0
DTR Flow Control Type Enable
EOF Character 0
Error Replace Character 0
Error Replacement Enabled 0
Event Character 0
Parity Check Enabled 0
RTS Flow Control Type Enable
XOff Character 19
XOffXMit Threshold 512
XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control 0
IRQ Number 4
I/O Port 0x03F8-0x03FF
Driver c:\winnt\system32\drivers\serial.sys (62416, 5.00.2195.2780)

Name COM2
Status OK
PNP Device ID ACPI\PNP0501\2
Maximum Input Buffer Size 0
Maximum Output Buffer Size False
Settable Baud Rate True
Settable Data Bits True
Settable Flow Control True
Settable Parity True
Settable Parity Check True
Settable Stop Bits True
Settable RLSD True
Supports RLSD True
Supports 16 Bit Mode False
Supports Special Characters False
Baud Rate 9600
Bits/Byte 8
Stop Bits 1
Parity None
Busy 0
Abort Read/Write on Error 0
Binary Mode Enabled -1
Continue XMit on XOff 0
CTS Outflow Control 0
Discard NULL Bytes 0
DSR Outflow Control 0
DSR Sensitivity 0
DTR Flow Control Type Enable
EOF Character 0
Error Replace Character 0
Error Replacement Enabled 0
Event Character 0
Parity Check Enabled 0
RTS Flow Control Type Enable
XOff Character 19
XOffXMit Threshold 512
XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control 0
IRQ Number 3
I/O Port 0x02F8-0x02FF
Driver c:\winnt\system32\drivers\serial.sys (62416, 5.00.2195.2780)

[Parallel]

Item Value
No parallel port information

[Storage]

[Following are sub-categories of this main category]

[Drives]

