TPC Benchmark™ C
Full Disclosure Report

BULL Novascale 5080 (8 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
June 30, 2004

Special Notices

BULL, the Sponsor of this benchmark test, believes that then 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 the report were obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly. BULL does not warrant or represent that a user can or will achieve similar performance expressed an transactions per minute (tpmC) or normalized price/performance ($ USD per pmC). No warranty of system performance or price/performance is expressed or implied in this report.

The following terms used in this publication are trademark of the Bull® company in the United States and/or other countries:

BULL
Novascale
Nova5080

The following terms used in this publication are trademark of the Microsoft® Corporation in the United States and/or other countries:

Windows® Server 2003
SQL Server 2000

The following terms used in this publication are trademark of the Intel® Corporation in the United States and/or other countries:

Intel®
Itanium®
Xeon™

The following terms used in this publication are trademark of the Transaction Processing Performance Council in the United States and/or other countries:

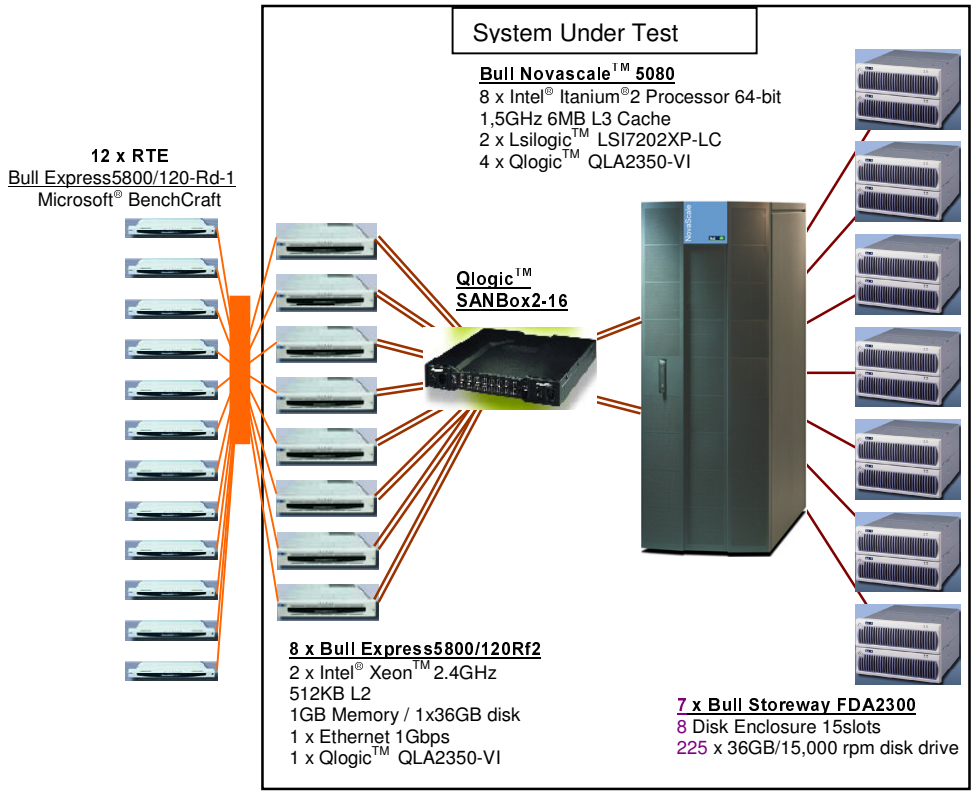
TPC Benchmark
TPC-C and tpmC

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

Bull	**BULL Novascale 5080 (8 SMP)** C/S with Express5800/120Rf-2	TPC Benchmark C Standard Specification version 5.3 Report Date: June 30, 2004

Total System Cost	TPC-C Throughput	Price / Performance	Availability Date
$ 793,606 USD	**175,366.24**	**$ 4.53 USD / tpmC**	**June 2004**

Processors	Database Manager	OS	Other Software	Number of Users
Server SMP : 8 * { Itanium2 64 bits 1,5GHz 512KB/L2 cache 6MB/L3 cache } 8 Clients : 2 * Xéon 2,4 GHz	Microsoft SQL Server 2000, Enterprise Edition (64-bit)	Microsoft Windows 2003 server, Datacenter Edition (64bit)	Windows 2000 Server Microsoft Visual C++ Microsoft COM+	**140 000**



System Component	Server		Each Client	
Processors Cache	8	Intel® Itanium® 2 Processor 64-bit 1,5GHz 6MB L3 Cache	2	Intel® Xeon™ 2.4GHz 512KB L2
Memory (in GB)	128	64 x 2GB	1	1GB
Disk Controllers	2	Lsilogic LSI7202XP-LC	1	36GB Ultra320 SCSI 15Krpm
Disk Drives	225	36GB/15Krpm	1	On-board SCSI
Total GB of Storage	7,48 TB	225 x 33,6GB (formated disk)	1	36 GB
Others	4	Qlogic QLA2350-VI 2Gbps	1 1	Qlogic QLA2350-VI 2Gbps On-board Ethernet controller 1Gbps

Bull	**Bull Novascale 5080** (8 SMP/128GB) C/S with Express 5800/120Rf-2				**TPC-C REV 5.3 EXECUTIVE SUMMARY** Report Date: 25/6/04		
Description	**Part Number**	**Third Party**		**Unit Price**	**Qty**	**Extended Price**	**3 yr. Maint. Price**
Server Hardware		**Brand Pricing**		**Price**			
Novascale 5080, 8 x Itanium2, 1.5Mhz/6MB, 128GB	CPXU303E-3000	Bull	1	311 777	1	311 777	33 376
HBA PCI-X to FC QLA2350 (+2 spares)	QLA2350	Qlogic	2	*890*	6	*5 340*	
HBA PCI-X to FC LSI7202XP-LC (+2 spares)	LSI7202XP-LC	LSI	3	*1 326*	4	*5 303*	
					Bull subtotal	311 777	33 376
					Third Party subtotal	*10 643*	*0*
					Subtotal	**322 420**	**33 376**
Disk Subsystems							
Expansion Rack 36U/19" (with one PDU)	RCKU204-0000	Bull	1	4 885	2	9 770	
FDA 2300 Fibre RAID 15-slot disks 2xSP 2x512MB	DSSU080-2300	Bull	1	25 025	7	175 175	22 143
1GB Memory Cache for FDA (2x512MB)	CMMU402-0FDA	Bull	1	1 950	14	27 300	
FDA Fibre Disk Enclosure 15-slots w/cables	FDEU010-0FDA	Bull	1	3 564	8	28 512	3 594
36GB Raid 2G Fiber Disk (15Krpm) for FDA	MSUU304-0FDA	Bull	1	698	225	157 039	
10% spare (disks 36GB)	MSUU304-0FDA	Bull	1	698	23	16 053	
FC optical LC to LC 2Gb SAN cable 15m	CBLH008-1500	Bull	1	246	4	984	
FC optical LC to LC 2Gb SAN cable 5m	CBLH008-0500	Bull	1	200	6	1 200	
					Bull subtotal	416 033	25 737
					Third Party subtotal	0	*0*
					Subtotal	**416 033**	**25 737**
Server Software							
Novascale Master	Included	Bull	1				
Preloaded Windows Serv.2003, DTC Ed.(up to 8-CPU)	EXSU263-NA0H	Microsoft	1	14 771	1	14 771	14 000
SQL Server 2000, Enterprise Edition (64-bit)	810-00560	Microsoft	4	*16 541*	8	*132 328*	*5 850*
					Bull subtotal	14 771	14 000
					Third Party subtotal	*132 328*	*5 850*
					Subtotal	**147 099**	**19 850**
Client Hardware					**8**		
Express 5800 / 120Rf-2 Xeon 2.4GHz/512KB	BAS B120RF29902	Bull	1	2 466	8	19 728	
Xeon 2.4GHz/512KB/533MHz Processor Kit	CPUXEON2.499907	Bull	1	698	8	5 587	
1GB ECC (2x512MB) DDR266 SDRAM Memory Kit	MEM1GBMBXX99932	Bull	1	564	8	4 512	
8x/24x low profile internal EIDE DVD-ROM drive	DVDSLIMXXX99902	Bull	1	175	8	1 402	
36GB Ultra320 SCSI HDD (15Krpm, 1")	HDD36GSCSI99961	Bull	1	299	8	2 390	
Natio Pack for 120Rf-2 - French	PKGBNATIOX00229	Included	1	0	8	0	
HBA PCI-X to FC QLA2350	QLA2350	Qlogic	2	*890*	8	*7 120*	
19" 36U Rack without dors with PDU 12 outlets	RCKH003-1000	Bull	1	3 384	1	3 384	
8-Server Monitor Concentrator	AEX-8000-00-BR	Bull	1	1 459	1	1 459	
7' BSU to Rack Mount Monitor Concentrator Monitor	ACN-0009-00-BR	Bull	1	71	8	566	
17" Display	ZCMB1720-H2-00	Bull	1	182	1	182	
Keyboard Tray	DRWH002-0000	Bull	1	258	1	258	
Compact Keyboard (102keys)+Mouse Kit for Rack	AIP-0020-00-BR	Bull	1	121	1	121	
12' Keyboard/Mouse/Monitor Extension Cable Set	ACN-0011-00BR	Bull	1	161	1	161	
3 years of warranty service to 4-h response, 24x7	BMHH024-S3C4	Bull	1	1 350	8		10 801
					Bull subtotal	39 751	10 801
					Third Party subtotal	*7 120*	*0*
					Subtotal	**46 871**	**10 801**
Client Software							
Preloaded Windows Server 2000, Licence 5 users	BUT-S210-IN-FF	Microsoft	1	984	8	7 872	included
Visual C++ Standard	254-00170	Microsoft	4	*109*	1	*109*	included
					Bull subtotal	7 872	0
					Third Party subtotal	*109*	*0*
					Subtotal	**7 981**	**0**
Network Components							
FC Fabric Switch SANbox 2-16 ports 2Gb	SANbox2-16	Qlogic	3	*11 029*	1	*11 029*	2 144
SFP-Shortwave LC Optic / 8 SFPs per pack (10% spare inclued)		Qlogic	3	*1 106*	2	*2 212*	0
FC optical LC to LC 2Gb SAN cable 5m	CBLH008-0500	Bull	1	200	8	1 600	
FC optical LC to LC 2Gb SAN cable 15m (+3 spares)	CBLH008-1500	Bull	1	246	7	1 722	
					Bull subtotal	3 322	2 144
					Third Party subtotal	*13 241*	
					Subtotal	**16 563**	**2 144**
					Total Bull	793 526	86 058
					Total Third Party	*163 441*	*5 850*
					TOTAL	**956 968**	**91 908**
Bull brand Large Purchase Cash Prepay Discount for similar quantities and configurations.						-238 058	-17 212
Notes: Pricing: 1-Bull, 2-Compuview, 3-Provantage, 4-Microsoft					**3-Year Cost of Ownership: $ USD**		**793 606**
					tpmC Rating:		**175366**
Result and methodology audited by François Raab of InfoSizing Inc. (www.sizing.com)					**$ USD /tpmC**		**4.53**