Item Value
Drive A:
Description 3 1/2 Inch Floppy Drive

Drive C:
Description Local Fixed Disk
Compressed False
File System NTFS
Size 16.93 GB (18,177,835,008 bytes)
Free Space 15.03 GB (16,135,589,888 bytes)
Volume Name
Volume Serial Number A87C0E3A
Partition Disk #0, Partition #0
Partition Size 16.93 GB (18,177,836,544 bytes)
Starting Offset 32256 bytes
Drive Description Disk drive
Drive Manufacturer (Standard disk drives)

```
Drive Model             QUANTUM ATLAS10K3_18_SCA SCSI Disk
Device
Drive BytesPerSector 512
Drive MediaLoaded    True
Drive MediaType      Fixed hard disk media
Drive Partitions     1
Drive SCSIBus        0
Drive SCSILogicalUnit          0
Drive SCSIPort       3
Drive SCSITargetId   0
Drive SectorsPerTrack          63
Drive Size 18186094080 bytes
Drive TotalCylinders 2211
Drive TotalSectors   35519715
Drive TotalTracks    563805
Drive TracksPerCylinder        255


[SCSI]

Item      Value
Name      Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Caption   Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Driver    adpu160m
Status    OK
PNP Device ID
          PCI\VEN_9005&DEV_00CF&SUBSYS_81B41033&REV
_01\5&17CCB9A0&0&38E818
Device ID
          PCI\VEN_9005&DEV_00CF&SUBSYS_81B41033&REV
_01\5&17CCB9A0&0&38E818
Device Map          Not Available
Index     Not Available
Max Number Controlled          Not Available
IRQ Number          50
I/O Port  0x3400-0x34FF
Driver    c:\winnt\system32\drivers\adpu160m.sys (64432,
v3.10a)

Name      Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Caption   Adaptec AIC-7899 Ultra160/m PCI SCSI Card
Driver    adpu160m
Status    OK
PNP Device ID
          PCI\VEN_9005&DEV_00CF&SUBSYS_81B41033&REV
_01\5&17CCB9A0&0&39E818
Device ID
          PCI\VEN_9005&DEV_00CF&SUBSYS_81B41033&REV
_01\5&17CCB9A0&0&39E818
Device Map          Not Available
Index     Not Available
Max Number Controlled          Not Available
IRQ Number          49
I/O Port  0x3000-0x3FFF
Driver    c:\winnt\system32\drivers\adpu160m.sys (64432,
v3.10a)

Name      QLogic QLA23xx PCI Fibre Channel Adapter
Caption   QLogic QLA23xx PCI Fibre Channel Adapter
Driver    ql2300
Status    OK
PNP Device ID
          PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_
02\5&890422F&0&50F818
Device ID
          PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_
02\5&890422F&0&50F818
Device Map          Not Available
Index     Not Available
Max Number Controlled          Not Available
IRQ Number          28
I/O Port  0x2000-0x3FFF
Driver    c:\winnt\system32\drivers\ql2300.sys (442328,
8.2.0.10 (W2K VI))


[Printing]

Name      Port Name Server Name
No printing information

[Problem Devices]

Device    PNP Device ID          Error Code
PCI Device
          PCI\VEN_8086&DEV_2541&SUBSYS_81B41033&REV_
03\3&267A616A&0&01          28
System Interrupt Controller
          PCI\VEN_8086&DEV_1461&SUBSYS_81B41033&REV_
03\4&82C0EAA&0&E018          28
System Interrupt Controller
          PCI\VEN_8086&DEV_1461&SUBSYS_81B41033&REV_
03\4&82C0EAA&0&F018          28
PCI Device
          PCI\VEN_8086&DEV_2546&SUBSYS_81B41033&REV_
03\3&267A616A&0&19          28
PCI Device
          PCI\VEN_8086&DEV_2483&SUBSYS_81B41033&REV_
02\3&267A616A&0&FB          28

[USB]

Device    PNP Device ID
Standard Universal PCI to USB Host Controller
          PCI\VEN_8086&DEV_2482&SUBSYS_81B41033&REV_
02\3&267A616A&0&E8
USB Root Hub        USB\ROOT_HUB\4&42B666F&0
Standard Universal PCI to USB Host Controller
          PCI\VEN_8086&DEV_2484&SUBSYS_81B41033&REV_
02\3&267A616A&0&E9
USB Root Hub        USB\ROOT_HUB\4&396CDB1C&0

[Software Environment]

[ Following are sub-categories of this main category ]

[Drivers]

Name      Description          File      Type      Started
          Start Mode           State     Status    Error
Control   Accept Pause         Accept Stop
abiosdsk  Abiosdsk  Not Available          Kernel Driver
          False     Disabled   Stopped   OK        Ignore
          False     False
abp480n5  abp480n5  Not Available          Kernel Driver
          False     Disabled   Stopped   OK        Normal
          False     False
acpi      Microsoft ACPI Driver
          c:\winnt\system32\drivers\acpi.sys          Kernel
Driver    True      Boot       Running   OK        Normal
          False     True
acpiec    Microsoft Embedded Controller Driver
          c:\winnt\system32\drivers\acpiec.sys        Kernel
Driver    True      Boot       Running   OK        Normal
          False     True
adpu160m  adpu160m  c:\winnt\system32\drivers\adpu160m.sys
          Kernel Driver          True      Boot      Running
          OK        Normal       False     True
afd       AFD Networking Support Environment
          c:\winnt\system32\drivers\afd.sys           Kernel
Driver    True      Auto       Running   OK        Normal
          False     True
aha154x   Aha154x   Not Available          Kernel Driver
          False     Disabled   Stopped   OK        Normal
          False     False
aic116x   aic116x   Not Available          Kernel Driver
          False     Disabled   Stopped   OK        Normal
          False     False
aic78u2   aic78u2   Not Available          Kernel Driver
          False     Disabled   Stopped   OK        Normal
          False     False
aic78xx   aic78xx   Not Available          Kernel Driver
          False     Disabled   Stopped   OK        Normal
          False     False
ami0nt    ami0nt    Not Available          Kernel Driver
          False     Disabled   Stopped   OK        Normal
          False     False
amsint    amsint    Not Available          Kernel Driver
          False     Disabled   Stopped   OK        Normal
          False     False
asc       asc       Not Available          Kernel Driver
          False     Disabled   Stopped   OK        Normal
          False     False
asc3350p  asc3350p  Not Available          Kernel Driver
          False     Disabled   Stopped   OK        Normal
          False     False
asc3550   asc3550   Not Available          Kernel Driver
          False     Disabled   Stopped   OK        Normal
          False     False
asyncmac  RAS Asynchronous Media Driver
          c:\winnt\system32\drivers\asyncmac.sys      Kernel
Driver    False     Manual     Stopped   OK        Normal
          False     False
atapi     Standard IDE/ESDI Hard Disk Controller
          c:\winnt\system32\drivers\atapi.sys         Kernel
Driver    True      Boot       Running   OK        Normal
          False     True
atdisk    Atdisk    Not Available          Kernel Driver
          False     Disabled   Stopped   OK        Ignore
          False     False
atirage3  atirage3  c:\winnt\system32\drivers\atimpab.sys
          Kernel Driver          True      Manual    Running
          OK        Ignore       False     True
```

```
atmarpc ATM ARP Client Protocol
        c:\winnt\system32\drivers\atmarpc.sys        Kernel
Driver  False   Manual  Stopped OK      Normal
        False   False
audstub Audio Stub Driver
        c:\winnt\system32\drivers\audstub.sys        Kernel
Driver  True    Manual  Running OK      Normal
        False   True
beep    Beep    c:\winnt\system32\drivers\beep.sys
        Kernel Driver           True    System  Running
        OK      Normal  False   True
buslogic BusLogic Not Available         Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
cd20xrnt cd20xrnt Not Available         Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
cdaudio Cdaudio c:\winnt\system32\drivers\cdaudio.sys
        Kernel Driver           False   System  Stopped
        OK      Ignore  False   False
cdfs    Cdfs    c:\winnt\system32\drivers\cdfs.sys
        File System Driver      True    Disabled Running
        OK      Normal  False   True
cdrom   CD-ROM Driver
        c:\winnt\system32\drivers\cdrom.sys          Kernel
Driver  True    System  Running OK      Normal
        False   True
changer Changer Not Available           Kernel Driver
        False   System  Stopped OK      Ignore
        False   False
cpqarray Cpqarray Not Available         Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
cpqarry2 cpqarry2 Not Available         Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
cpqfcalm cpqfcalm Not Available         Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
cpqfws2e cpqfws2e Not Available         Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
dac960nt dac960nt Not Available         Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
deckzpsx deckzpsx Not Available         Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
dfsdriver DfsDriver c:\winnt\system32\drivers\dfs.sys
        File System Driver      True    Boot    Running
        OK      Normal  False   True
disk    Disk Driver
        c:\winnt\system32\drivers\disk.sys           Kernel
Driver  True    Boot    Running OK      Normal
        False   True
diskperf Diskperf c:\winnt\system32\drivers\diskperf.sys
        Kernel Driver           True    Boot    Running
        OK      Normal  False   True
dmboot  dmboot  c:\winnt\system32\drivers\dmboot.sys
        Kernel Driver           False   Disabled Stopped
        OK      Normal  False   False
dmio    Logical Disk Manager Driver
        c:\winnt\system32\drivers\dmio.sys           Kernel
Driver  True    Boot    Running OK      Normal
        False   True
dmload  dmload  c:\winnt\system32\drivers\dmload.sys
        Kernel Driver           True    Boot    Running
        OK      Normal  False   True
e1000   Intel(R) PRO/1000 Adapter Driver
        c:\winnt\system32\drivers\e1000nt5.sys       Kernel
Driver  True    Manual  Running OK      Normal
        False   True
e100b   Intel(R) PRO Adapter Driver
        c:\winnt\system32\drivers\e100bnt5.sys       Kernel
Driver  True    Manual  Running OK      Normal
        False   True
efs     EFS     c:\winnt\system32\drivers\efs.sys
        File System Driver      True    Disabled Running
        OK      Normal  False   True
fastfat Fastfat c:\winnt\system32\drivers\fastfat.sys
        File System Driver      True    Disabled Running
        OK      Normal  False   True
fd16_700 Fd16_700 Not Available         Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
fdc     Floppy Disk Controller Driver
        c:\winnt\system32\drivers\fdc.sys            Kernel
Driver  True    Manual  Running OK      Normal
        False   True
fips    Fips    c:\winnt\system32\drivers\fips.sys
        Kernel Driver           True    Auto    Running
        OK      Normal  False   True
fireport fireport Not Available         Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
flashpnt flashpnt Not Available         Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
flpydisk Floppy Disk Driver
        c:\winnt\system32\drivers\flpydisk.sys       Kernel
Driver  True    Manual  Running OK      Normal
        False   True
ftdisk  Volume Manager Driver
        c:\winnt\system32\drivers\ftdisk.sys         Kernel
Driver  True    Boot    Running OK      Normal
        False   True
gpc     Generic Packet Classifier
        c:\winnt\system32\drivers\msgpc.sys          Kernel
Driver  True    Manual  Running OK      Normal
        False   True
i8042prt i8042 Keyboard and PS/2 Mouse Port Driver
        c:\winnt\system32\drivers\i8042prt.sys       Kernel
Driver  True    System  Running OK      Normal
        False   True
ini910u ini910u Not Available           Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
intelide Intellde Not Available         Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
ipfilterdriver  IP Traffic Filter Driver
        c:\winnt\system32\drivers\ipfltdrv.sys       Kernel
Driver  False   Manual  Stopped OK      Normal
        False   False
ipinip  IP in IP Tunnel Driver
        c:\winnt\system32\drivers\ipinip.sys         Kernel
Driver  False   Manual  Stopped OK      Normal
        False   False
ipnat   IP Network Address Translator
        c:\winnt\system32\drivers\ipnat.sys          Kernel
Driver  False   Manual  Stopped OK      Normal
        False   False
ipsec   IPSEC driver
        c:\winnt\system32\drivers\ipsec.sys          Kernel
Driver  True    Manual  Running OK      Normal
        False   True
ipsraidn ipsraidn Not Available         Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
isapnp  PnP ISA/EISA Bus Driver
        c:\winnt\system32\drivers\isapnp.sys         Kernel
Driver  True    Boot    Running OK      Critical
        False   True
kbdclass Keyboard Class Driver
        c:\winnt\system32\drivers\kbdclass.sys       Kernel
Driver  True    System  Running OK      Normal
        False   True
ksecdd  KSecDD  c:\winnt\system32\drivers\ksecdd.sys
        Kernel Driver           True    Boot    Running
        OK      Normal  False   True
lbrtfdc lbrtfdc Not Available           Kernel Driver
        False   System  Stopped OK      Ignore
        False   False
lp6nds35 lp6nds35 Not Available         Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
mnmdd   mnmdd   c:\winnt\system32\drivers\mnmdd.sys
        Kernel Driver           True    System  Running
        OK      Ignore  False   True
modem   Modem   c:\winnt\system32\drivers\modem.sys
        Kernel Driver           False   Manual  Stopped
        OK      Ignore  False   False
mouclass Mouse Class Driver
        c:\winnt\system32\drivers\mouclass.sys       Kernel
Driver  True    System  Running OK      Normal
        False   True
mountmgr MountMgr c:\winnt\system32\drivers\mountmgr.sys
        Kernel Driver           True    Boot    Running
        OK      Normal  False   True
mraid35x mraid35x Not Available         Kernel Driver
        False   Disabled Stopped OK      Normal
        False   False
mrxsmb  MRXSMB  c:\winnt\system32\drivers\mrxsmb.sys
        File System Driver      True    System  Running
        OK      Normal  False   True
msfs    Msfs    c:\winnt\system32\drivers\msfs.sys
        File System Driver      True    System  Running
        OK      Normal  False   True
mskssrv Microsoft Streaming Service Proxy
        c:\winnt\system32\drivers\mskssrv.sys        Kernel
Driver  False   Manual  Stopped OK      Normal
        False   False
mspclock Microsoft Streaming Clock Proxy
```

```
        c:\winnt\system32\drivers\mspclock.sys      Kernel
Driver  False   Manual  Stopped OK      Normal
        False   False
mspqm   Microsoft Streaming Quality Manager Proxy
        c:\winnt\system32\drivers\mspqm.sys        Kernel
Driver  False   Manual  Stopped OK      Normal
        False   False
mup     Mup     c:\winnt\system32\drivers\mup.sys
        File System Driver      True    Boot    Running
        OK      Normal  False   True
ncrc710 Ncrc710 Not Available           Kernel Driver
        False   Disabled        Stopped OK      Normal
        False   False
ndis    NDIS System Driver
        c:\winnt\system32\drivers\ndis.sys         Kernel
Driver  True    Boot    Running OK      Normal
        False   True
ndistapi        Remote Access NDIS TAPI Driver
        c:\winnt\system32\drivers\ndistapi.sys     Kernel
Driver  True    Manual  Running OK      Normal
        False   True
ndiswan Remote Access NDIS WAN Driver
        c:\winnt\system32\drivers\ndiswan.sys      Kernel
Driver  True    Manual  Running OK      Normal
        False   True
ndproxy NDIS Proxy
        c:\winnt\system32\drivers\ndproxy.sys      Kernel
Driver  True    Manual  Running OK      Normal
        False   True
netbios NetBIOS Interface
        c:\winnt\system32\drivers\netbios.sys      File
System Driver   True    System  Running OK
        Normal  False   True
netbt   NetBios over Tcpip
        c:\winnt\system32\drivers\netbt.sys        Kernel
Driver  True    System  Running OK      Normal
        False   True
netdetect       NetDetect       c:\winnt\system32\drivers\netdtect.sys
        Kernel Driver           False   Manual  Stopped
        OK      Normal  False   False
npfs    Npfs    c:\winnt\system32\drivers\npfs.sys
        File System Driver      True    System  Running
        OK      Normal  False   True
ntfs    Ntfs    c:\winnt\system32\drivers\ntfs.sys
        File System Driver      True    Disabled        Running
        OK      Normal  False   True
null    Null    c:\winnt\system32\drivers\null.sys
        Kernel Driver           True    System  Running
        OK      Normal  False   True
nwlnkflt        IPX Traffic Filter Driver
        c:\winnt\system32\drivers\nwlnkflt.sys     Kernel
Driver  False   Manual  Stopped OK      Normal
        False   False
nwlnkfwd        IPX Traffic Forwarder Driver
        c:\winnt\system32\drivers\nwlnkfwd.sys     Kernel
Driver  False   Manual  Stopped OK      Normal
        False   False
parallel        Parallel        c:\winnt\system32\drivers\parallel.sys
        Kernel Driver           False   Auto    Stopped
        OK      Ignore  False   False
parport Parport c:\winnt\system32\drivers\parport.sys
        Kernel Driver           False   Auto    Stopped
        OK      Ignore  False   False
partmgr PartMgr c:\winnt\system32\drivers\partmgr.sys
        Kernel Driver           True    Boot    Running
        OK      Normal  False   True
parvdm  ParVdm  c:\winnt\system32\drivers\parvdm.sys
        Kernel Driver           False   Auto    Stopped
        OK      Ignore  False   False
pci     PCI Bus Driver
        c:\winnt\system32\drivers\pci.sys          Kernel
Driver  True    Boot    Running OK      Critical
        False   True
pcidump PCIDump Not Available           Kernel Driver
        False   System  Stopped OK      Ignore
        False   False
pciide  PCIIde  c:\winnt\system32\drivers\pciide.sys
        Kernel Driver           True    Boot    Running
        OK      Normal  False   True
pcmcia  Pcmcia  c:\winnt\system32\drivers\pcmcia.sys
        Kernel Driver           False   Disabled        Stopped
        OK      Normal  False   False
pdcomp  PDCOMP  Not Available           Kernel Driver
        False   Manual  Stopped OK      Ignore
        False   False
pdframe PDFRAME Not Available           Kernel Driver
        False   Manual  Stopped OK      Ignore
        False   False
pdreli  PDRELI  Not Available           Kernel Driver
        False   Manual  Stopped OK      Ignore
        False   False
pdrframe        PDRFRAME        Not Available           Kernel
Driver  False   Manual  Stopped OK      Ignore
        False   False
pptpminiport    WAN Miniport (PPTP)
        c:\winnt\system32\drivers\rasppptp.sys     Kernel
Driver  True    Manual  Running OK      Normal
        False   True
ptilink Direct Parallel Link Driver
        c:\winnt\system32\drivers\ptilink.sys      Kernel
Driver  True    Manual  Running OK      Normal
        False   True
ql1080  ql1080  Not Available           Kernel Driver
        False   Disabled        Stopped OK      Normal
        False   False
ql10wnt Ql10wnt Not Available           Kernel Driver
        False   Disabled        Stopped OK      Normal
        False   False
ql1240  ql1240  Not Available           Kernel Driver
        False   Disabled        Stopped OK      Normal
        False   False
ql2100  ql2100  Not Available           Kernel Driver
        False   Disabled        Stopped OK      Normal
        False   False
ql2300  ql2300  c:\winnt\system32\drivers\ql2300.sys
        Kernel Driver           True    Boot    Running
        OK      Normal  False   True
qlvika  qlvika  c:\winnt\system32\drivers\qlvika.sys
        Kernel Driver           True    Auto    Running
        OK      Normal  False   True
rasacd  Remote Access Auto Connection Driver
        c:\winnt\system32\drivers\rasacd.sys       Kernel
Driver  True    System  Running OK      Normal
        False   True
rasl2tp WAN Miniport (L2TP)
        c:\winnt\system32\drivers\rasl2tp.sys      Kernel
Driver  True    Manual  Running OK      Normal
        False   True
raspti  Direct Parallel
        c:\winnt\system32\drivers\raspti.sys       Kernel
Driver  True    Manual  Running OK      Normal
        False   True
rca     Microsoft Streaming Network Raw Channel Access
        c:\winnt\system32\drivers\rca.sys          Kernel
Driver  False   Manual  Stopped OK      Normal
        False   False
rdbss   Rdbss   c:\winnt\system32\drivers\rdbss.sys
        File System Driver      True    System  Running
        OK      Normal  False   True
rdpdr   Terminal Server Device Redirector Driver
        c:\winnt\system32\drivers\rdpdr.sys        Kernel
Driver  True    Manual  Running OK      Normal
        False   True
rdpwd   RDPWD   c:\winnt\system32\drivers\rdpwd.sys
        Kernel Driver           False   Manual  Stopped
        OK      Ignore  False   False
redbook Digital CD Audio Playback Filter Driver
        c:\winnt\system32\drivers\redbook.sys      Kernel
Driver  False   System  Stopped OK      Normal
        False   False
serenum Serenum Filter Driver
        c:\winnt\system32\drivers\serenum.sys      Kernel
Driver  True    Manual  Running OK      Normal
        False   True
serial  Serial port driver
        c:\winnt\system32\drivers\serial.sys       Kernel
Driver  True    System  Running OK      Ignore
        False   True
sfloppy Sfloppy c:\winnt\system32\drivers\sfloppy.sys
        Kernel Driver           False   System  Stopped
        OK      Ignore  False   False
sglfb   sglfb   Not Available           Kernel Driver
        False   System  Stopped OK      Normal
        False   False
simbad  Simbad  Not Available           Kernel Driver
        False   Disabled        Stopped OK      Normal
        False   False
sparrow Sparrow Not Available           Kernel Driver
        False   Disabled        Stopped OK      Normal
        False   False
spud    Special Purpose Utility Driver
        c:\winnt\system32\drivers\spud.sys         Kernel
Driver  True    Manual  Running OK      Normal
        False   True
srv     Srv     c:\winnt\system32\drivers\srv.sys
        File System Driver      True    Manual  Running
        OK      Normal  False   True
swenum  Software Bus Driver
        c:\winnt\system32\drivers\swenum.sys       Kernel
Driver  True    Manual  Running OK      Normal
        False   True
symc810 symc810 Not Available           Kernel Driver
        False   Disabled        Stopped OK      Normal
```

```
          False     False
symc8xx   symc8xx   Not Available          Kernel Driver
          False     Disabled  Stopped     OK        Normal
          False     False
sym_hi    sym_hi    Not Available          Kernel Driver
          False     Disabled  Stopped     OK        Normal
          False     False
tcpip     TCP/IP Protocol Driver
          c:\winnt\system32\drivers\tcpip.sys          Kernel
Driver    True      System    Running     OK        Normal
          False     True
tdasync   TDASYNC   c:\winnt\system32\drivers\tdasync.sys
          Kernel Driver         False     Manual    Stopped
          OK        Ignore      False     False
tdipx     TDIPX     c:\winnt\system32\drivers\tdipx.sys
          Kernel Driver         False     Manual    Stopped
          OK        Ignore      False     False
tdnetb    TDNETB    c:\winnt\system32\drivers\tdnetb.sys
          Kernel Driver         False     Manual    Stopped
          OK        Ignore      False     False
tdpipe    TDPIPE    c:\winnt\system32\drivers\tdpipe.sys
          Kernel Driver         False     Manual    Stopped
          OK        Ignore      False     False
tdspx     TDSPX     c:\winnt\system32\drivers\tdspx.sys
          Kernel Driver         False     Manual    Stopped
          OK        Ignore      False     False
tdtcp     TDTCP     c:\winnt\system32\drivers\tdtcp.sys
          Kernel Driver         False     Manual    Stopped
          OK        Ignore      False     False
termdd    Terminal Device Driver
          c:\winnt\system32\drivers\termdd.sys          Kernel
Driver    True      Auto        Running     OK        Normal
          False     True
tga       tga       Not Available          Kernel Driver
          False     System    Stopped     OK        Ignore
          False     False
udfs      Udfs      c:\winnt\system32\drivers\udfs.sys
          File System Driver    False     Disabled  Stopped
          OK        Normal      False     False
uhcd      Microsoft USB Universal Host Controller Driver
          c:\winnt\system32\drivers\uhcd.sys          Kernel
Driver    True      Manual    Running     OK        Normal
          False     True
ultra66   ultra66   Not Available          Kernel Driver
          False     Disabled  Stopped     OK        Normal
          False     False
update    Microcode Update Driver
          c:\winnt\system32\drivers\update.sys          Kernel
Driver    True      Manual    Running     OK        Normal
          False     True
usbhub    Microsoft USB Standard Hub Driver
          c:\winnt\system32\drivers\usbhub.sys          Kernel
Driver    True      Manual    Running     OK        Normal
          False     True
vgasave   VgaSave   c:\winnt\system32\drivers\vga.sys
          Kernel Driver         True      System    Running
          OK        Ignore      False     True
wanarp    Remote Access IP ARP Driver
          c:\winnt\system32\drivers\wanarp.sys          Kernel
Driver    True      Manual    Running     OK        Normal
          False     True
wdica     WDICA     Not Available          Kernel Driver
          False     Manual    Stopped     OK        Ignore
          False     False

[Environment Variables]

Variable  Value     User Name
ComSpec   %SystemRoot%\system32\cmd.exe
          <SYSTEM>
Os2LibPath          %SystemRoot%\system32\os2\dll;
          <SYSTEM>
Path
          %SystemRoot%\system32;%SystemRoot%;%SystemR
oot%\System32\Wbem;C:\Program Files\Microsoft SQL
Server\80\Tools\BINN<SYSTEM>
windir    %SystemRoot%      <SYSTEM>
OS        Windows_NT        <SYSTEM>
PROCESSOR_ARCHITECTURE  x86       <SYSTEM>
PROCESSOR_LEVEL 15        <SYSTEM>
PROCESSOR_IDENTIFIER      x86 Family 15 Model 2 Stepping
4, GenuineIntel     <SYSTEM>
PROCESSOR_REVISION        0204      <SYSTEM>
NUMBER_OF_PROCESSORS      4         <SYSTEM>
PATHEXT
          .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WS
H         <SYSTEM>
TEMP      %SystemRoot%\TEMP         <SYSTEM>
TMP       %SystemRoot%\TEMP         <SYSTEM>
TEMP      %USERPROFILE%\Local Settings\Temp
          ACL01\Administrator
TMP       %USERPROFILE%\Local Settings\Temp
          ACL01\Administrator

[Jobs]

[ Following are sub-categories of this main category ]

[Print]

Document Size       Owner     Notify    Status    Time
Submitted Start Time Until Time Elapsed Time      Pages
Printed   Job ID    Priority  Parameters        Driver
Name      Print Processor     Host Print Queue  Data Type
          Name
No print jobs



[Network Connections]

Local Name          Remote Name         Type      Status
          User Name
No network connections information



[Running Tasks]

Name      Path      Process ID          Priority  Min
Working Set         Max Working Set     Start Time Version
          Size      File Date
system idle process Not Available       0         0
          Not Available       Not Available       Not
Available Unknown   Unknown   Unknown
system    Not Available       8         8         0
          1413120   Not Available       Unknown   Unknown
          Unknown
smss.exe  c:\winnt\system32\smss.exe      188       11
          204800    1413120   4/20/2003 8:22:17 AM
          5.00.2195.2901      44.27 KB (45,328 bytes)
          12/8/1999 5:00:00 AM
csrss.exe Not Available       216       13        Not
Available Not Available       4/20/2003 8:22:22 AM Unknown
          Unknown   Unknown
winlogon.exe        c:\winnt\system32\winlogon.exe 212
          13        204800    1413120   4/20/2003 8:22:24 AM
          5.00.2195.2953      173.77 KB (177,936 bytes)
          12/8/1999 5:00:00 AM
services.exe        c:\winnt\system32\services.exe 268
          9         204800    1413120   4/20/2003 8:22:24 AM
          5.00.2195.2780      86.77 KB (88,848 bytes)
          12/8/1999 5:00:00 AM
lsass.exe c:\winnt\system32\lsass.exe     280       9
          204800    1413120   4/20/2003 8:22:25 AM
          5.00.2195.2964      32.77 KB (33,552 bytes)
          12/8/1999 5:00:00 AM
svchost.exe         c:\winnt\system32\svchost.exe 460
          8         204800    1413120   4/20/2003 8:22:27 AM
          5.00.2134.1         7.77 KB (7,952 bytes) 12/8/1999
5:00:00 AM
spoolsv.exe         c:\winnt\system32\spoolsv.exe 488
          8         204800    1413120   4/20/2003 8:22:28 AM
          5.00.2161.1         43.77 KB (44,816 bytes)
          1/7/2003 11:17:31 PM
msdtc.exe c:\winnt\system32\msdtc.exe     516       8
          204800    1413120   4/20/2003 8:22:28 AM
          1999.9.3421.3       6.77 KB (6,928 bytes) 1/7/2003
11:25:57 PM
svchost.exe         c:\winnt\system32\svchost.exe 680
          8         204800    1413120   4/20/2003 8:22:29 AM
          5.00.2134.1         7.77 KB (7,952 bytes) 12/8/1999
5:00:00 AM
llssrv.exe c:\winnt\system32\llssrv.exe   700       9
          204800    1413120   4/20/2003 8:22:29 AM
          5.00.2195.2649      114.27 KB (117,008 bytes)
          5/4/2001 12:05:02 PM
regsvc.exe          c:\winnt\system32\regsvc.exe  756
          8         204800    1413120   4/20/2003 8:22:29 AM
          5.00.2195.2104      65.27 KB (66,832 bytes)
          1/7/2003 2:55:00 PM
rsys.exe  Not Available       836       8         Not
Available Not Available       4/20/2003 8:22:31 AM Unknown
          Unknown   Unknown
mstask.exe          c:\winnt\system32\mstask.exe  864
          8         204800    1413120   4/20/2003 8:22:40 AM
          4.71.2195.1         115.27 KB (118,032 bytes)
          1/7/2003 2:54:53 PM
winmgmt.exe         c:\winnt\system32\wbem\winmgmt.exe
          944       8         204800    1413120   4/20/2003
8:22:40 AM          1.50.1085.0029      192.08 KB (196,685
bytes)    1/7/2003 2:55:11 PM
inetinfo.exe        c:\winnt\system32\inetsrv\inetinfo.exe
          960       8         204800    1413120   4/20/2003
```