Numerical Quantities Summary

MQTh, Computed Maximum Qualified Throughput			tpmC =	175 366,24		
Response Times (in seconds)	**Average**		**Max**		**90th %-tile**	
New Order	0,46		23,05		0,96	
Payment	0,42		21,87		0,91	
Delivery	0,11		1,43		0,11	
Stock Level	0,63		17,98		1,23	
Order Status	0,45		16,84		0,95	
Delivery (Def)	0,14		2,38		0,21	
Menu	0,00		1,34		0,01	
Response time delay added for emulated components					0,1	
Transaction Mix, in percent of total transaction						
New Order					44,88%	
Payment					43,04%	
Delivery					4,03%	
Stock Level					4,03%	
Order Status					4,03%	
Keying/Think Times (in seconds)	**Min**		**Average**		**Max**	
New Order	18,01	0,00	18,02	12,07	18,16	120,83
Payment	3,01	0,00	3,02	12,06	3,16	120,80
Delivery	2,01	0,00	2,02	5,08	2,16	50,72
Stock Level	2,01	0,00	2,02	5,08	2,16	50,70
Order Status	2,01	0,00	2,02	10,04	2,16	100,70
Test duration						
Ramp-up time					43	minutes
Measurement interval					120	minutes
Number of checkpoints					4	
Checkpoint interval					30	
Number of transactions (all types) completed in measurement interval					48 781 795	

Table of content

List of Figures

Abstract

This report documents the full disclosure information required by the TPC Benchmark™ C Standard Specification Revision 5.3 dated December 2003 for the measurements on BULL Novascale 5080 (8 SMP). 8Clients (BULL Express5800/120Rf-2) were used as the front-end clients

The operating system and the DBMS used on the server were Microsoft Windows 2003 server, Datacenter Edition (64-bit, and Microsoft SQL Server 2000, Enterprise Edition (64-bit).
The operating system on the clients was Microsoft Windows 2000 server, Enterprise Edition (32-bit). Those clients ran Microsoft IIS server 5.0 and COM+

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

TPC Benchmark™ C Metrics

The standard TPC BenchmarkTM C metrics, tpcmC (transactions per minute), price per tpmC (three year capital cost per measured tpmC) are reported.

System	Softwares	Total System Cost	tpmC	$ USD per tpmC	Availability Date
BULL Novascale 5080 (8 SMP)	Microsoft Windows 2003 server, Datacenter Edition (64-bit) (build 3790)-SP1 version 1159 Microsoft SQL Server 2000, Enterprise Edition (64-bit) (build 883)	**$ 793,606 USD**	**175,366.24**	**4.53**	**June 2004**

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 François RAAB of Info Sizing Inc., to verify compliance with the relevant TPC specifications.

Preface

The TPC Benchmark™ C specification was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmark and to disseminate objective, verifiable performance data to the industry. This full disclosure report is based on the TPC Benchmark™ C Standard specifications Version 5.3 dated April 24, 2004.

TPC Benchmark™ Overview

The TPC describes this benchmark in Clause 0.1 of the specifiaction 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 TPC-C results must include the tpmC rate, the associated price-per-tpmC, and the availability date of the priced configuration.

Although these specifications express implementation in terms of a relational data model with conventional locking scheme, the database may be implemented using any commercially available database management system (DBMS), database server, file system, or other data repository that provides a functionally equivalent implementation. The terms "table", "row", and "column" are used in this document only as examples of logical data structures.

TPC-C uses terminology and metrics that are similar to other benchmarks, originated by the TPC or others. Such similarity in terminology does not in any way imply that TPC-C results are comparable to other benchmarks. The only benchmark results comparable to TPC-C are other TPC-C results conformant with the same revision.

Despite the fact that this benchmark offers a rich environment that emulates many OLTP applications, this benchmark does not reflect the entire range of OLTP requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

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

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 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 version 5.3.

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

1 General Items

1.1 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 TPPC-C standard specification.

1.2 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.

1.3 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 ate the beginning of this report.

1.4 Application Program

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

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

1.5 Sponsor

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

This benchmark test was sponsored by BULL S.A.

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.

1.6 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:

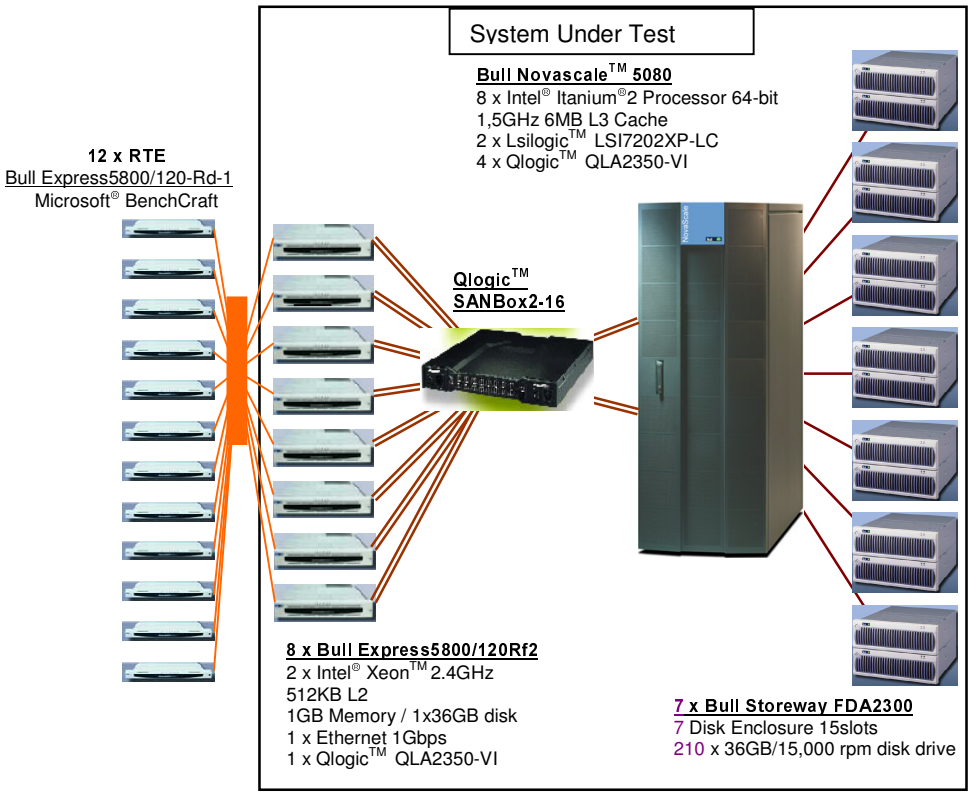
- *Number and type of processors.*
- *Size of allocated memory, and any specific mapping/partitioning of memory unique to the test.*
- *Number and type of disk units (and controllers, if applicable).*
- *Number of channels or bus connections to disk units, including their protocol type.*
- *Number of 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 shows the measured configuration diagram.
Figure 2 shows the priced configuration.

1.7 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: BULL Novascale 5080 (8 SMP) Measured Configuration Diagram

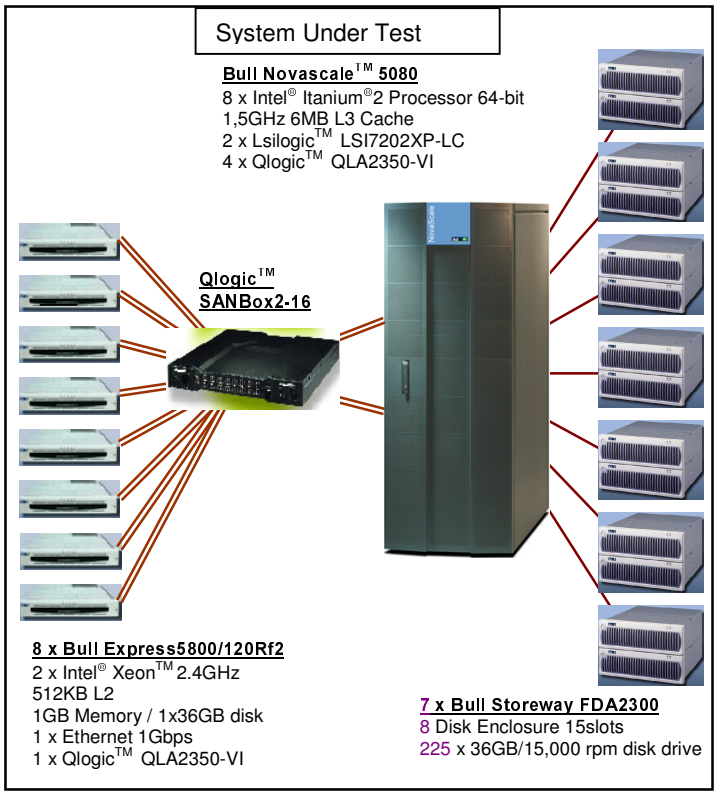


1.8 Priced System Configuration

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

Figure 2: BULL Novascale 5080 (8 SMP), Priced Configuration Diagram

8 Disk Enclosures instead of 7
225 disks instead of 210 (+15 disks for Log).



2 Clause 1: Logical Database Design and Related Items

2.1 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.

2.2 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.

2.3 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.

2.4 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.

2.5 Replication of Tables

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

No tables were replicated in this benchmark test.

2.6 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.

3 Clause 2: Transaction and Terminal profiles Related Items

3.1 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.

3.2 Terminal Input/Output Screen Layout

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

All screen layouts followed the specification exactly.

3.3 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.

3.4 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.

3.5 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.

3.6 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		Values
New Order	Home warehouse order lines	**99,00%**
	Remote warehouse order lines	**1,00%**
	Rolled back transactions	**0,99%**
	Average items per order	**10**
Payment	Home warehouse payments	**85,07%**
	Remote warehouse payments	**14,93%**
	Assessed by last name	**59,95%**
Order Status	Accessed by last name	**59,75%**
Delivery	Skipped deliveries	**0,00**
Transaction Mix	New Order	**44,88%**
	Payment	**43,04%**
	Delivery	**4,03%**
	Stock Level	**4,03%**
	Order Status	**4,03%**

3.7 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.

4 Clause 3: Transaction and System Properties Related Items

4.1 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:

- **A**tomicity
- **C**onsistency
- **I**solation
- **D**urability

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.

4.1.1 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.

4.1.1.1 Atomicity of Completed Transaction

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 has been changed appropriately.

4.1.1.2 Atomicity of Aborted Transaction

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 the 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 a 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 W_YTD, D_YTD, C_BALANCE, C_YTD_PAYMENT and C_PAYMENT_CNT were retrieved again. It was verified that none of the values had changed.

4.1.2 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.

3.3.2.1. Consistency Condition 1
Entries in the WAREHOUSE and DISTRICT tables must satisfy the relationship:
W_YTD = sum(D_YTD)
for each warehouse defined by (W_ID = D_W_ID).

3.3.2.2 Consistency Condition 2
Entries in the DISTRICT, ORDER, and NEW-ORDER tables must satisfy the relationship:
D_NEXT_O_ID - 1 = max(O_ID) = max(NO_O_ID)
for each district defined by (D_W_ID = O_W_ID = NO_W_ID) and (D_ID = O_D_ID = NO_D_ID). This condition does not apply to the NEW-ORDER table for any districts which have no outstanding new orders (i.e., the number of rows is zero).

3.3.2.3 Consistency Condition 3

Entries in the NEW-ORDER table must satisfy the relationship:
max(NO_O_ID) - min(NO_O_ID) + 1 = [number of rows in the NEW-ORDER table for this district]
for each district defined by NO_W_ID and NO_D_ID. This condition does not apply to any districts which have no outstanding new orders (i.e., the number of rows is zero).

3.3.2.4 Consistency Condition 4
Entries in the ORDER and ORDER-LINE tables must satisfy the relationship:
sum(O_OL_CNT) = [number of rows in the ORDER-LINE table for this district]
for each district defined by (O_W_ID = OL_W_ID) and (O_D_ID = OL_D_ID).

4.1.2.1 Test scenario

- *Verify that the database is initially consistent by verifying that it meets the consistency conditions defined in Clauses 3.3.2.1 to 3.3.2.4. Describe the steps used to do this in sufficient detail so that the steps are independently repeatable.*
- *Immediately after performing the verification process, use the standard driving mechanism to submit transactions to the SUT. The transaction rate must be within 10% of the reported tpmC rate and meet all other requirements of a reported measurement interval (see Clause 5.5), including the requirement that the interval contain at least one check-point (see Clause 5.5.2.2). The SUT must be run at this rate for at least 5 minutes.*
- *Stop submitting transactions to the SUT and then repeat the verification steps done for Clause 3.3.3.1. The database must still be consistent after applying transactions. Consistency Condition 4 need only be verified for rows added to the ORDER and ORDER-LINE tables since the previous verification.*

Consistency conditions 1 to 4 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 30 minutes under 127,000 users (12,700 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.

4.1.3 Isolation Tests

Operations of concurrent data base transactions must yield results which are indistinguishable from the results which would be obtained by forcing each transaction to be serially executed to completion in some order.

Isolation tests 1 to 9 were executed using shell scripts t 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.

4.1.4 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 parts of memory (loss of contents).*
- *Power Failure*

4.1.4.1 Loss of Memory, System and Logs.

The test was done by subtraction of a disk from the database Logs, then the system was crashed (loss of power). After the reboot and the database recovery, the durability was guaranteed.

4.1.4.2 Loss of Data

The test was done by the subtraction of a disk from the database data. A backup Log was done. Then the database has been restored from a backup done just before the test, and the recovery of logs was applied. The durability was guaranteed.

5 Clause 4: Scalling and Database Population Related Items

5.1 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 build with 14000 warehouses.

Table 2: Number of Rows for Server

Table	Cardinality (in rows)	Row Length (in bytes)	Table Size (in MB)
Warehouse	14 000	105	1,56
District	140 000	110	16,29
Customer.1	420 000 000	183	98 387,80
Customer.2 txt	420 000 000	500	210 000,00
History	420 000 000	54	26 435,85
Orders	420 000 000	29	14 823,28
New Order	126 000 000	14	3 125,19
Order Line	4 200 000 000	61	292 430,40
Stock	1 400 000 000	315	436 602,05
Item	100 000	94	10,11
Deleted Warehouse Rows	0		

5.2 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) LOADER_NURAND_C	123
C_LAST (Run) CLIENT_NURAND	233

5.3 Distribution of Tables and Logs

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

The following table depicts the distribution of the database over the disks of the tested and priced system.

Table 3: Data Distribution

Novascale 5080 IOB Box		Storeway FDA2300					Storage from the Windows Server point of view						
IOB#-Slot#	HBA#	#	SP#	RAID#	DEU#	# of HDD	Disk Name	Disk Capacity	Partition#1 misc (raw) 37GB	Partition#2 cs (raw) 45 GB	Partition#3 cust(raw) 21GB	Partition#3 backup (NTFS) 250GB	Free
0-10	LSI7202XP	1	0	0	0	15	Disk 6	499	misc1	cs1	cust1	back1	146 GB
			1	0	1	15	Disk 7	499	misc2	cs2	cust2	back2	146 GB
		2	0	0	0	15	Disk 8	499	misc3	cs3	cust3	back3	146 GB
			1	0	1	15	Disk 9	499	misc4	cs4	cust4	back4	146 GB
0-9	LSI7202XP	3	0	0	0	15	Disk 10	499	misc5	cs5	cust5	back5	146 GB
			1	0	1	15	Disk 11	499	misc6	cs6	cust6	back6	146 GB
		4	0	0	0	15	Disk 12	499	misc7	cs7	cust7	back7	146 GB
			1	0	1	15	Disk 13	499	misc8	cs8	cust8	back8	146 GB
0-9	LSI7202XP	5	0	0	0	15	Disk 14	499	misc9	cs9	cust9	back9	146 GB
			1	0	1	15	Disk 15	499	misc10	cs10	cust10	back10	146 GB
		6	0	0	0	15	Disk 16	499	misc11	cs11	cust11	back11	146 GB
			1	0	1	15	Disk 17	499	misc12	cs12	cust12	back12	146 GB
0-1	LSI7202XP	7	0	0	0	15	Disk 0	30	C: OS				
							Disk 2	166	D: Swap1 92GB				
							Disk 3	166	E: Swap2 92GB				
							Disk 4	100	MSSQL\Data 50GB				
							Disk 5	33	Not Used				
			1	10	1	14	Disk 1	233	Log 233GB				
			1	10	2	14	Disk 6	233	Log 233GB				

5.4 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.

5.5 Database Mapping

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

No partitioning or replicated was used.

5.6 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 60-day space calculation is shown in Appendix D.
To calculate the space required to sustain the database log for 8 hours of growth as 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-ORDER transactions was verified from a RTE report covering the entire run.

6. The space used was divided by the number of NEW-ORDER transactions giving a space used per NEW-ORDER transaction.
7. The space used per transaction was multiplied by the measured tpmC rate times 480 minutes (8 hours).

The results of the above steps yielded a requirement of 426.55 GB of logspace (i.e. 853.10 GB of mirrored transaction log volume) to be available to sustain 8 hours. Total space of priced disks for the transaction log volume was 931.28 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.

6 Clause 5: Performance Metrics and Response Time Items

6.1 Throughput

Measured tpmC must be reported.

Table 4: Measured tpmC

175,366.24 tpmC

6.2 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	0,46	23,05	0,96
Payment	0,42	21,87	0,91
Interactive Delivery	0,11	1,43	0,11
Stock Level	0,63	17,98	1,23
Order Status	0,45	16,84	0,95
Deferred Delivery	0,14	2,38	0,21
Menu	0,00	1,34	0,01

6.3 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	18,01	18,16
Payment	3,02	3,01	3,16
Delivery	2,02	2,01	2,16
Stock Level	2,02	2,01	2,16
Order Status	2,02	2,01	2,16

Table 7: Think Times

Type	Average	Minimum	Maximum
New Order	12,07	0,00	120,83
Payment	12,06	0,00	120,80
Interactive Delivery	5,08	0,00	50,72
Stock Level	5,08	0,00	50,70
Order Status	10,04	0,00	100,70