8:22:40 AM 5.00.0984 14.27 KB (14,608 bytes)
1/7/2003 2:56:13 PM
dfssvc.exe c:\winnt\system32\dfssvc.exe 1036 8
204800 1413120 4/20/2003 8:22:41 AM
5.00.2195.2841 88.27 KB (90,384 bytes)
1/7/2003 2:54:39 PM
svchost.exe c:\winnt\system32\svchost.exe 1232
8 204800 1413120 4/20/2003 8:23:26 AM
5.00.2134.1 7.77 KB (7,952 bytes) 12/8/1999
5:00:00 AM
explorer.exe c:\winnt\explorer.exe 732 8
204800 1413120 4/20/2003 8:35:13 AM
5.00.3315.2846 237.27 KB (242,960 bytes)
1/7/2003 2:55:07 PM
cmd.exe c:\winnt\system32\cmd.exe 1348 8
204800 1413120 4/20/2003 8:35:15 AM
5.00.2195.2104 230.77 KB (236,304 bytes)
12/8/1999 5:00:00 AM
dllhost.exe Not Available 468 8
Not Available Not Available 4/20/2003
8:49:51 AM Unknown Unknown Unknown
mmc.exe c:\winnt\system32\mmc.exe 6256 8
204800 1413120 4/20/2003 10:15:22 AM
5.00.2195.2301 589.27 KB (603,408 bytes)
1/7/2003 2:54:47 PM
rsvp.exe c:\winnt\system32\rsvp.exe 6284 8
204800 1413120 4/20/2003 10:19:17 AM
5.00.2167.1 172.77 KB (176,912 bytes)
12/8/1999 5:00:00 AM

[Loaded Modules]

Name Version Size File Date Manufacturer
Path
traffic.dll 5.00.2139.1 30.77 KB (31,504 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\traffic.dll
rsvp.exe 5.00.2167.1 172.77 KB (176,912 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\rsvp.exe
wbemprox.dll 1.50.1085.0045 40.08 KB (41,040
bytes) 1/7/2003 2:55:11 PM Microsoft Corporation
c:\winnt\system32\wbem\wbemprox.dll
mlang.dll 5.00.3103.1000 510.77 KB (523,024 bytes)
1/7/2003 2:54:47 PM Microsoft Corporation
c:\winnt\system32\mlang.dll
cabinet.dll 5.00.2147.1 54.77 KB (56,080 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\cabinet.dll
msinfo32.dll 5.00.2177.1 312.27 KB (319,760
bytes) 1/7/2003 2:30:36 PM Microsoft Corporation
c:\program files\common files\microsoft
shared\msinfo\msinfo32.dll
mmcndmgr.dll 5.00.2178.1 815.27 KB (834,832
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\mmcndmgr.dll
mmc.exe 5.00.2195.2301 589.27 KB (603,408 bytes)
1/7/2003 2:54:47 PM Microsoft Corporation
c:\winnt\system32\mmc.exe
cmd.exe 5.00.2195.2104 230.77 KB (236,304 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\cmd.exe
urlmon.dll 5.00.3315.1000 441.27 KB (451,856 bytes)
1/7/2003 2:55:05 PM Microsoft Corporation
c:\winnt\system32\urlmon.dll
faxshell.dll 5.00.2134.1 8.27 KB (8,464 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\faxshell.dll
msacm32.dll 5.00.2134.1 65.27 KB (66,832
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\msacm32.dll
avifil32.dll 5.00.2134.1 76.27 KB (78,096 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\avifil32.dll
msvfw32.dll 5.00.2134.1 113.77 KB (116,496
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\msvfw32.dll
docprop2.dll 5.00.2178.1 297.77 KB (304,912
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\docprop2.dll
browselc.dll 5.00.3315.2846 34.50 KB (35,328
bytes) 1/7/2003 2:54:35 PM Microsoft Corporation
c:\winnt\system32\browselc.dll
shdoclc.dll 5.00.3315.2879 324.50 KB (332,288
bytes) 1/7/2003 2:55:02 PM Microsoft Corporation
c:\winnt\system32\shdoclc.dll
wininet.dll 5.00.3315.1000 456.77 KB (467,728 bytes)
1/7/2003 2:55:06 PM Microsoft Corporation
c:\winnt\system32\wininet.dll
linkinfo.dll 5.00.2134.1 15.77 KB (16,144 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\linkinfo.dll
powrprof.dll 5.00.3103.1000 13.27 KB (13,584
bytes) 1/7/2003 2:54:59 PM Microsoft Corporation
c:\winnt\system32\powrprof.dll
batmeter.dll 5.00.3103.1000 20.27 KB (20,752
bytes) 1/7/2003 2:54:35 PM Microsoft Corporation
c:\winnt\system32\batmeter.dll
stobject.dll 5.00.2195.2780 79.27 KB (81,168
bytes) 1/7/2003 2:55:03 PM Microsoft Corporation
c:\winnt\system32\stobject.dll
webcheck.dll 5.00.3315.1000 251.77 KB (257,808
bytes) 1/7/2003 2:55:06 PM Microsoft Corporation
c:\winnt\system32\webcheck.dll
ntshrui.dll 5.00.2134.1 46.77 KB (47,888 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\ntshrui.dll
mydocs.dll 5.00.2920.0000 55.77 KB (57,104
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\mydocs.dll
browseui.dll 5.00.3315.2846 788.77 KB (807,696
bytes) 1/7/2003 2:54:35 PM Microsoft Corporation
c:\winnt\system32\browseui.dll
shdocvw.dll 5.00.3315.2879 1.05 MB (1,104,144
bytes) 1/7/2003 2:55:02 PM Microsoft Corporation
c:\winnt\system32\shdocvw.dll
explorer.exe 5.00.3315.2846 237.27 KB (242,960
bytes) 1/7/2003 2:55:07 PM Microsoft Corporation
c:\winnt\explorer.exe
tapisrv.dll 5.00.2195.2955 169.27 KB (173,328 bytes)
1/7/2003 2:55:04 PM Microsoft Corporation
c:\winnt\system32\tapisrv.dll
dfssvc.exe 5.00.2195.2841 88.27 KB (90,384 bytes)
1/7/2003 2:54:39 PM Microsoft Corporation
c:\winnt\system32\dfssvc.exe
tpcc_com_all.dll 1, 0, 0, 1 80.00 KB (81,920 bytes)
1/8/2003 4:28:12 PM
c:\inetpub\wwwroot\tpcc_c~2.dll
qlvipl.dll Not Available 92.05 KB (94,262 bytes)
3/5/2003 3:50:01 PM Not Available
c:\winnt\system32\qlvipl.dll
dbmsqlgc.dll 2000.080.0760.00 32.56 KB (33,340
bytes) 1/29/2003 6:28:17 PM Microsoft Corporation
c:\winnt\system32\dbmsqlgc.dll
dbnetlib.dll 2000.081.9031 60.00 KB (61,440
bytes) 9/27/2002 11:22:44 AM Microsoft
Corporation c:\winnt\system32\dbnetlib.dll
ntwdblib.dll 2000.080.0194.00 268.06 KB (274,489
bytes) 1/8/2003 4:25:00 PM Microsoft Corporation
c:\winnt\system32\ntwdblib.dll
tpcc_dblib.dll Not Available 28.00 KB (28,672
bytes) 1/8/2003 4:28:10 PM Not Available
c:\inetpub\wwwroot\tpcc_dblib.dll
tpcc_com.dll Not Available 24.00 KB (24,576
bytes) 1/8/2003 4:28:11 PM Not Available
c:\inetpub\wwwroot\tpcc_com.dll
tpcc.dll 0, 4, 0, 0 92.00 KB (94,208 bytes) 1/8/2003
4:28:10 PM Microsoft c:\inetpub\wwwroot\tpcc.dll
mfc42.dll 6.00.8665.0 972.05 KB (995,383 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\mfc42.dll
wam.dll 5.00.0984 70.77 KB (72,464 bytes) 1/7/2003
2:56:15 PM Microsoft Corporation
c:\winnt\system32\inetsrv\wam.dll
odbcint.dll 3.520.9001.0 88.00 KB (90,112 bytes)
1/29/2003 6:28:07 PM Microsoft Corporation
c:\winnt\system32\odbcint.dll
comdlg32.dll 5.00.3103.1000 236.77 KB (242,448
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\comdlg32.dll
odbc32.dll 3.520.9030.0 196.00 KB (200,704 bytes)
1/29/2003 6:28:07 PM Microsoft Corporation
c:\winnt\system32\odbc32.dll
comsvcs.dll 2000.2.3471.1 1.35 MB (1,417,488
bytes) 1/7/2003 2:54:38 PM Microsoft Corporation
c:\winnt\system32\comsvcs.dll
iislog.dll 5.00.0984 75.27 KB (77,072 bytes) 1/7/2003
2:56:13 PM Microsoft Corporation
c:\winnt\system32\inetsrv\iislog.dll
ntfsdrv.dll 5.00.0984 36.77 KB (37,648 bytes) 1/7/2003
2:56:14 PM Microsoft Corporation
c:\winnt\system32\inetsrv\ntfsdrv.dll
ntlsapi.dll 5.00.2134.1 6.77 KB (6,928 bytes) 12/8/1999
5:00:00 AM Microsoft Corporation
c:\winnt\system32\ntlsapi.dll
aqueue.dll 5.00.0984 312.77 KB (320,272 bytes) 1/7/2003
2:56:04 PM Microsoft Corporation
c:\winnt\system32\inetsrv\aqueue.dll
httpext.dll 0.9.3940.21 435.27 KB (445,712 bytes)
1/7/2003 2:56:12 PM Microsoft Corporation
c:\winnt\system32\inetsrv\httpext.dll
fpexedll.dll 4.0.2.4324 20.06 KB (20,541 bytes)
1/7/2003 2:56:02 PM Microsoft Corporation