6.4 Response Time Frequency Distribution Curves

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

Figure 3: New-Order Response Time Distribution

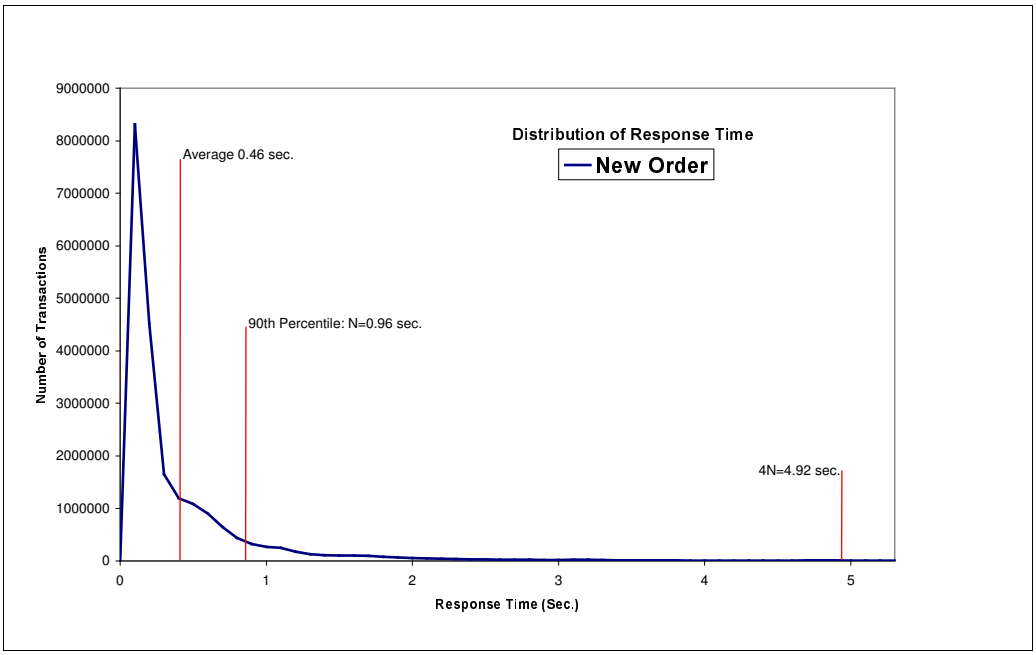


Figure 4: Payment Response Time Distribution

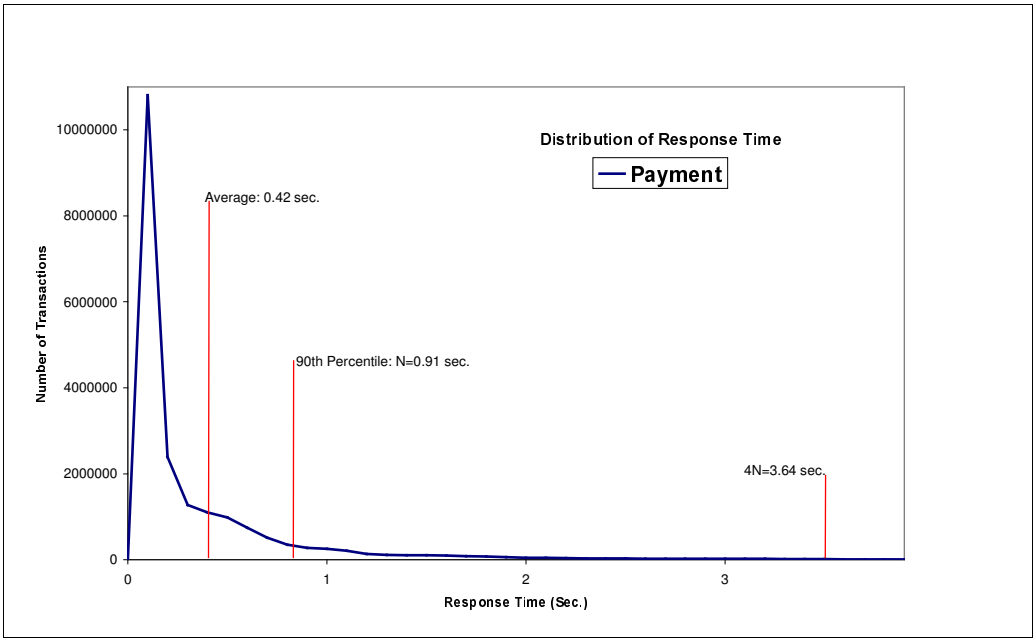


Figure 5: Order-Status Response Time Distribution

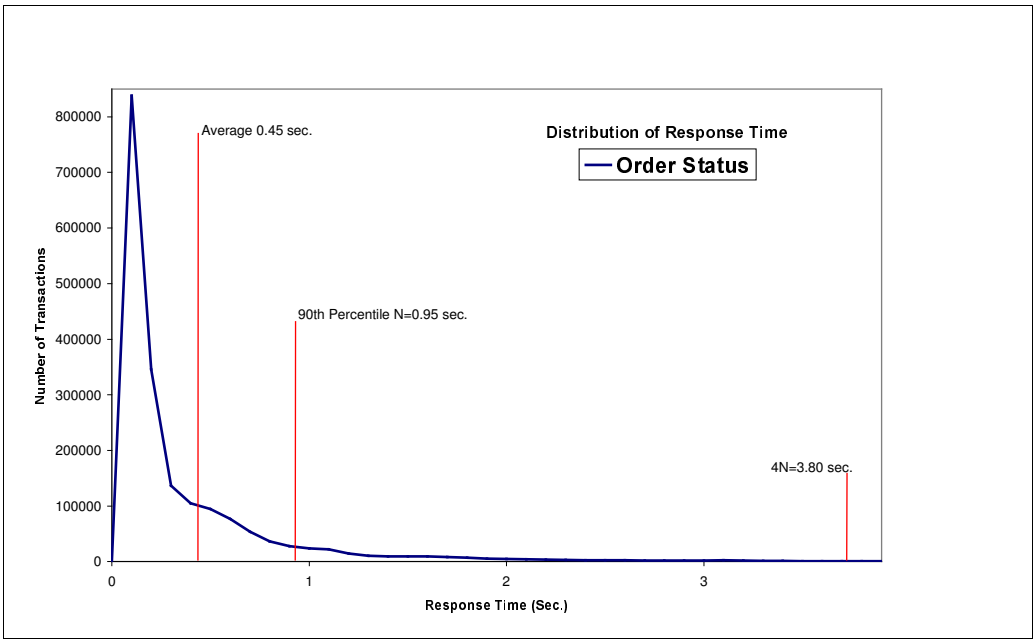


Figure 6: Delivery Response Time Distribution

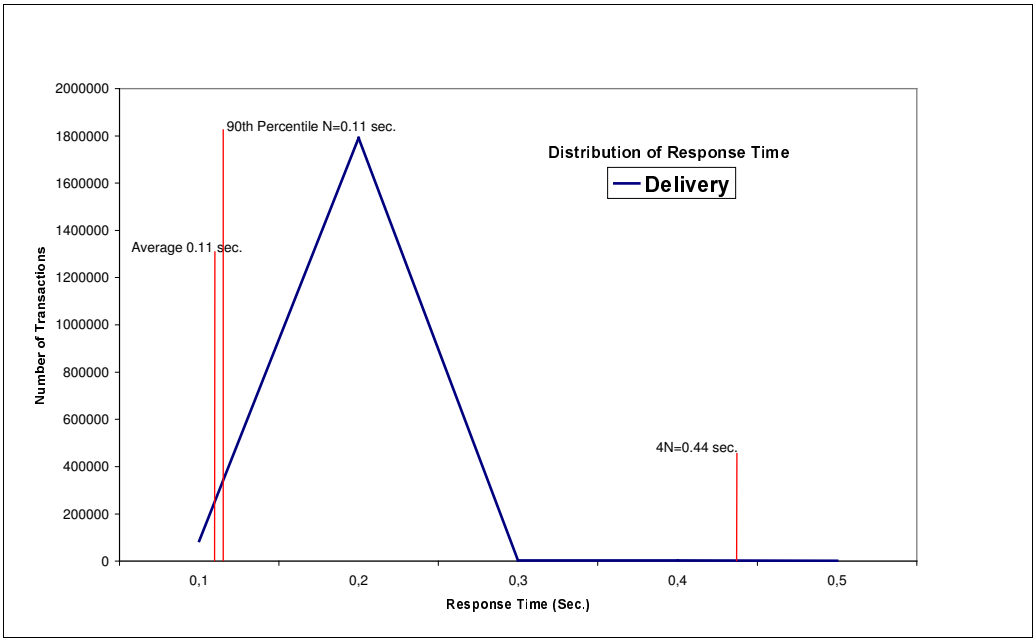
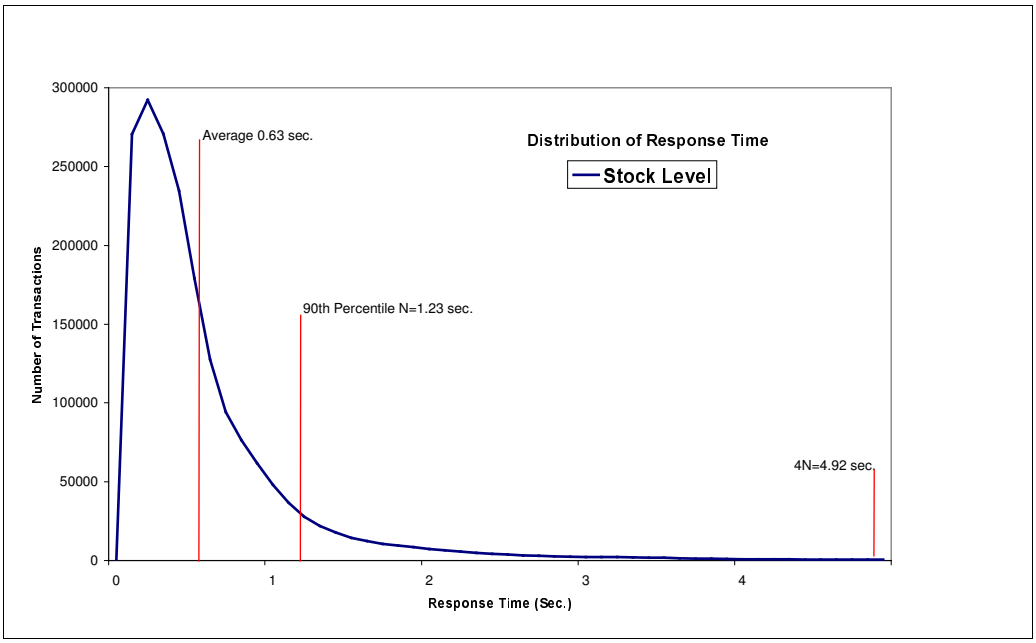