c:\program files\common files\microsoft shared\web server extensions\40\bin\fpexedll.dll
md5filt.dll 5.00.0984 32.77 KB (33,552 bytes) 1/7/2003 2:56:13 PM Microsoft Corporation
c:\winnt\system32\inetsrv\md5filt.dll
gzip.dll 5.00.0984 30.27 KB (30,992 bytes) 1/7/2003 2:56:12 PM Microsoft Corporation
c:\winnt\system32\inetsrv\gzip.dll
compfilt.dll 5.00.0984 22.77 KB (23,312 bytes) 1/7/2003 2:56:12 PM Microsoft Corporation
c:\winnt\system32\inetsrv\compfilt.dll
seo.dll 5.00.0984 229.27 KB (234,768 bytes) 1/7/2003 2:56:14 PM Microsoft Corporation
c:\winnt\system32\inetsrv\seo.dll
sspifilt.dll 5.00.0984 43.27 KB (44,304 bytes) 1/7/2003 2:56:14 PM Microsoft Corporation
c:\winnt\system32\inetsrv\sspifilt.dll
iscomlog.dll 5.00.0984 24.77 KB (25,360 bytes) 1/7/2003 2:56:13 PM Microsoft Corporation
c:\winnt\system32\inetsrv\iscomlog.dll
lonsint.dll 5.00.0984 11.77 KB (12,048 bytes) 1/7/2003 2:56:13 PM Microsoft Corporation
c:\winnt\system32\inetsrv\lonsint.dll
inetsloc.dll 5.00.0984 20.27 KB (20,752 bytes) 1/7/2003 2:54:45 PM Microsoft Corporation
c:\winnt\system32\inetsloc.dll
w3svc.dll 5.00.0984 343.27 KB (351,504 bytes) 1/7/2003 2:56:15 PM Microsoft Corporation
c:\winnt\system32\inetsrv\w3svc.dll
staxmem.dll 5.00.0984 8.27 KB (8,464 bytes) 1/7/2003 11:26:44 PM Microsoft Corporation
c:\winnt\system32\staxmem.dll
exstrace.dll 5.00.0984 13.77 KB (14,096 bytes) 1/7/2003 11:26:47 PM Microsoft Corporation
c:\winnt\system32\exstrace.dll
rwnh.dll 5.00.0984 10.77 KB (11,024 bytes) 1/7/2003 11:28:02 PM Microsoft Corporation
c:\winnt\system32\rwnh.dll
fcachdll.dll 5.00.0984 43.77 KB (44,816 bytes) 1/7/2003 11:28:01 PM Microsoft Corporation
c:\winnt\system32\fcachdll.dll
iisfecnv.dll 5.00.0984 7.27 KB (7,440 bytes) 1/7/2003 11:26:45 PM Microsoft Corporation
c:\winnt\system32\inetsrv\iisfecnv.dll
isatq.dll 5.00.0984 60.27 KB (61,712 bytes) 1/7/2003 2:56:13 PM Microsoft Corporation
c:\winnt\system32\inetsrv\isatq.dll
infocomm.dll 5.00.0984 238.27 KB (243,984 bytes) 1/7/2003 2:56:13 PM Microsoft Corporation
c:\winnt\system32\inetsrv\infocomm.dll
smtpsvc.dll 5.00.0984 420.27 KB (430,352 bytes) 1/7/2003 2:56:14 PM Microsoft Corporation
c:\winnt\system32\inetsrv\smtpsvc.dll
security.dll 5.00.2154.1 5.77 KB (5,904 bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\security.dll
svcext.dll 5.00.0984 39.77 KB (40,720 bytes) 1/7/2003 2:56:14 PM Microsoft Corporation
c:\winnt\system32\inetsrv\svcext.dll
admexs.dll 5.00.0984 27.77 KB (28,432 bytes) 1/7/2003 2:56:11 PM Microsoft Corporation
c:\winnt\system32\inetsrv\admexs.dll
wamreg.dll 5.00.0984 45.77 KB (46,864 bytes) 1/7/2003 2:56:15 PM Microsoft Corporation
c:\winnt\system32\inetsrv\wamreg.dll
metadata.dll 5.00.0984 68.77 KB (70,416 bytes) 1/7/2003 2:56:13 PM Microsoft Corporation
c:\winnt\system32\inetsrv\metadata.dll
iismap.dll 5.00.0984 55.77 KB (57,104 bytes) 1/7/2003 2:54:44 PM Microsoft Corporation
c:\winnt\system32\iismap.dll
nsepm.dll 5.00.0984 43.27 KB (44,304 bytes) 1/7/2003 2:56:14 PM Microsoft Corporation
c:\winnt\system32\inetsrv\nsepm.dll
admwprox.dll 5.00.0984 31.77 KB (32,528 bytes) 1/7/2003 11:26:47 PM Microsoft Corporation
c:\winnt\system32\admwprox.dll
coadmin.dll 5.00.0984 39.27 KB (40,208 bytes) 1/7/2003 2:56:11 PM Microsoft Corporation
c:\winnt\system32\inetsrv\coadmin.dll
iisadmin.dll 5.00.0984 15.27 KB (15,632 bytes) 1/7/2003 2:56:12 PM Microsoft Corporation
c:\winnt\system32\inetsrv\iisadmin.dll
rpcref.dll 5.00.0984 4.27 KB (4,368 bytes) 1/7/2003 2:56:14 PM Microsoft Corporation
c:\winnt\system32\inetsrv\rpcref.dll
iisrtl.dll 5.00.0984 119.77 KB (122,640 bytes) 1/7/2003 2:54:44 PM Microsoft Corporation
c:\winnt\system32\iisrtl.dll
inetinfo.exe 5.00.0984 14.27 KB (14,608 bytes) 1/7/2003 2:56:13 PM Microsoft Corporation
c:\winnt\system32\inetsrv\inetinfo.exe
netui1.dll 5.00.2134.1 210.27 KB (215,312 bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\netui1.dll
netui0.dll 5.00.2134.1 70.27 KB (71,952 bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\netui0.dll
ntlanman.dll 5.00.2157.1 35.27 KB (36,112 bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\ntlanman.dll
perfos.dll 5.00.2155.1 21.27 KB (21,776 bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\perfos.dll
wshnetbs.dll 5.00.2134.1 7.77 KB (7,952 bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\wshnetbs.dll
ntmarta.dll 5.00.2195.2862 98.77 KB (101,136 bytes) 1/7/2003 2:54:56 PM Microsoft Corporation
c:\winnt\system32\ntmarta.dll
provthrd.dll 1.50.1085.0000 68.07 KB (69,708 bytes) 1/7/2003 2:30:24 PM Microsoft Corporation
c:\winnt\system32\wbem\provthrd.dll
ntevt.dll 1.50.1085.0000 192.06 KB (196,669 bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\ntevt.dll
psapi.dll 5.00.2134.1 28.27 KB (28,944 bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\psapi.dll
framedyn.dll 1.50.1085.0000 164.05 KB (167,992 bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\wbem\framedyn.dll
cimwin32.dll 1.50.1085.0038 1.02 MB (1,073,232 bytes) 1/7/2003 2:55:10 PM Microsoft Corporation
c:\winnt\system32\wbem\cimwin32.dll
wbemsvc.dll 1.50.1085.0007 40.07 KB (41,036 bytes) 1/7/2003 2:55:11 PM Microsoft Corporation
c:\winnt\system32\wbem\wbemsvc.dll
wbemess.dll 1.50.1085.0039 364.07 KB (372,804 bytes) 1/7/2003 2:55:11 PM Microsoft Corporation
c:\winnt\system32\wbem\wbemess.dll
fastprox.dll 1.50.1085.0037 144.08 KB (147,536 bytes) 1/7/2003 2:55:10 PM Microsoft Corporation
c:\winnt\system32\wbem\fastprox.dll
wbemcore.dll 1.50.1085.0036 628.07 KB (643,140 bytes) 1/7/2003 2:55:11 PM Microsoft Corporation
c:\winnt\system32\wbem\wbemcore.dll
wbemcomn.dll 1.50.1085.0021 692.07 KB (708,675 bytes) 1/7/2003 2:55:11 PM Microsoft Corporation
c:\winnt\system32\wbem\wbemcomn.dll
winmgmt.exe 1.50.1085.0029 192.08 KB (196,685 bytes) 1/7/2003 2:55:11 PM Microsoft Corporation
c:\winnt\system32\wbem\winmgmt.exe
msidle.dll 5.00.2920.0000 6.27 KB (6,416 bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\msidle.dll
mstask.exe 4.71.2195.1 115.27 KB (118,032 bytes) 1/7/2003 2:54:53 PM Microsoft Corporation
c:\winnt\system32\mstask.exe
regsvc.exe 5.00.2195.2104 65.27 KB (66,832 bytes) 1/7/2003 2:55:00 PM Microsoft Corporation
c:\winnt\system32\regsvc.exe
llsrpc.dll 5.00.2149.1 45.77 KB (46,864 bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\llsrpc.dll
llssrv.exe 5.00.2195.2649 114.27 KB (117,008 bytes) 5/4/2001 12:05:02 PM Microsoft Corporation
c:\winnt\system32\llssrv.exe
wmi.dll 5.00.2191.1 6.27 KB (6,416 bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\wmi.dll
netshell.dll 5.00.2195.2779 457.27 KB (468,240 bytes) 1/7/2003 2:54:54 PM Microsoft Corporation
c:\winnt\system32\netshell.dll
netman.dll 5.00.2195.2779 89.27 KB (91,408 bytes) 1/7/2003 2:54:54 PM Microsoft Corporation
c:\winnt\system32\netman.dll
rasdlg.dll 5.00.2195.2671 514.27 KB (526,608 bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\rasdlg.dll
netcfgx.dll 5.00.2195.2228 534.77 KB (547,600 bytes) 1/7/2003 2:54:54 PM Microsoft Corporation
c:\winnt\system32\netcfgx.dll
rasmans.dll 5.00.2195.2728 147.27 KB (150,800 bytes) 1/7/2003 2:54:59 PM Microsoft Corporation
c:\winnt\system32\rasmans.dll
ntmsdba.dll 5.00.2195.2779 167.27 KB (171,280 bytes) 1/7/2003 2:54:56 PM Microsoft Corporation
c:\winnt\system32\ntmsdba.dll
sens.dll 5.00.2163.1 36.77 KB (37,648 bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\sens.dll
ntmssvc.dll 5.00.2195.2779 391.27 KB (400,656

bytes) 1/7/2003 2:54:56 PM Microsoft Corporation
c:\winnt\system32\ntmssvc.dll
es.dll 2000.2.3471.1 222.27 KB (227,600 bytes)
1/7/2003 2:54:42 PM Microsoft Corporation
c:\winnt\system32\es.dll
mtxoci.dll 2000.2.3471.1 101.77 KB (104,208 bytes)
1/7/2003 2:54:54 PM Microsoft Corporation
c:\winnt\system32\mtxoci.dll
resutils.dll 5.00.2195.2787 39.77 KB (40,720 bytes)
1/7/2003 2:55:00 PM Microsoft Corporation
c:\winnt\system32\resutils.dll
clusapi.dll 5.00.2195.2104 54.27 KB (55,568 bytes)
1/7/2003 2:54:37 PM Microsoft Corporation
c:\winnt\system32\clusapi.dll
msvcp50.dll 5.00.7051 552.50 KB (565,760 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\msvcp50.dll
xolehlp.dll 1999.9.3421.3 17.27 KB (17,680 bytes)
1/7/2003 11:25:57 PM Microsoft Corporation
c:\winnt\system32\xolehlp.dll
msdtclog.dll 1999.9.3421.3 89.77 KB (91,920
bytes) 1/7/2003 11:25:57 PM Microsoft Corporation
c:\winnt\system32\msdtclog.dll
mtxclu.dll 2000.2.3471.1 51.27 KB (52,496 bytes)
1/7/2003 2:54:54 PM Microsoft Corporation
c:\winnt\system32\mtxclu.dll
msdtcprx.dll 2000.2.3471.1 665.77 KB (681,744
bytes) 1/7/2003 2:54:49 PM Microsoft Corporation
c:\winnt\system32\msdtcprx.dll
txfaux.dll 2000.2.3471.1 374.27 KB (383,248 bytes)
1/7/2003 2:55:05 PM Microsoft Corporation
c:\winnt\system32\txfaux.dll
msdtctm.dll 2000.2.3471.1 1.07 MB (1,120,528
bytes) 1/7/2003 2:54:49 PM Microsoft Corporation
c:\winnt\system32\msdtctm.dll
msdtc.exe 1999.9.3421.3 6.77 KB (6,928 bytes) 1/7/2003
11:25:57 PM Microsoft Corporation
c:\winnt\system32\msdtc.exe
inetpp.dll 5.00.2195.2842 65.27 KB (66,832 bytes)
1/7/2003 2:54:45 PM Microsoft Corporation
c:\winnt\system32\inetpp.dll
win32spl.dll 5.00.2195.2780 92.27 KB (94,480
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\win32spl.dll
usbmon.dll 5.00.2195.2780 11.27 KB (11,536
bytes) 1/7/2003 2:55:05 PM Microsoft Corporation
c:\winnt\system32\usbmon.dll
tcpmon.dll 5.00.2195.2780 40.77 KB (41,744 bytes)
1/7/2003 2:55:04 PM Microsoft Corporation
c:\winnt\system32\tcpmon.dll
pjlmon.dll 5.00.2165.1 12.77 KB (13,072 bytes)
12/1/1999 8:39:36 AM Microsoft Corporation
c:\winnt\system32\pjlmon.dll
cnbjmon.dll 5.00.2134.1 43.77 KB (44,816
bytes) 12/1/1999 8:38:48 AM Microsoft Corporation
c:\winnt\system32\cnbjmon.dll
localspl.dll 5.00.2195.2793 246.77 KB (252,688
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\localspl.dll
spoolss.dll 5.00.2161.1 61.77 KB (63,248
bytes) 1/7/2003 11:17:31 PM Microsoft Corporation
c:\winnt\system32\spoolss.dll
spoolsv.exe 5.00.2161.1 43.77 KB (44,816
bytes) 1/7/2003 11:17:31 PM Microsoft Corporation
c:\winnt\system32\spoolsv.exe
rpcss.dll 5.00.2195.2815 231.27 KB (236,816 bytes)
1/7/2003 2:55:00 PM Microsoft Corporation
c:\winnt\system32\rpcss.dll
svchost.exe 5.00.2134.1 7.77 KB (7,952 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\svchost.exe
dssenh.dll 5.00.2195.2228 142.77 KB (146,192 bytes)
1/7/2003 2:56:07 PM Microsoft Corporation
c:\winnt\system32\dssenh.dll
oakley.dll 5.00.2195.2785 378.77 KB (387,856 bytes)
1/7/2003 2:54:56 PM Microsoft Corporation
c:\winnt\system32\oakley.dll
mfc42u.dll 6.00.8665.0 972.05 KB (995,384 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\mfc42u.dll
polagent.dll 5.00.2183.1 108.27 KB (110,864
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\polagent.dll
scecli.dll 5.00.2195.2780 105.27 KB (107,792 bytes)
1/7/2003 2:55:01 PM Microsoft Corporation
c:\winnt\system32\scecli.dll
atl.dll 3.00.8449 57.56 KB (58,938 bytes) 12/8/1999
5:00:00 AM Microsoft Corporation
c:\winnt\system32\atl.dll
certcli.dll 5.00.2195.2778 130.77 KB (133,904 bytes)
1/7/2003 2:54:37 PM Microsoft Corporation
c:\winnt\system32\certcli.dll
esent.dll 6.0.3940.13 1.08 MB (1,135,376 bytes)
1/7/2003 2:54:42 PM Microsoft Corporation
c:\winnt\system32\esent.dll
ntdsatq.dll 5.00.2195.2878 31.27 KB (32,016 bytes)
1/7/2003 2:54:55 PM Microsoft Corporation
c:\winnt\system32\ntdsatq.dll
ntdsa.dll 5.00.2195.2899 990.77 KB (1,014,544 bytes)
1/7/2003 2:54:55 PM Microsoft Corporation
c:\winnt\system32\ntdsa.dll
kdcsvc.dll 5.00.2195.2878 137.77 KB (141,072 bytes)
1/7/2003 2:54:47 PM Microsoft Corporation
c:\winnt\system32\kdcsvc.dll
sfmapi.dll 5.00.2134.1 38.77 KB (39,696 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\sfmapi.dll
rassfm.dll 5.00.2195.2671 21.27 KB (21,776 bytes)
1/7/2003 2:55:00 PM Microsoft Corporation
c:\winnt\system32\rassfm.dll
mpr.dll 5.00.2195.2779 53.27 KB (54,544 bytes)
1/7/2003 2:54:48 PM Microsoft Corporation
c:\winnt\system32\mpr.dll
rsabase.dll 5.00.2195.2228 128.27 KB (131,344
bytes) 5/4/2001 12:05:02 PM Microsoft Corporation
c:\winnt\system32\rsabase.dll
schannel.dll 5.00.2195.2922 138.27 KB (141,584
bytes) 5/4/2001 12:05:02 PM Microsoft Corporation
c:\winnt\system32\schannel.dll
netlogon.dll 5.00.2195.2865 357.77 KB (366,352
bytes) 1/7/2003 2:54:54 PM Microsoft Corporation
c:\winnt\system32\netlogon.dll
msv1_0.dll 5.00.2195.2900 111.77 KB (114,448 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\msv1_0.dll
kerberos.dll 5.00.2195.2913 198.77 KB (203,536
bytes) 1/7/2003 2:54:47 PM Microsoft Corporation
c:\winnt\system32\kerberos.dll
msprivs.dll 5.00.2154.1 41.50 KB (42,496
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\msprivs.dll
samsrv.dll 5.00.2195.2918 369.77 KB (378,640 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\samsrv.dll
lsasrv.dll 5.00.2195.2964 492.77 KB (504,592 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\lsasrv.dll
lsass.exe 5.00.2195.2964 32.77 KB (33,552 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\lsass.exe
msi.dll 2.0.2600.0 1.90 MB (1,991,168 bytes) 1/7/2003
2:54:50 PM Microsoft Corporation
c:\winnt\system32\msi.dll
appmgmts.dll 5.00.2168.1 117.77 KB (120,592
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\appmgmts.dll
xactsrv.dll 5.00.2134.1 90.27 KB (92,432 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\xactsrv.dll
wmicore.dll 5.00.2195.2842 72.27 KB (74,000
bytes) 1/7/2003 2:55:06 PM Microsoft Corporation
c:\winnt\system32\wmicore.dll
rasadhlp.dll 5.00.2168.1 7.27 KB (7,440 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\rasadhlp.dll
winrnr.dll 5.00.2160.1 18.77 KB (19,216 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\winrnr.dll
rnr20.dll 5.00.2195.2871 35.77 KB (36,624 bytes)
1/7/2003 2:55:00 PM Microsoft Corporation
c:\winnt\system32\rnr20.dll
wshtcpip.dll 5.00.2195.2104 17.27 KB (17,680
bytes) 1/7/2003 2:55:07 PM Microsoft Corporation
c:\winnt\system32\wshtcpip.dll
msafd.dll 5.00.2195.2779 106.77 KB (109,328 bytes)
1/7/2003 2:54:48 PM Microsoft Corporation
c:\winnt\system32\msafd.dll
mswsock.dll 5.00.2195.2871 62.77 KB (64,272
bytes) 1/7/2003 2:54:53 PM Microsoft Corporation
c:\winnt\system32\mswsock.dll
msgsvc.dll 5.00.2195.2939 34.27 KB (35,088
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\msgsvc.dll
browser.dll 5.00.2195.2778 48.27 KB (49,424
bytes) 1/7/2003 2:54:35 PM Microsoft Corporation
c:\winnt\system32\browser.dll
alrsvc.dll 5.00.2134.1 17.77 KB (18,192 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\alrsvc.dll
trkwks.dll 5.00.2166.1 88.77 KB (90,896 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\trkwks.dll
seclogon.dll 5.00.2135.1 15.77 KB (16,144

bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\seclogon.dll
psbase.dll 5.00.2195.2779 111.77 KB (114,448 bytes)
1/7/2003 2:54:59 PM Microsoft Corporation
c:\winnt\system32\psbase.dll
cryptsvc.dll 5.00.2181.1 61.77 KB (63,248
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\cryptsvc.dll
cryptdll.dll 5.00.2135.1 41.27 KB (42,256 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\cryptdll.dll
wkssvc.dll 5.00.2195.2780 95.27 KB (97,552 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\wkssvc.dll
srvsvc.dll 5.00.2195.2904 79.27 KB (81,168 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\srvsvc.dll
cfgmgr32.dll 5.00.2134.1 16.77 KB (17,168
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\cfgmgr32.dll
dmserver.dll 2195.2778.297.3 11.77 KB (12,048
bytes) 1/7/2003 2:54:40 PM VERITAS Software Corp.
c:\winnt\system32\dmserver.dll
lmhsvc.dll 5.00.2195.2778 9.77 KB (10,000 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\lmhsvc.dll
dnsrslvr.dll 5.00.2195.2778 88.77 KB (90,896
bytes) 1/7/2003 2:54:40 PM Microsoft Corporation
c:\winnt\system32\dnsrslvr.dll
tapi32.dll 5.00.2182.1 123.27 KB (126,224 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\tapi32.dll
rasman.dll 5.00.2195.2780 54.77 KB (56,080 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\rasman.dll
rasapi32.dll 5.00.2195.2671 189.77 KB (194,320
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\rasapi32.dll
rtutils.dll 5.00.2168.1 43.77 KB (44,816 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\rtutils.dll
adsldpc.dll 5.00.2195.2842 127.27 KB (130,320
bytes) 1/7/2003 2:54:34 PM Microsoft Corporation
c:\winnt\system32\adsldpc.dll
activeds.dll 5.00.2195.2778 174.77 KB (178,960
bytes) 1/7/2003 2:54:31 PM Microsoft Corporation
c:\winnt\system32\activeds.dll
mprapi.dll 5.00.2181.1 79.27 KB (81,168 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\mprapi.dll
iphlpapi.dll 5.00.2173.2 67.77 KB (69,392
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\iphlpapi.dll
icmp.dll 5.00.2134.1 7.27 KB (7,440 bytes) 12/8/1999
5:00:00 AM Microsoft Corporation
c:\winnt\system32\icmp.dll
dhcpcsvc.dll 5.00.2195.2778 88.77 KB (90,896
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\dhcpcsvc.dll
eventlog.dll 5.00.2178.1 43.77 KB (44,816
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\eventlog.dll
ntdsapi.dll 5.00.2195.2661 55.77 KB (57,104 bytes)
1/7/2003 2:54:55 PM Microsoft Corporation
c:\winnt\system32\ntdsapi.dll
scesrv.dll 5.00.2195.2780 226.27 KB (231,696 bytes)
1/7/2003 2:55:01 PM Microsoft Corporation
c:\winnt\system32\scesrv.dll
umpnpmgr.dll 5.00.2182.1 86.27 KB (88,336
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\umpnpmgr.dll
services.exe 5.00.2195.2780 86.77 KB (88,848
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\services.exe
clbcatq.dll 2000.2.3471.1 496.77 KB (508,688 bytes)
1/7/2003 2:54:37 PM Microsoft Corporation
c:\winnt\system32\clbcatq.dll
oleaut32.dll 2.40.4517 612.27 KB (626,960 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\oleaut32.dll
cscui.dll 5.00.2195.2959 228.27 KB (233,744 bytes)
1/7/2003 2:54:38 PM Microsoft Corporation
c:\winnt\system32\cscui.dll
winspool.drv 5.00.2195.2780 109.77 KB (112,400
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\winspool.drv
winscard.dll 5.00.2134.1 77.27 KB (79,120
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\winscard.dll
wlnotify.dll 5.00.2195.2780 53.77 KB (55,056
bytes) 1/7/2003 2:55:06 PM Microsoft Corporation
c:\winnt\system32\wlnotify.dll
cscdll.dll 5.00.2195.2401 98.27 KB (100,624 bytes)
1/7/2003 2:54:38 PM Microsoft Corporation
c:\winnt\system32\cscdll.dll
lz32.dll 5.00.2134.1 9.77 KB (10,000 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\lz32.dll
version.dll 5.00.2134.1 15.77 KB (16,144 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\version.dll
rsaenh.dll 5.00.2195.2228 130.77 KB (133,904 bytes)
1/7/2003 2:56:07 PM Microsoft Corporation
c:\winnt\system32\rsaenh.dll
mscat32.dll 5.131.2134.1 7.77 KB (7,952 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\mscat32.dll
ole32.dll 5.00.2195.2887 969.77 KB (993,040 bytes)
1/7/2003 2:54:58 PM Microsoft Corporation
c:\winnt\system32\ole32.dll
imagehlp.dll 5.00.2195.2778 125.77 KB (128,784
bytes) 5/4/2001 12:05:02 PM Microsoft Corporation
c:\winnt\system32\imagehlp.dll
msasn1.dll 5.00.2134.1 51.27 KB (52,496
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\msasn1.dll
crypt32.dll 5.131.2195.2833 451.27 KB (462,096 bytes)
1/7/2003 2:54:38 PM Microsoft Corporation
c:\winnt\system32\crypt32.dll
wintrust.dll 5.131.2195.2779 162.27 KB (166,160
bytes) 1/7/2003 2:55:06 PM Microsoft Corporation
c:\winnt\system32\wintrust.dll
shlwapi.dll 5.00.3315.1000 282.77 KB (289,552
bytes) 1/7/2003 2:55:03 PM Microsoft Corporation
c:\winnt\system32\shlwapi.dll
shell32.dll 5.00.3315.2902 2.25 MB (2,359,056 bytes)
1/7/2003 2:55:02 PM Microsoft Corporation
c:\winnt\system32\shell32.dll
msgina.dll 5.00.2195.2779 324.27 KB (332,048 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\msgina.dll
comctl32.dll 5.81 537.77 KB (550,672 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\comctl32.dll
setupapi.dll 5.00.2195.2663 555.77 KB (569,104
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\setupapi.dll
winmm.dll 5.00.2161.1 184.77 KB (189,200 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\winmm.dll
winsta.dll 5.00.2195.2386 36.77 KB (37,648 bytes)
1/7/2003 2:55:06 PM Microsoft Corporation
c:\winnt\system32\winsta.dll
wsock32.dll 5.00.2195.2871 21.27 KB (21,776
bytes) 1/7/2003 2:55:07 PM Microsoft Corporation
c:\winnt\system32\wsock32.dll
dnsapi.dll 5.00.2195.2785 130.77 KB (133,904 bytes)
1/7/2003 2:54:40 PM Microsoft Corporation
c:\winnt\system32\dnsapi.dll
wldap32.dll 5.00.2195.2797 125.27 KB (128,272
bytes) 1/7/2003 2:55:06 PM Microsoft Corporation
c:\winnt\system32\wldap32.dll
ws2help.dll 5.00.2134.1 17.77 KB (18,192
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\ws2help.dll
ws2_32.dll 5.00.2195.2780 67.77 KB (69,392 bytes)
1/7/2003 2:55:06 PM Microsoft Corporation
c:\winnt\system32\ws2_32.dll
samlib.dll 5.00.2195.2780 49.77 KB (50,960 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\samlib.dll
netrap.dll 5.00.2134.1 11.27 KB (11,536 bytes)
12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\netrap.dll
netapi32.dll 5.00.2195.2808 303.77 KB (311,056
bytes) 1/7/2003 2:54:54 PM Microsoft Corporation
c:\winnt\system32\netapi32.dll
profmap.dll 5.00.2181.1 29.27 KB (29,968
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\profmap.dll
secur32.dll 5.00.2195.2862 46.77 KB (47,888
bytes) 1/7/2003 2:55:01 PM Microsoft Corporation
c:\winnt\system32\secur32.dll
sfc.dll 5.00.2195.2896 92.11 KB (94,320 bytes)
1/7/2003 2:55:01 PM Microsoft Corporation
c:\winnt\system32\sfc.dll
nddeapi.dll 5.00.2137.1 15.27 KB (15,632
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\nddeapi.dll
userenv.dll 5.00.2195.2780 361.77 KB (370,448
bytes) 12/8/1999 5:00:00 AM Microsoft Corporation
c:\winnt\system32\userenv.dll
user32.dll 5.00.2195.2821 392.77 KB (402,192 bytes)

```
            12/8/1999 5:00:00 AM Microsoft Corporation
            c:\winnt\system32\user32.dll
gdi32.dll   5.00.2195.2778       228.77 KB (234,256 bytes)
            12/8/1999 5:00:00 AM Microsoft Corporation
            c:\winnt\system32\gdi32.dll
rpcrt4.dll  5.00.2195.2832       437.27 KB (447,760 bytes)
            1/7/2003 2:55:00 PM  Microsoft Corporation
            c:\winnt\system32\rpcrt4.dll
advapi32.dll        5.00.2195.2867       351.77 KB (360,208
bytes)      12/8/1999 5:00:00 AM Microsoft Corporation
            c:\winnt\system32\advapi32.dll
kernel32.dll        5.00.2195.2778       714.77 KB (731,920
bytes)      12/8/1999 5:00:00 AM Microsoft Corporation
            c:\winnt\system32\kernel32.dll
msvcrt.dll  6.10.8924.0          284.05 KB (290,869 bytes)
            5/4/2001 12:05:02 PM Microsoft Corporation
            c:\winnt\system32\msvcrt.dll
winlogon.exe        5.00.2195.2953       173.77 KB (177,936
bytes)      12/8/1999 5:00:00 AM Microsoft Corporation
            c:\winnt\system32\winlogon.exe
sfcfiles.dll 5.00.2195.2967      948.27 KB (971,024 bytes)
            1/7/2003 2:55:01 PM  Microsoft Corporation
            c:\winnt\system32\sfcfiles.dll
ntdll.dll   5.00.2195.2779       478.77 KB (490,256 bytes)
            5/4/2001 12:05:02 PM Microsoft Corporation
            c:\winnt\system32\ntdll.dll
smss.exe    5.00.2195.2901       44.27 KB (45,328 bytes)
            12/8/1999 5:00:00 AM Microsoft Corporation
            c:\winnt\system32\smss.exe

[Services]

Display Name            Name        State       Start Mode
            Service Type            Path        Error Control
            Start Name              Tag ID
Alerter     Alerter     Running     Auto        Share Process
            c:\winnt\system32\services.exe      Normal
            LocalSystem             0
Application Management              AppMgmt     Running     Manual
            Share Process           c:\winnt\system32\services.exe
            Normal      LocalSystem             0
Computer Browser        Browser     Running     Auto        Share
Process     c:\winnt\system32\services.exe      Normal
            LocalSystem             0
Indexing Service        cisvc       Stopped     Manual      Share
Process     c:\winnt\system32\cisvc.exe         Normal
            LocalSystem             0
ClipBook    ClipSrv     Stopped     Manual      Own Process
            c:\winnt\system32\clipsrv.exe       Normal
            LocalSystem             0
Distributed File System             Dfs         Running     Auto
            Own Process             c:\winnt\system32\dfssvc.exe
            Normal      LocalSystem             0
DHCP Client             Dhcp        Running     Auto        Share
Process     c:\winnt\system32\services.exe      Normal
            LocalSystem             0
Logical Disk Manager Administrative Service                 dmadmin
            Stopped     Manual      Share Process
            c:\winnt\system32\dmadmin.exe /com   Normal
            LocalSystem             0
Logical Disk Manager                dmserver    Running     Auto
            Share Process           c:\winnt\system32\services.exe
            Normal      LocalSystem             0
DNS Client              Dnscache    Running     Auto        Share
Process     c:\winnt\system32\services.exe      Normal
            LocalSystem             0
Event Log   Eventlog     Running     Auto        Share Process
            c:\winnt\system32\services.exe      Normal
            LocalSystem             0
COM+ Event System       EventSystem             Running     Manual
            Share Process           c:\winnt\system32\svchost.exe -
k netsvcs   Normal      LocalSystem             0
Fax Service             Fax         Stopped     Manual      Own
Process     c:\winnt\system32\faxsvc.exe        Normal
            LocalSystem             0
IIS Admin Service       IISADMIN    Running     Auto        Share
Process     c:\winnt\system32\inetsrv\inetinfo.exe          Normal
            LocalSystem             0
Intersite Messaging     IsmServ     Stopped     Disabled    Own
Process     c:\winnt\system32\ismserv.exe       Normal
            LocalSystem             0
Kerberos Key Distribution Center                kdc         Stopped
            Disabled    Share Process
            c:\winnt\system32\lsass.exe         Normal
            LocalSystem             0
Server      lanmanserver            Running     Auto        Share
Process     c:\winnt\system32\services.exe      Normal
            LocalSystem             0
Workstation             lanmanworkstation       Running     Auto
            Share Process           c:\winnt\system32\services.exe
            Normal      LocalSystem             0
License Logging Service             LicenseService          Running
            Auto        Own Process
            c:\winnt\system32\llssrv.exe        Normal
            LocalSystem             0
TCP/IP NetBIOS Helper Service       LmHosts     Running     Auto
            Share Process           c:\winnt\system32\services.exe
            Normal      LocalSystem             0
Messenger               Messenger               Running     Auto
            Share Process           c:\winnt\system32\services.exe
            Normal      LocalSystem             0
NetMeeting Remote Desktop Sharing   mnmsrvc     Stopped
            Manual      Own Process
            c:\winnt\system32\mnmsrvc.exe       Normal
            LocalSystem             0
Distributed Transaction Coordinator MSDTC       Running
            Auto        Own Process
            c:\winnt\system32\msdtc.exe         Normal
            LocalSystem             0
Windows Installer       MSIServer   Stopped     Manual      Share
Process     c:\winnt\system32\msiexec.exe /v    Normal
            LocalSystem             0
Network DDE             NetDDE      Stopped     Manual      Share
Process     c:\winnt\system32\netdde.exe        Normal
            LocalSystem             0
Network DDE DSDM        NetDDEdsdm              Stopped     Manual
            Share Process           c:\winnt\system32\netdde.exe
            Normal      LocalSystem             0
Net Logon   Netlogon    Stopped     Manual      Share Process
            c:\winnt\system32\lsass.exe         Normal
            LocalSystem             0
Network Connections     Netman      Running     Manual
            Share Process           c:\winnt\system32\svchost.exe -
k netsvcs   Normal      LocalSystem             0
File Replication        NtFrs       Stopped     Manual      Own
Process     c:\winnt\system32\ntfrs.exe         Ignore
            LocalSystem             0
NT LM Security Support Provider     NtLmSsp     Stopped
            Manual      Share Process
            c:\winnt\system32\lsass.exe         Normal
            LocalSystem             0
Removable Storage       NtmsSvc     Running     Auto        Share
Process     c:\winnt\system32\svchost.exe -k netsvcs    Normal
            LocalSystem             0
Plug and Play           PlugPlay    Running     Auto        Share
Process     c:\winnt\system32\services.exe      Normal
            LocalSystem             0
IPSEC Policy Agent      PolicyAgent             Running     Auto
            Share Process           c:\winnt\system32\lsass.exe
            Normal      LocalSystem             0
Protected Storage       ProtectedStorage        Running     Auto
            Share Process           c:\winnt\system32\services.exe
            Normal      LocalSystem             0
PsShutdown              PsShutdownSvc           Stopped     Manual
            Own Process
            c:\winnt\system32\pssdnsvc.exe                  Normal
            LocalSystem             0
Remote Access Auto Connection Manager RasAuto   Stopped
            Manual      Share Process
            c:\winnt\system32\svchost.exe -k netsvcs    Normal
            LocalSystem             0
Remote Access Connection Manager    RasMan      Stopped
            Manual      Share Process
            c:\winnt\system32\svchost.exe -k netsvcs    Normal
            LocalSystem             0
Routing and Remote Access           RemoteAccess        Stopped
            Disabled    Share Process
            c:\winnt\system32\svchost.exe -k netsvcs    Normal
            LocalSystem             0
Remote Registry Service             RemoteRegistry      Running
            Auto        Own Process
            c:\winnt\system32\regsvc.exe        Normal
            LocalSystem             0
Remote Command Service              RMSYS       Running     Auto
            Own Process             c:\benchcraft\rsys.exe
            Normal      .\Administrator         0
Remote Procedure Call (RPC) Locator RpcLocator
            Stopped     Manual      Own Process
            c:\winnt\system32\locator.exe       Normal
            LocalSystem             0
Remote Procedure Call (RPC)         RpcSs       Running     Auto
            Share Process           c:\winnt\system32\svchost -k
rpcss       Normal      LocalSystem             0
QoS RSVP    RSVP        Running     Manual      Own Process
            c:\winnt\system32\rsvp.exe -s       Normal
            LocalSystem             0
Security Accounts Manager           SamSs       Running     Auto
            Share Process           c:\winnt\system32\lsass.exe
            Normal      LocalSystem             0
Smart Card Helper       SCardDrv    Stopped     Manual      Share
Process     c:\winnt\system32\scardsvr.exe      Ignore
            LocalSystem             0
Smart Card              SCardSvr    Stopped     Manual      Share
```

```
Process  c:\winnt\system32\scardsvr.exe  Ignore
         LocalSystem  0
Task Scheduler  Schedule  Running  Auto  Share
Process  c:\winnt\system32\mstask.exe  Normal
         LocalSystem  0
RunAs Service  seclogon  Running  Auto  Share
Process  c:\winnt\system32\services.exe  Ignore
         LocalSystem  0
System Event Notification  SENS  Running  Auto
         Share Process  c:\winnt\system32\svchost.exe -
k netsvcs  Normal  LocalSystem  0
Internet Connection Sharing  SharedAccess  Stopped
         Manual  Share Process
         c:\winnt\system32\svchost.exe -k netsvcs  Normal
         LocalSystem  0
Simple Mail Transport Protocol (SMTP)  SMTPSVC  Running
         Auto  Share Process
         c:\winnt\system32\inetsrv\inetinfo.exe  Normal
         LocalSystem  0
Print Spooler  Spooler  Running  Auto  Own
Process  c:\winnt\system32\spoolsv.exe  Normal
         LocalSystem  0
Performance Logs and Alerts  SysmonLog  Stopped
         Auto  Own Process
         c:\winnt\system32\smlogsvc.exe  Normal
         LocalSystem  0
Telephony  TapiSrv  Running  Manual  Share Process
         c:\winnt\system32\svchost.exe -k tapisrv  Normal
         LocalSystem  0
Terminal Services  TermService  Stopped  Manual
         Own Process  c:\winnt\system32\termsrv.exe
         Normal  LocalSystem  0
Telnet  TlntSvr  Stopped  Manual  Own Process
         c:\winnt\system32\tlntsvr.exe  Normal
         LocalSystem  0
Distributed Link Tracking Server  TrkSvr  Stopped
         Manual  Share Process
         c:\winnt\system32\services.exe  Normal
         LocalSystem  0
Distributed Link Tracking Client  TrkWks  Running  Auto
         Share Process  c:\winnt\system32\services.exe
         Normal  LocalSystem  0
Uninterruptible Power Supply  UPS  Stopped  Manual
         Own Process  c:\winnt\system32\ups.exe
         Normal  LocalSystem  0
Utility Manager  UtilMan  Stopped  Manual  Own
Process  c:\winnt\system32\utilman.exe  Normal
         LocalSystem  0
Windows Time  W32Time  Stopped  Manual  Share
Process  c:\winnt\system32\services.exe  Normal
         LocalSystem  0
World Wide Web Publishing Service  W3SVC  Running
         Auto  Share Process
         c:\winnt\system32\inetsrv\inetinfo.exe  Normal
         LocalSystem  0
Windows Management Instrumentation  WinMgmt  Running
         Auto  Own Process
         c:\winnt\system32\wbem\winmgmt.exe  Ignore
         LocalSystem  0
Windows Management Instrumentation Driver Extensions
         Wmi  Running  Manual  Share Process
         c:\winnt\system32\services.exe  Normal
         LocalSystem  0

[Program Groups]

Group Name  Name  User Name
Accessories  Default User:Accessories  Default
User
Accessories\Accessibility  Default
User:Accessories\Accessibility  Default User
Accessories\Entertainment  Default
User:Accessories\Entertainment  Default User
Accessories\System Tools  Default
User:Accessories\System Tools  Default User
Startup  Default User:Startup  Default User
Accessories  All Users:Accessories  All Users
Accessories\Accessibility  All
Users:Accessories\AccessibilityAll 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 ServerAll Users:Microsoft SQL Server  All Users
Startup  All Users:Startup  All Users
Terminal Services Client  All Users:Terminal Services
Client  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
Benchcraft  ACL01\Administrator:Benchcraft
         ACL01\Administrator
QLogic Corporation  ACL01\Administrator:QLogic Corporation
         ACL01\Administrator
QLogic Corporation\SANblade Control VIX
         ACL01\Administrator:QLogic Corporation\SANblade
Control VIX  ACL01\Administrator
Startup  ACL01\Administrator:Startup  ACL01\Administrator

[Startup Programs]

Program  Command User Name  Location
Command Prompt  c:\winnt\system32\cmd.exe
         ACL01\Administrator Startup
Microsoft Office  c:\progra~1\micros~4\office\osa9.exe -b -l
         All Users  Common Startup

[OLE Registration]

Object  Local Server
Microsoft Excel Worksheet  Not Available
Microsoft Excel Chart  Not Available
Microsoft Excel Worksheet
         C:\PROGRA~1\MICROS~4\Office\EXCEL.EXE
Microsoft Excel Chart
         C:\PROGRA~1\MICROS~4\Office\EXCEL.EXE
Sound (OLE2)  sndrec32.exe
Media Clip mplay32.exe
Video Clip mplay32.exe /avi
MIDI Sequence  mplay32.exe /mid
         Not Available
Microsoft Excel Chart  Not Available
Microsoft Excel 4.0 Macro  Not Available
Sound  Not Available
Media Clip Not Available
Image Document  "C:\Program Files\Windows
NT\Accessories\ImageVue\KodakImg.exe"
WordPad Document  "%ProgramFiles%\Windows
NT\Accessories\WORDPAD.EXE"
Windows Media Services DRM Storage object  Not
Available
Bitmap Image  mspaint.exe

[Internet Explorer 5]

[ Following are sub-categories of this main category ]

[Summary]

Item  Value
Version  5.00.3315.1000
Build  53315.1000
Product ID 51876-335-9494637-05637
Application Path  C:\Program Files\Internet Explorer
Language English (United States)
Active Printer  Not Available

Cipher Strength  168-bit
Content Advisor  Disabled
IEAK Install  No

[File Versions]

File  Version  Size  Date  Path  Company
advapi32.dll  5.0.2195.2867  352 KB  5/4/2001
12:05:02 PM  C:\WINNT\system32  Microsoft
Corporation
advpack.dll  5.0.3103.1000  87 KB  5/4/2001
12:05:02 PM  C:\WINNT\system32  Microsoft
Corporation
browselc.dll  5.0.3315.2846  35 KB  5/4/2001
12:05:02 PM  C:\WINNT\system32  Microsoft
Corporation
browseui.dll  5.0.3315.2846  789 KB  5/4/2001
12:05:02 PM  C:\WINNT\system32  Microsoft
Corporation
ckcnv.exe 5.0.2189.1 9 KB  12/8/1999 5:00:00 AM
```