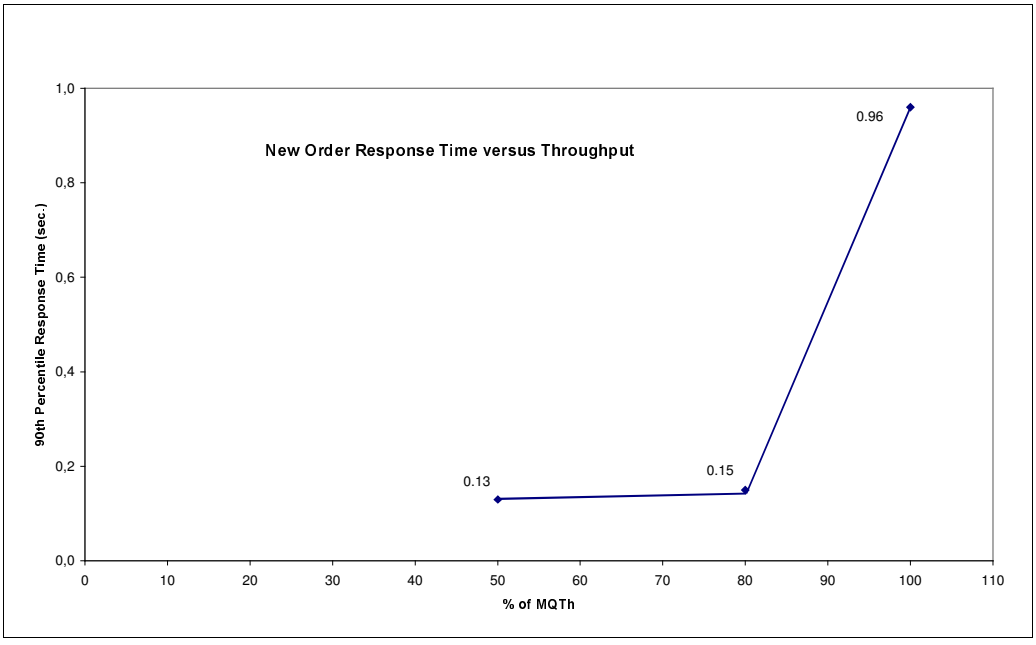
Figure 7: Stock-Level Response Time Distribution



6.5 Response time versus Throughput for the New-Order transaction.

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

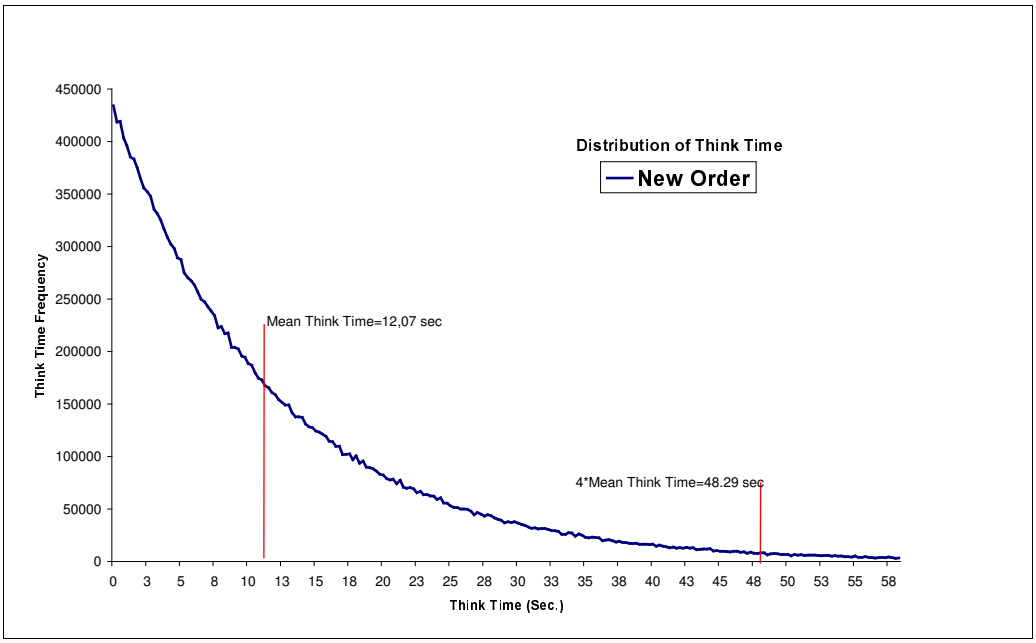
Figure 8: Response Time vs. Throughput for the New-Order transaction.



6.6 New-Order Think Time Frequency Distribution

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

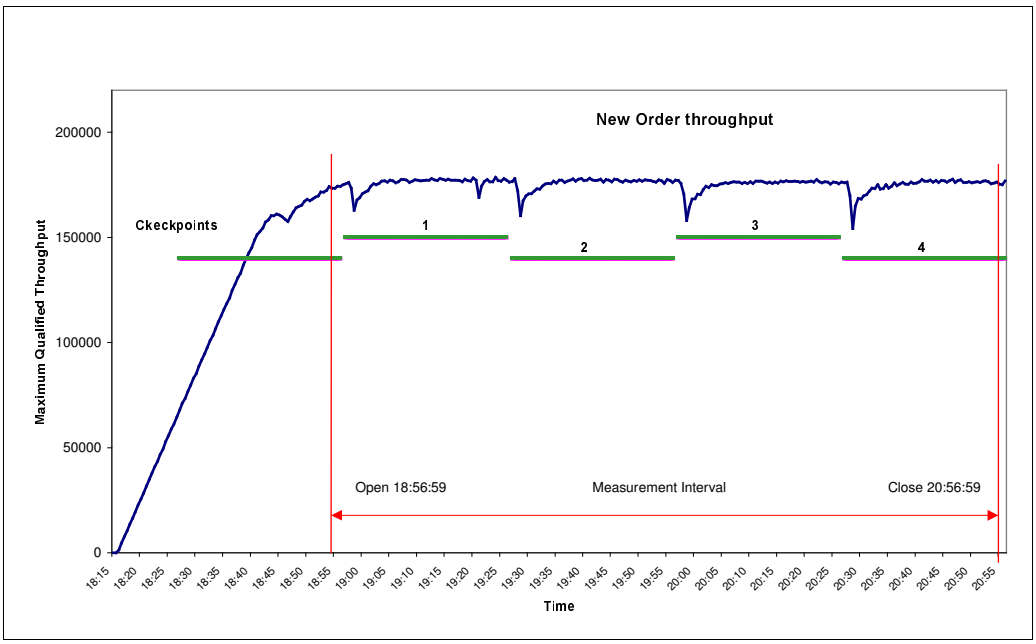
Figure 9: New-Order Think Time



6.7 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 10: New-Order Throughput vs. Elapsed Time



6.8 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 the figure 10.

6.9 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.

6.10 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.

6.11 Regulation of Transaction Mix

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

The RTE was given a weighted random distribution that could not be adjusted during the run.

6.12 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.

6.13 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 were one checkpoint before and four checkpoints during measurement.
The time of the first checkpoint is 10 secondes after the start of the measurement, and the checkpoint interval is 30 minutes.

Extract from SqlServerLog.txt

```
2004-06-24 18:57:08.26 spid707   Ckpt dbid 10 started (0)
2004-06-24 18:57:08.29 spid707   Ckpt dbid 10 phase 1 ended (0)
2004-06-24 19:26:38.28 spid707   Ckpt dbid 10 complete
2004-06-24 19:27:08.17 spid707   Ckpt dbid 10 started (0)
2004-06-24 19:27:08.19 spid707   Ckpt dbid 10 phase 1 ended (0)
2004-06-24 19:56:37.84 spid707   Ckpt dbid 10 complete
2004-06-24 19:57:08.34 spid707   Ckpt dbid 10 started (0)
2004-06-24 19:57:08.38 spid707   Ckpt dbid 10 phase 1 ended (0)
2004-06-24 20:26:38.03 spid707   Ckpt dbid 10 complete
2004-06-24 20:27:08.46 spid707   Ckpt dbid 10 started (0)
2004-06-24 20:27:08.49 spid707   Ckpt dbid 10 phase 1 ended (0)
2004-06-24 20:56:37.55 spid707   Ckpt dbid 10 complete
2004-06-24 20:58:04.30 spid707   Ckpt dbid 10 started (0)
2004-06-24 20:58:04.32 spid707   Ckpt dbid 10 phase 1 ended (0)
```

Table 8: Measurement Period Duration and Checkpoint Duration

	Start	End	Duration	Duration (sec.)
Measurement Period	**18:56:59**	**20:56:59**	**2:00:00**	**7200**
Ramp-up	**18:13:47**	**18:56:59**	**0:43:12**	**2592**
1nd Checkpoint	**18:57:08**	**19:26:38**	**0:29:30**	**1770**
2rd Checkpoint	**19:27:08**	**19:56:37**	**0:29:29**	**1769**
3th Checkpoint	**19:57:08**	**20:26:38**	**0:29:30**	**1770**
4th Checkpoint	**20:27:08**	**20:56:37**	**0:29:29**	**1769**

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

7.1 Description 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.

7.2 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.

7.3 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 (Hyper Text Transfer Protocol) over TCP/IP. **One tenth of a second (100 milliseconds) was added to each transaction time to compensate for the overhead of the Web browser.**

7.4 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 figures 1 and 2 show the tested and priced benchmark configurations.

7.5 Network configuration 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 and 2 in this report have the network configuration of both the tested system and the priced system.

7.6 Network Bandwidth

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

7.6.1 VI network between Data Server and Clients

The Database Server contains **4** VI-NIC adapters (Qlogic™ HDA PCI-X to FC QLA2350). These VI-NIC adapters were connected to 1 FC Fabric Switch (Qlogic™ Sandbox 2-16) with 2Gbps bandwidth. Each Client contains 1 VI-NIC adapters (Qlogic™ HDA PCI-X to FC QLA2350) which was connected to the Fabric Switch.