C:\WINNT\system32 Microsoft Corporation
comctl32.dll 5.81.3103.1000 538 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
crypt32.dll 5.131.2195.2833 451 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
enhsig.dll <File Missing> Not Available Not Available Not Available Not Available
iemigrat.dll <File Missing> Not Available Not Available Not Available Not Available
iesetup.dll 5.0.3103.1000 57 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
iexplore.exe 5.0.2920.0 59 KB 12/8/1999 5:00:00 AM C:\Program Files\Internet Explorer Microsoft Corporation
imagehlp.dll 5.0.2195.2778 126 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
imghelp.dll <File Missing> Not Available Not Available Not Available Not Available
inseng.dll 5.0.3103.1000 72 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
jobexec.dll 5.0.0.1 47 KB 12/8/1999 5:00:00 AM C:\WINNT\system32 Microsoft Corporation
jscript.dll 5.1.0.5907 476 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
jsproxy.dll 5.0.2920.0 13 KB 12/8/1999 5:00:00 AM C:\WINNT\system32 Microsoft Corporation
msaahtml.dll <File Missing> Not Available Not Available Not Available Not Available
mshtml.dll 5.0.3315.2870 2290 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
msjava.dll 5.0.3802.0 923 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
msoss.dll <File Missing> Not Available Not Available Not Available Not Available
msxml.dll 8.0.5718.1 493 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
occache.dll 5.0.3103.1000 86 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
ole32.dll 5.0.2195.2887 970 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
oleaut32.dll 2.40.4517.0 612 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
olepro32.dll 5.0.4517.0 160 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
rsabase.dll 5.0.2195.2228 128 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
rsaenh.dll 5.0.2195.2228 131 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
rsapi32.dll <File Missing> Not Available Not Available Not Available Not Available
rsasig.dll <File Missing> Not Available Not Available Not Available Not Available
schannel.dll 5.1.2195.0 138 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
shdoc401.dll <File Missing> Not Available Not Available Not Available Not Available
shdocvw.dll 5.0.3315.2879 1078 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
shell32.dll 5.0.3315.2902 2304 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
shlwapi.dll 5.0.3315.1000 283 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
url.dll 5.0.2920.0 82 KB 12/8/1999 5:00:00 AM C:\WINNT\system32 Microsoft Corporation
urlmon.dll 5.0.3315.1000 441 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
vbscript.dll 5.1.0.5907 428 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
webcheck.dll 5.0.3315.1000 252 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
win.com 5.0.2134.1 24 KB 12/8/1999 5:00:00 AM C:\WINNT\system32 Microsoft Corporation
wininet.dll 5.0.3315.1000 457 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
winsock.dll 3.10.0.103 3 KB 12/8/1999 5:00:00 AM C:\WINNT\system32 Microsoft Corporation
wintrust.dll 5.131.2195.2779 162 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
wsock.vxd <File Missing> Not Available Not Available Not Available Not Available
wsock32.dll 5.0.2195.2871 21 KB 5/4/2001 12:05:02 PM C:\WINNT\system32 Microsoft Corporation
wsock32n.dll <File Missing> Not Available Not Available Not Available Not Available

[Connectivity]

Item Value
Connection Preference Never dial
EnableHttp1.1 1
ProxyHttp1.1 0

LAN Settings

AutoConfigProxy wininet.dll
AutoProxyDetectMode Disabled
AutoConfigURL
Proxy Disabled
ProxyServer
ProxyOverride

[Cache]

[Following are sub-categories of this main category]

[Summary]

Item Value
Page Refresh Type Automatic
Temporary Internet Files Folder C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files
Total Disk Space 17335 MB
Available Disk Space 15388 MB
Maximum Cache Size 541 MB
Available Cache Size 542 MB

[List of Objects]

Program File Status CodeBase
No cached object information available

[Content]

[Following are sub-categories of this main category]

[Summary]

Item Value
Content Advisor Disabled

[Personal Certificates]

Issued To Issued By Validity Signature Algorithm
Administrator Administrator 1/7/2003 to 12/14/2102 sha1RSA

[Other People Certificates]

Issued To Issued By Validity Signature Algorithm
No other people certificate information available

[Publishers]

Name
No publisher information available

[Security]

Zone Security Level
Local intranet Medium-low
Trusted sites Low
Internet Medium
Restricted sites High

<Microsoft® SQL Server™ 2000 setting>

Startup Parameters

sqlservr -c -x -T3502 -T3428

-c	Start SQL Server™ 2000 independently of the Microsoft® Windows® NT Service Control Manager.
-x	Disable the keeping of CPU time and cache-hit ration statistics.
-T3502	Prints a message to the log at the beginning and end of each checkpoint.
-T3428	Use half of the memory available on the system during the recovery process to keep a hash table which enables recovery to proceed smoothly.