7.6.2 Ethernet network between RTEs and Clients

Each RTE has **1** Ethernet adapter (1Gbps) to connect to Client systems.
Each Client has **1** Ethernet adapters (1Gbps) to connect to RTE systems.
The network bandwidth between RTE systems and clients systems was 1Gbps.

1 segment was used for the connection of this tested configuration.

7.7 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.

8 Clause 7: Pricing Related Items

8.1 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 detailed list of all hardware and software for the priced configuration is listed in the system pricing summary and page 4 of this document.

8.1.1 Discount

The components supplied by Bull (hardware and software) are discounted by 30% from list price, only for similar quantities and configuration and in case of a prepay cash.

8.1.2 Third-party quotations

All third-party quotations are included in Appendix E at the end of this document.

The aggregate price of all items priced from a web site is lower than 5% of the 3 year Price.

Description	Source	Extended Price
HBA PCI-X to FC Qlogic QLA2350	Compuview Microsystem Inc.	5340+7120
HBA PCI-X to FC LSI LSI7202XP-LC	Provantage	5303
FC Fabric Switch Qlogic SANbox2-16 ports 2Gb	Provantage	11029
SFP-Shortwave LC Optic / 8 SFPs per pack	Provantage	2212
Total		**31004**
5% of $ 793,606 USD		39680

8.1.3 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.

The system has been priced in **$ USD** for the **Europe**.

8.2 Maintenance pricing.

Hardware and software maintenance must be figured at a standard pricing which provides 7 days/week, 24 hours/day coverage, either on-site, or if available as standard offering, via a central support facility. Hardware maintenance maximum response time must not exceed 4 hours, on any part whose replacement is necessary for the resumption of operation.

The three years support pricing for Bull S.A. consists of one year warranty (two years for disks) included in the system package price and two years support price, defined as full care in the “GlobalCare” Bull maintenance offer.

The Silver service offers a high quality of service, permanent prevention and accelerated response time. It is composed of the following services:

Hardware maintenance

- Reception of calls 24hrs 7/7 through a number allocated by Bull or upon the automatic telephone server or being recorded on the web.
- Telephone product assistance from 24 hours/day, 7 days/week
- Remote maintenance and remote diagnostics of the system.

- Access to technical product data via the web.
- On-site intervention: 7 days/week, with 4 hours response time.

Customer spareable and replaceable hardware are priced for:

- HBA PCI-X to FC QLA2350-CK
  Number of QLA2350 used: 4 in the server Novascale 5080, 1 in each Express 5800 / 102Rf-2
  Total: 12, Number of spares: 2

- HBA PCI-X to FC LSI7202XP-LC
  Number of LSI7202XP-LC used: 2 in the server Novascale 5080,
  Total: 2, Number of spares: 2

- SFP-Shortwave LC Optic / 8 SFPs per pack, 16 SFPs in 2 packs.
  Number of SFPs used: 4 between SANbox2-16 and server Novascale 5080, 1 between SANbox and each Express 5800 / 120 Rf-2
  Total: 4 + (1 x 8) = 12, Number of spares: 4

- FC optical LC to LC 2Gb cable
  Number of 5m cable used: 6 + 8 = 14
  Number of 15m cable used: 4 + 4 = 8
  Total: 22 cacles, Number of spares (15m cable): 3

- 36 GB Raid 2G Fiber Disk (15Krpm) for FDA2300
  Number of disks used: 225
  10% spare: 23

Software maintenance
In choosing the Silver level of service, the equivalent level of software support is chosen automatically.
- Access to technical product data via the web.

Warranty extension option.
This option offers the security of receiving a level of service at a set price over three years.
The method of payment is one unique payment at the start of the three-year period, in this case it benefits of 20% discount.

8.3 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 is available from the date of publication of this report: **June 30, 2004.**

8.4 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: **175,366.24** tpmC
- Price per tpmC: **$ 4.53 USD** per tpmC
- Total 3-year cost of ownership: **$ 793,606 USD**

8.5 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:
- **8** Microsoft SQL Server 2000, Enterprise Edition (64-bit) (build 883), Processor License
- **1** Microsoft Windows 2003 server, Datacenter Edition (64-bit) (build 3790)-SP1 version 1159 preloaded on the server, its price is included in the hardware server price.
- **8** Microsoft Windows 2000 server, Enterprise Edition (32-bit) preloaded on clients, the price is included in the price of the client system.
- **1** Visual C++ Standard

9 Clause 8: Audit Related Items

9.1 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 François RAAB of Info Sizing Inc. For the test described in this report.

9.2 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.

Request 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

10 Auditor's letter

Benchmark Sponsor:
Jean-François Lemerre
Bull S.A.
1, rue de Provence
38432 Echirolles
France

June 30, 2004

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

Platform:	**Bull Novascale 5080 (8 SMP) c/s**
Operating system:	**Microsoft Windows Server 2003 Datacenter Edition**
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: Bull Novascale 5080				
8 x Itanium2 (1.5 GHz)	128 GB Main (6 MB L3 per cpu)	225 x 36 GB 15Krpm	0.96 Seconds	**175,366.24**
Clients: Eight (8) Bull ^ Express5800/120Rf2 (Specification for each)				
2 x Xeon (2.4 GHz)	1 GB Main (512 KB L2 per cpu)	1 x 36 GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC's requirements for the benchmark.

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

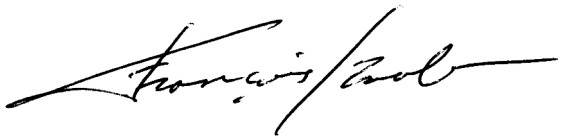
The following verification items were given special attention:

- The database records were the proper size
- The database was properly scaled and populated
- The required ACID properties were met
- The transactions were correctly implemented
- Input data was generated according to the specified percentages
- The transaction cycle times included the required keying and think times
- The reported response times were correctly measured.
- All 90% response times were under the specified maximums
- At least 90% of all delivery transactions met the 80 Second completion time limit
- The reported measurement interval was 120 minutes (7200 seconds)
- The reported measurement interval was representative of steady state conditions
- Four checkpoints were taken during the reported 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: Source Codes

Code used to implement the TPC-C transactions

Kit MSTPCC451\WEBCLNT\common\src\error.h

```
/*                      FILE:                           ERROR.H
 *                                                      Microsoft TPC-C Kit Ver. 4.20.000
 *                                                      Copyright Microsoft, 1999
 *          All Rights Reserved
 *
 *                                                      Version 4.10.000 audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *                      PURPOSE:        Header file for error exception classes.
 *
 *  Change history:
 *                                      4.20.000 - updated rev number to match kit
 *                                      4.21.000 - fixed bug: ~CBaseErr needed to be declared virtual
 */

#pragma once

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

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

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

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


#define ERR_TYPE_LOGIC                                                                                  -1                      //logic error in program; internal error
#define ERR_SUCCESS                                                                                             0                       //success (a non-error error)
#define ERR_BAD_ITEM_ID                                                                                 1                       //expected abort record in txnRecord
#define ERR_TYPE_DELIVERY_POST                                                  2                       //expected delivery post failed
#define                 ERR_TYPE_WEBDLL                                                                 3                       //tpcc web generated error
#define                 ERR_TYPE_SQL                                                                    4                       //sql server generated error
#define                 ERR_TYPE_DBLIB                                                                  5                       //dblib generated error
#define                 ERR_TYPE_ODBC                                                                   6                       //odbc generated error
#define ERR_TYPE_SOCKET                                                                                 7                       //error on communication socket client rte only
#define ERR_TYPE_DEADLOCK                                                                       8                       //dblib and odbc only deadlock condition
#define ERR_TYPE_COM                                                                                    9                       //error from COM call
#define ERR_TYPE_TUXEDO                                                                                 10                      //tuxedo error
#define ERR_TYPE_OS                                                                                             11                      //operating system error
#define ERR_TYPE_MEMORY                                                                                 12                      //memory allocation error
#define ERR_TYPE_TPCC_ODBC                                                                      13                      //error from tpcc odbc txn module
#define ERR_TYPE_TPCC_DBLIB                                                                     14                      //error from tpcc dblib txn module
#define ERR_TYPE_DELISRV                                                                        15                      //delivery server error
#define ERR_TYPE_TXNLOG                                                                                 16                      //txn log error
#define ERR_TYPE_BCCONN                                                                                 17                      //Benchcraft connection class
#define ERR_TYPE_TPCC_CONN                                                                      18                      //Benchcraft connection class
#define ERR_TYPE_ENCINA                                                                                 19                      //Encina error
#define ERR_TYPE_COMPONENT                                                                      20                      //error from COM component
#define ERR_TYPE_RTE                                                                                    21                      //Benchcraft rte
#define ERR_TYPE_AUTOMATION                                                                     22                      //Benchcraft automation errors
#define ERR_TYPE_DRIVER                                                                                 23                      //Driver engine errors
#define ERR_TYPE_RTE_BASE                                                                       24                      //Framework errors
#define                 ERR_BUF_OVERFLOW                                                        25                      //Buffer overflow during receive
#define ERR_TYPE_SOAP_HTTP                                                                      26                      //HTTP/SOAP dll generated error
// TPC-W error types
#define ERR_TYPE_TPCW_CONN                                                                      50                      //Benchcraft connection class
#define ERR_TYPE_TPCW_HTML                                                                      51                      //error from TpcwHtml dll
#define ERR_TYPE_TPCW_USER                                                                      52                      //error from TPC-W user class
#define ERR_TYPE_TPCW_ENG_BASE                                                  53
#define ERR_TYPE_TPCW_ENG_OS                                                    54
#define ERR_TYPE_HTML_RESP                                                                      55
#define ERR_TYPE_TPCW_ODBC                                                                      56
#define ERR_TYPE_SCHANNEL                                                                       57
#define ERR_TYPE_THINK_LIST                                                                     58

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


class CBaseErr
{
public:

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


                                                if (szLoc)
                                                {
                                                                        m_szLoc = new char[strlen(szLoc)+1/*m_szLoc_size*/];
                                                                        strcpy(m_szLoc, szLoc);
                                                }
                                                else
                                                                        m_szLoc                 = NULL;

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



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


                                                if (szLoc)
                                                {
                                                                        m_szLoc = new char[strlen(szLoc)+1/*m_szLoc_size*/];
                                                                        strcpy(m_szLoc, szLoc);
                                                }
                                                else
                                                                        m_szLoc                 = NULL;

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

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

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

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

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

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

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


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


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

        //short         m_errType;
};

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

        CSocketErr(Action eAction, LPCTSTR szLocation = NULL);

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

        Action          m_eAction;
        char            *m_szErrorText;

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

};

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

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

        Action          m_eAction;

private:
        char m_szMsg[ERR_MSG_BUF_SIZE];
};

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

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


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

        int ErrorType() {return ERR_BUF_OVERFLOW;}
        char *ErrorText() {return ERR_INS_BUF_OVERFLOW;}
};
```