Prior to the TPC-C run, the following command was executed via osql to turn off automatic checkpoints. Checkpoints were issued manually during the run at 30 minute intervals.

dbcc traceon(-1, 3505)

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. IA-64
--            Copyright Microsoft, 2000,2001
--  Purpose:  Returns SQL Server version string



print " "
select convert(char(30), getdate(),9)
print " "

 ------------------------------
 Apr 20 2003  5:17:46:980PM

(1 row affected)

1> 2> 3>
select @@version




 ------------------------------------------------------------------------------
        ------------------------------------------------------------------------
        ------------------------------------------------------------------------
        ------------------------------------------------------------------------
        ---------
 Microsoft SQL Server  2000 - 8.00.782 (Intel IA-64)
        Apr  4 2003 10:35:34
        C
        opyright (c) 1988-2003 Microsoft Corporation
        Enterprise Edition (64-bi
        t) on Windows NT 5.2 (Build 3718: )



(1 row affected)
1> 2>
1> 2> 3> 4> 5> 6> 7> 8> 9> 10>
--  File:     CONFIG.SQL
--            Microsoft TPC-C Benchmark Kit Ver. IA-64
--            Copyright Microsoft, 2000,2001
--  Purpose:  Collects SQL Server configuration parameters


print " "
select convert(char(30), getdate(),9)
print " "

 ------------------------------
 Apr 20 2003  5:17:47:300PM

(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                       -2147483648  2147483647           -1          -
1
 affinity64 mask                     -2147483648  2147483647            0
0
 allow updates                                0           1            0
0
 awe enabled                                  0           1            0
0
 c2 audit mode                                0           1            0
0
 cost threshold for parallelism               0       32767            5
5
 Cross DB Ownership Chaining                  0           1            0
0
 cursor threshold                            -1  2147483647           -1          -
1
 default full-text language                   0  2147483647         1033
1033
 default language                             0        9999            0
0
 fill factor (%)                              0         100            0
0
 index create memory (KB)                   704  2147483647            0
0
 lightweight pooling                          0           1            1
1
 locks                                     5000  2147483647            0
0
 max degree of parallelism                    0          32            1
1
 max server memory (MB)                       4  2147483647       504000
504000
```

```
 max text repl size (B)                      0  2147483647        65536
65536
 max worker threads                         32       32767          820
820
 media retention                             0         365            0
0
 min memory per query (KB)                 512  2147483647         1024
1024
 min server memory (MB)                      0  2147483647            0
0
 nested triggers                             0           1            1
1
 network packet size (B)                   512       65536         4096
4096
 open objects                                0  2147483647            0
0
 priority boost                              0           1            1
1
 query governor cost limit                   0  2147483647            0
0
 query wait (s)                             -1  2147483647           -1          -
1
 recovery interval (min)                     0       32767          119
119
 remote access                               0           1            1
1
 remote login timeout (s)                    0  2147483647           20
20
 remote proc trans                           0           1            0
0
 remote query timeout (s)                    0  2147483647          600
600
 scan for startup procs                      0           1            0
0
 set working set size                        0           1            0
0
 show advanced options                       0           1            1
1
 two digit year cutoff                    1753        9999         2049
2049
 user connections                            0       32767            0
0
 user options                                0       32767            0
0

1>
```

<Disk Array configuration>

HBA#	Storage S2300#	SP#	DEU#	RAID Level	LUN Capacity	Disk Type (GB) 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	0	1/0	232.82 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	
			1	1/0	232.82 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	
			2	1/0	232.82 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	
			3	1/0	232.82 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	
1	1	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
	2	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
2	3	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
	4	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
3	5	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
	6	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
4	7	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
	8	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
5	9	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
	10	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
6	11	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
	12	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
7	13	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
	14	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
8	15	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
	16	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
9	17	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
	18	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
10	19	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
	20	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
11	21	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
	22	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
12	23	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
	24	0	0	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			1	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36
			2	0	498.92 GB	36	36	36	36	36	36	36	36	36	36	36	36	36	36	36

Appendix D : Space Calculation

60 Day Space

Note : Numbers are in KBytes unless otherwise specified

Warehouses	42000	**tpmC**	514034.72	**tpmC/W**	12.24

Table	**Rows**	**Data**	**Index**	**5% Space**	**8H Space**	**Total Space**
Warehouse	44,000	4,760	408	258		5,426
District	440,000	49,584	672	2,513		52,769
Item	100,000	9,528	568	232		10,328
New-order	396,000,000	7,055,680	18,144		3,520,000	10,593,824
History	1,320,000,000	78,805,976	306,608		12,252,548	91,365,132
Orders	1,320,000,000	43,102,048	21,026,296		9,931,866	74,060,210
Customer	1,320,000,000	960,000,008	59,909,960	23,457,929		1,043,367,897
Order-line	13,199,956,988	879,997,136	2,082,696		136,611,966	1,018,691,798
Stock	4,400,000,000	1,408,000,008	2,981,952	32,452,585		1,443,434,545
Totals		3,377,024,728	86,327,304	55,913,518	162,316,380	3,681,581,930

DB File Group	**Count**	**Size**	**Needed**	**Overhead**	**Not Needed**
MSSQL_misc_fg	72	1,474,560,000	1,206,727,283	12,067,273	255,765,445
MSSQL_cs_fg	72	2,580,480,000	2,511,670,467	25,116,705	43,692,829
Totals		4,055,040,000	3,718,397,749	37,183,977	299,458,273

Dynamic space	974,853,721	Sum of Data for Order, Order-Line and History (excluding free extents)
Static space	2,581,595,807	Data + Index + 5% Space + Overhead - Dynamic space
Free space	199,132,200	Total Seg. Size - Dynamic Space - Static Space - Not Needed
Daily growth	182,221,332	(Dynamic space/W * 62.5)* tpmC
Daily spread	-74,199,799	Free space - 1.5 * Daily growth (zero if negative)
60 day (KB)	13,514,875,753	Static space + 60 (daily growth + daily spread)
60 day (GB)	12888.79	Excludes OS, Paging and RDBMS Logs
Log size (MB)	800000.00	Total size of log file
% Log used	60.314488	% of log file used during entire run
Total N-O Txn	105916338	Total count of N-O transactions during entire run
Log per N-O txn	4.66	Number of Kbytes per New-Order transaction
8 Hour Log (GB)	1097.70	need double for mirroring
os, file sys, swap	36.00	

	Disk size (GB)	**Priced Qty**	**Priced (GB)**	**Needed(GB)**	**Extra (GB)**
Database, Sys	33.259	1080	35919.72	12,924.79	23,028.34
	33.410	1	33.41		
Mirrored Log	33.259	70	2328.13	2,195.39	132.74

Appendix E : Price Quotation

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
http://www.microsoft.com/

Microsoft

April 15, 2003

NEC Corporation
Keiichi Yamada
1-10 Nisshin-cho,
Fuchu-shi
Tokyo, 1838501

Yamada-san:

Here is the information you requested regarding pricing for several Microsoft products to be used in conjunction with your TPC-C benchmark testing.

All pricing shown is in US Dollars ($).

Part Number	Description	Unit Price	Quantity	Price
810-00560	**SQL Server 2000 Enterprise Edition (64-bit)** *Per processor licensing* *Discount Schedule: Open Program Level C* *Unit Price reflects a 17% discount from the retail unit price of $19,999.*	$16,541	32	$529,312
C11-00821	**Windows 2000 Server** *Server license only - No CALs* *Discount Schedule: Open Program - No Level* *Unit Price reflects a 8% discount from the retail unit price of $799.*	$738	32	$23,616
254-00170	**Visual C++ Standard** *No discounts applied*	$109	1	$109
PRO-PRORS-16U-01	**Database Server Support Package** *1 Year Term*	$1,950	3	$5,850

Some products may not be currently orderable but will be available through Microsoft's normal distribution channels by April 2, 2003.

This quote is valid for the next 90 days.

If we can be of any further assistance, please contact Jamie Reding at (425) 703-0510 or jamiere@microsoft.com.

Reference ID: PCkeya0315045425
Please include this Reference ID in any correspondence regarding this price quote.

CDW The Right Technology. Right Away.™ ASK

Home | About CDW | Customer Support | View Cart |

SmartSearch™ GO

Brands Hardware Software Networking Accessories Telephony Services Training 800 &

RESOURCES
- Order Status
- My Company
- My Account
- Account Team
- New Accounts
- Rebates
- Special Events
- CDW Outlet
- Technical Support
- E-Newsletters
- Solutions Library

ONLINE HELP
PRINTABLE VERSION

SHOPPING CART

· Your Saved Carts · Save This Cart
· Edit Saved Carts · Send To An Associate

YOUR SHOPPING CART :

Continue to Checkout >>

Quantity	Product	CDW	Usually Ships	Price	Ext. Price
103	AESP CAT5 RJ-45M to RJ-45M Molded 25' Patch cable gray	126706	Same Day	$11.00	$1,133.00
3	HITACHI 36.9GB HD 10KRPM 3.5	399867	4-6 Days	$239.03	$717.09
7	Intel PRO/100 S Server Adapter, 5-pack	250852	Same Day	$286.03	$2,002.21
33	NEC AccuSync 50	192126	Same Day	$119.69	$3,949.77
15	QLogic SANblade 2340 2Gb 133MHz PCI-X Single Channel HBA	408081	Same Day	$1,329.32	$19,939.80

Click ⊖ to remove an item from your cart — Sub-Total $27,741.87

Update

Continue to Checkout >>

QuickCart: ⊕ Enter a **CDW part number** to quickly add it to your cart.
* **Sample:** CDW Part #

Usually Ships: Same Day
CDW Part: XXXXXX
Mfg. Part: XXXXXX-XXXXXX
UNSPEC: XXXXXXXX

Shipping Calc: Enter a postal code to quickly estimate shipping cost.

< Continue Shopping

http://www.cdw.com/shop/checkout/cart.asp 2003/04/20

QLogic Corporation • 26600 Laguna Hills Drive • Aliso Viejo • CA 92656 • Ph: (949) 389-6000

To:	Keiichi Yamada NEC Corp. Express Server Performance Lab	**Fax:**	
From:	Gregory Munck	**Date:**	April 15, 2003
Re:	MSRP For QLogic HBA, SFP & Switch QLA2350 (Single Port HBA) SFP (8-Shortwave LC Optic 2/1) Sanbox2-16 port Switch	**Pages incl cover:**	1
cc:			

Yamada-san,

QLogic is pleased to provide the following pricing for your TPC Benchmark publication:

Product	**MSRP**
QLA2350	**$2095**
SFP	**$2150**
SANBOX2-16	**$17,995**

If you have any questions or concerns please do not hesitate to let me know.

Greg Munck
Asia Pacific Regional Sales Manager
Greg.munck@qlogic.com
(949) 389-6420

FORM: AP279 Rev. B (11/01)