Kit MSTPCC451\WEBCLNT\common\src\ReadRegistry.cpp

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

```
*                                                       not yet audited
*
*               PURPOSE:        Implementation for TPC-C Tuxedo class.
*               Contact:        Charles Levine (clevine@microsoft.com)
*
*  Change history:
*                               4.20.000 - first version
*/


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

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

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

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

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

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

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

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

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

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

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

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

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

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

                RegCloseKey(hKey);

                return FALSE;
}
```

Kit MSTPCC451\WEBCLNT\common\src\ReadRegistry.h

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

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

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

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

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );
```

Kit MSTPCC451\WEBCLNT\common\src\trans.h

```
/*              FILE:                           TRANS.H
 *                                                      Microsoft TPC-C Kit Ver. 4.42.000
```

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

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

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

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

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

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

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

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

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

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

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

```
} OL_ORDER_STATUS_DATA;

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

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


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

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

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

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

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

Kit MSTPCC451\WEBCLNT\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;
};
```

Kit MSTPCC451\WEBCLNT\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 );
}
```

Kit MSTPCC451\WEBCLNT\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
```

Kit MSTPCC451\WEBCLNT\common\txnlog\include\txnlog.h

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

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

#pragma once

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

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

  typedef struct _TXN_ORDERSTATUS
        {
                BYTE            CustByName;
        } TXN_ORDERSTATUS;

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


        // Common header for all records in txn log.  The TxnType field is
        // a switch which identifies the particular variant.
        #define TXN_REC_TYPE_CONTROL            1               //
        #define TXN_REC_TYPE_TPCC               2               // replaces TRANSACTION_TYPE_TPCC
        #define TXN_REC_TYPE_TPCC_DELIV_DEF     3
        #define TXN_REC_TYPE_TPCW               4               // replaces TRANSACTION_TYPE_TPCW
```

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


typedef struct _TXN_RECORD_CONTROL
{
        // common header; must exactly match TXN_RECORD_HEADER
        JULIAN_TIME     TxnStartT0;                     // start of txn
        BYTE            TxnType;                        // = TXN_REC_TYPE_CONTROL
        BYTE            TxnSubType;                     // depends on TxnType
        // end of common header

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


// TPC-C Txn Record Layout:
//
//'TxnStartT0' is a Julian timestamp corresponding to the moment the
//txn is sent to the SUT, i.e., beginning of response time.  Deltas
//are in milliseconds.  Note that if RTDelay > 0, then the txn was
//delayed by this amount.  The delay occurs at the beginning of the
//response time.  So if RTDelay > 0, then the txn was actually sent
//at TxnStartT0 + RTDelay.
//
//Graphically:
//
//  time -->
//
//    |--- Menu  ---|-- Keying  --|-- Response --|--- Think  --|
//     <- DeltaT1 -> <- DeltaT2 -> <- DeltaT4  -> <- DeltaT3 ->
//                   ^
//                   ^ TxnStartT0
//
//RTDelay is the amount of response time delay included in DeltaT4.
//RTDelay is recorded per txn because this value can be changed on
//the fly, and so may vary from txn to txn.
//
//TxnStatus is the txn completion code.  It is used to indicate errors.
//For example, in the New Order txn, 1% of txns abort.  TxnStatus will
//reflect this.

typedef struct _TXN_RECORD_TPCC
{
        // common header; must exactly match TXN_RECORD_HEADER
        JULIAN_TIME     TxnStartT0;                     // start of txn
        BYTE            TxnType;                        // = TXN_REC_TYPE_TPCC
        BYTE            TxnSubType;                     // depends on TxnType
        // end of common header

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

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

// TPC-C Deferred Delivery Txn Record Layout:
//
//Incorporating delivery transaction information into the above
        //structure would increase the size of TXN_DETAILS from 8 to 42 bytes.
//Hence, we store delivery transaction details in a separate structure.
//
typedef struct _TXN_RECORD_TPCC_DELIV_DEF
{
        // common header; must exactly match TXN_RECORD_HEADER
        JULIAN_TIME     TxnStartT0;                     // start of txn
        BYTE            TxnType;                        // = TXN_REC_TYPE_TPCC_DELIV_DEF
        BYTE            TxnSubType;                     // = 0
        // end of common header

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

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

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

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

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

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

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

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

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

        //The fields below must exactly match TXN_RECORD_TPCW_USER_DATA
```

```
		unsigned __int32	uiUser;				// user number
		JULIAN_TIME				USMD;			// USMD for this user
		BYTE				bRetCust;		// returning customer?
	} TXN_RECORD_TPCW_USER, *PTXN_RECORD_TPCW_USER;


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

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


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

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

	} TXN_LOG_HEADER, *PTXN_LOG_HEADER;

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



#define		READ_BUFFER_SIZE		64*1024
//#define	WRITE_BUFFER_SIZE		 8*1024
#define		WRITE_BUFFER_SIZE		 128*1024

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

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

#define		TXN_LOG_OS_ERROR	1
#define		TXN_LOG_NOT_SORTED	2

#define SKIP_CTRL_RECS			1

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

                        public:

                                        CTxnLog(LPCTSTR szFileName, DWORD dwOpts, char *szDriver = NULL);
                                        ~CTxnLog(void);

                                        int WriteToLog(PTXN_RECORD_TPCC pTxnRcrd);
                                        int WriteToLog(PTXN_RECORD_TPCC_DELIV_DEF pTxnRcrd);
                                        int WriteToLog(PTXN_RECORD_CONTROL pCtrlRec);
                                        int WriteToLog(PTXN_RECORD_HEADER pCtrlRec);
                                        int WriteToLog(PTXN_RECORD_TPCW pTxnRcrd);          //support for TPC-W

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

                                        void CloseTransactionLogFile(void);

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

                                        int Sort(void);
                                        PTXN_RECORD_HEADER GetSortedRecord();

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

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

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

                                        int ErrorType() {return ERR_TYPE_TXNLOG;};
                                        char *ErrorTypeStr() { return "TXN LOG"; }

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

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

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

Kit MSTPCC451\WEBCLNT\db_dblib_dll\src\tpcc_dblib.cpp

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

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

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

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

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

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

#define                 DEFCLPACKSIZE                           4096
#define MAX_SP_NAME_LEN                 256                     //maximum length of a stored procedure name with parameters

// 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;
}


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

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

```
    sprintf(szMsg, TEXT("Trace in TPCC_DBLIB.DLL: "));
    lpszStrings[0] = szMsg;
    lpszStrings[1] = lpszMsg;

    if (hEventSource != NULL)
                {
        ReportEvent(hEventSource, // handle of event source
            EVENTLOG_INFORMATION_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);
    }
}


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

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

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

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

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

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

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

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

                return 0;
}

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

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

                return;
}

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
 *
 */

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

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

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

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

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


CTPCC_DBLIB::CTPCC_DBLIB (
                LPCSTR szServer,                // name of SQL server
                LPCSTR szUser,                          // user name for login
                LPCSTR szPassword,                      // password for login
                LPCSTR szHost,                          // workstation name; shows up in sp_who; max 30 chars, only first 10 kept by SQL Server
                LPCSTR szDatabase                       // name of database to use
                // LPCWSTR szSPPrefix )          prefix to append to the stored procedure names -> 2 premiers car de Host
                )
{
                LOGINREC        *login;
                const BYTE      *pData;
                char                    szServer2[32];
                // char                 msg[256];
```

```
		char 					*idx;

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

		m_MaxRetries = 10;					// how many retries on deadlock

		// wcsncpy(m_szSPPrefix, szSPPrefix, sizeof(m_szSPPrefix)/sizeof(m_szSPPrefix[0]));

		m_szSPPrefix[0] = 0;
		strcpy(szServer2,szServer);
		idx = strchr(szServer2,':');
		if (idx)
		{
			*idx=0;
			strcpy(m_szSPPrefix,szServer2);
			strcpy(szServer2,idx+1);
		}

		// sprintf(msg,"   szServer:%s, PrefixSP:%s",szServer2,m_szSPPrefix);
		// WriteMessageToEventLog(TEXT(msg));

		// 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);

		// WriteMessageToEventLog(TEXT("Login"));

		m_dbproc = dbopen(login, szServer2);

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

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

		// WriteMessageToEventLog(TEXT("Set user data"));

			// 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;

		char				szName[MAX_SP_NAME_LEN];

		ResetError();

		while (TRUE)
		{
			try
			{
				//dbrpcinit(m_dbproc, "tpcc_stocklevel", 0);
				sprintf(szName, "%stpcc_stocklevel", m_szSPPrefix);
				dbrpcinit(m_dbproc, szName, 0);

				dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.StockLevel.w_id);		// @w_id int
				dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.StockLevel.d_id);		// @d_id tinyint
				dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *) &m_txn.StockLevel.threshold);	// @threshhold smallint

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

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

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

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

				DiscardNextRows(0);
				DiscardNextResults(0);

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

```
                                                throw;
                                        }
                        }               // while (TRUE)

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


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

                        int                                             iTryCount = 0;
                        const BYTE      *pData;

                        char                            szName[MAX_SP_NAME_LEN];

                        ResetError();

                        while (TRUE)
                        {
                                        try
                                        {
                                                        // dbrpcinit(m_dbproc, "tpcc_neworder", 0);
                                                        sprintf(szName, "%stpcc_neworder", m_szSPPrefix);
                                                        dbrpcinit(m_dbproc, szName, 0);

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

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

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

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


                                                        // Get order line results
                                                        m_txn.NewOrder.total_amount = 0;
                                                        for (i = 0; i<m_txn.NewOrder.o_ol_cnt; i++)
                                                        {
                                                                        if (dbresults(m_dbproc) != SUCCEED)
                                                                                        ThrowError(CDBLIBERR::eDbResults);

                                                                        if (dbnumcols(m_dbproc) != 5)
                                                                                        ThrowError(CDBLIBERR::eWrongNumCols);

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

                                                                        if(pData=dbdata(m_dbproc, 1))
                                                                                        UtilStrCpy(m_txn.NewOrder.OL[i].ol_i_name, pData, dbdatlen(m_dbproc, 1));
                                                                        if(pData=dbdata(m_dbproc, 2))
                                                                                        m_txn.NewOrder.OL[i].ol_stock = (*(DBSMALLINT *) pData);
                                                                        if(pData=dbdata(m_dbproc, 3))
                                                                                        UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_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;

                        char                            szName[MAX_SP_NAME_LEN];

                        ResetError();

                        while (TRUE)
                        {
                                try
                                {
                                        // dbrpcinit(m_dbproc, "tpcc_payment", 0);
                                        sprintf(szName, "%stpcc_payment", m_szSPPrefix);
                                        dbrpcinit(m_dbproc, szName, 0);

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

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

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

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

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

                                        if (dbnumcols(m_dbproc) != 27)
                                                ThrowError(CDBLIBERR::eWrongNumCols);

                                        if (pData=dbdata(m_dbproc, 1))
                                                m_txn.Payment.c_id = *((DBINT *) pData);
                                        if (pData=dbdata(m_dbproc, 2))
                                                UtilStrCpy(m_txn.Payment.c_last, pData, dbdatlen(m_dbproc, 2));
                                        if (pData=dbdata(m_dbproc, 3))
                                        {
                                                datetime = *((DBDATETIME *) pData);
                                                dbdatecrack(m_dbproc, &daterec, &datetime);
                                                m_txn.Payment.h_date.year   = daterec.year;
                                                m_txn.Payment.h_date.month  = daterec.month;
                                                m_txn.Payment.h_date.day    = daterec.day;
                                                m_txn.Payment.h_date.hour   = daterec.hour;
                                                m_txn.Payment.h_date.minute = daterec.minute;
                                                m_txn.Payment.h_date.second = daterec.second;
                                        }
                                        if (pData=dbdata(m_dbproc, 4))
                                                UtilStrCpy(m_txn.Payment.w_street_1, pData, dbdatlen(m_dbproc, 4));
                                        if (pData=dbdata(m_dbproc, 5))
                                                UtilStrCpy(m_txn.Payment.w_street_2, pData, dbdatlen(m_dbproc, 5));
                                        if (pData=dbdata(m_dbproc, 6))
                                                UtilStrCpy(m_txn.Payment.w_city, pData, dbdatlen(m_dbproc, 6));
                                        if (pData=dbdata(m_dbproc, 7))
                                                UtilStrCpy(m_txn.Payment.w_state, pData, dbdatlen(m_dbproc, 7));
                                        if (pData=dbdata(m_dbproc, 8))
                                                UtilStrCpy(m_txn.Payment.w_zip, pData, dbdatlen(m_dbproc, 8));
                                        if (pData=dbdata(m_dbproc, 9))
                                                UtilStrCpy(m_txn.Payment.d_street_1, pData, dbdatlen(m_dbproc, 9));
                                        if (pData=dbdata(m_dbproc, 10))
                                                UtilStrCpy(m_txn.Payment.d_street_2, pData, dbdatlen(m_dbproc, 10));
                                        if (pData=dbdata(m_dbproc, 11))
                                                UtilStrCpy(m_txn.Payment.d_city, pData, dbdatlen(m_dbproc, 11));
                                        if (pData=dbdata(m_dbproc, 12))
                                                UtilStrCpy(m_txn.Payment.d_state, pData, dbdatlen(m_dbproc, 12));
                                        if (pData=dbdata(m_dbproc, 13))
                                                UtilStrCpy(m_txn.Payment.d_zip, pData, dbdatlen(m_dbproc, 13));
                                        if (pData=dbdata(m_dbproc, 14))
                                                UtilStrCpy(m_txn.Payment.c_first, pData, dbdatlen(m_dbproc, 14));
                                        if (pData=dbdata(m_dbproc, 15))
                                                UtilStrCpy(m_txn.Payment.c_middle, pData, dbdatlen(m_dbproc, 15));
                                        if (pData=dbdata(m_dbproc, 16))
                                                UtilStrCpy(m_txn.Payment.c_street_1, pData, dbdatlen(m_dbproc, 16));
                                        if (pData=dbdata(m_dbproc, 17))
                                                UtilStrCpy(m_txn.Payment.c_street_2, pData, dbdatlen(m_dbproc, 17));
                                        if (pData=dbdata(m_dbproc, 18))
                                                UtilStrCpy(m_txn.Payment.c_city, pData, dbdatlen(m_dbproc, 18));
                                        if (pData=dbdata(m_dbproc, 19))
                                                UtilStrCpy(m_txn.Payment.c_state, pData, dbdatlen(m_dbproc, 19));
                                        if (pData=dbdata(m_dbproc, 20))
                                                UtilStrCpy(m_txn.Payment.c_zip, pData, dbdatlen(m_dbproc, 20));
                                        if (pData=dbdata(m_dbproc, 21))
                                                UtilStrCpy(m_txn.Payment.c_phone, pData, dbdatlen(m_dbproc, 21));
                                        if (pData=dbdata(m_dbproc, 22))
                                        {
                                                datetime = *((DBDATETIME *) pData);
                                                dbdatecrack(m_dbproc, &daterec, &datetime);
                                                m_txn.Payment.c_since.year   = daterec.year;
                                                m_txn.Payment.c_since.month  = daterec.month;
                                                m_txn.Payment.c_since.day    = daterec.day;
                                                m_txn.Payment.c_since.hour   = daterec.hour;
                                                m_txn.Payment.c_since.minute = daterec.minute;
                                                m_txn.Payment.c_since.second = daterec.second;
                                        }
                                        if(pData=dbdata(m_dbproc, 23))
                                                UtilStrCpy(m_txn.Payment.c_credit, pData, dbdatlen(m_dbproc, 23));
                                        if(pData=dbdata(m_dbproc, 24))
                                                dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,24), SQLFLT8, (BYTE *)&m_txn.Payment.c_credit_lim, 8);
                                        if(pData=dbdata(m_dbproc, 25))
                                                dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,25), SQLFLT8, (BYTE *)&m_txn.Payment.c_discount, 8);
                                        if(pData=dbdata(m_dbproc, 26))
                                                dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,26), SQLFLT8, (BYTE *)&m_txn.Payment.c_balance, 8);
                                        if(pData=dbdata(m_dbproc, 27))
                                                UtilStrCpy(m_txn.Payment.c_data, pData, dbdatlen(m_dbproc, 27));

                                        DiscardNextRows(0);
                                        DiscardNextResults(0);


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

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

```
                }
        }               // while (TRUE)

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


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

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

        char                                    szName[MAX_SP_NAME_LEN];

        ResetError();

        while (TRUE)
        {
                try
                {
                        // dbrpcinit(m_dbproc, "tpcc_orderstatus", 0);
                        sprintf(szName, "%stpcc_orderstatus", m_szSPPrefix);
                        dbrpcinit(m_dbproc, szName, 0);

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

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


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

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

                        if (dbnumcols(m_dbproc) != 5)
                                ThrowError(CDBLIBERR::eWrongNumCols);

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

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


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

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

                        if (dbnumcols(m_dbproc) != 8)
                                ThrowError(CDBLIBERR::eWrongNumCols);

                        if(pData=dbdata(m_dbproc, 1))
                                m_txn.OrderStatus.c_id = (*(DBINT *) pData);
                        if(pData=dbdata(m_dbproc, 2))
                                UtilStrCpy(m_txn.OrderStatus.c_last, pData, dbdatlen(m_dbproc,2));
                        if(pData=dbdata(m_dbproc, 3))
                                UtilStrCpy(m_txn.OrderStatus.c_first, pData, dbdatlen(m_dbproc,3));
                        if(pData=dbdata(m_dbproc, 4))
                                UtilStrCpy(m_txn.OrderStatus.c_middle, pData, dbdatlen(m_dbproc, 4));
                        if(pData=dbdata(m_dbproc, 5))
                        {
                                datetime = *((DBDATETIME *) pData);
                                dbdatecrack(m_dbproc, &daterec, &datetime);
                                m_txn.OrderStatus.o_entry_d.year   = daterec.year;
                                m_txn.OrderStatus.o_entry_d.month  = daterec.month;
                                m_txn.OrderStatus.o_entry_d.day    = daterec.day;
                                m_txn.OrderStatus.o_entry_d.hour   = daterec.hour;
                                m_txn.OrderStatus.o_entry_d.minute = daterec.minute;
                                m_txn.OrderStatus.o_entry_d.second = daterec.second;
                        }
                        if(pData=dbdata(m_dbproc, 6))
                                m_txn.OrderStatus.o_carrier_id = (*(DBSMALLINT *) pData);
                        if(pData=dbdata(m_dbproc, 7))
                                dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,7),
                                                                        SQLFLT8, (BYTE *)&m_txn.OrderStatus.c_balance, 8);
                        if(pData=dbdata(m_dbproc, 8))
                                m_txn.OrderStatus.o_id = (*(DBINT *) pData);

                        DiscardNextRows(0);
                        DiscardNextResults(0);


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

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

```
		}			// while (TRUE)

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


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

		char			szName[MAX_SP_NAME_LEN];

		ResetError();

		while (TRUE)
		{
			try
			{
				// dbrpcinit(m_dbproc, "tpcc_delivery", 0);
				sprintf(szName, "%stpcc_delivery", m_szSPPrefix);
				dbrpcinit(m_dbproc, szName, 0);

				dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.Delivery.w_id);
				dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.Delivery.o_carrier_id);

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

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

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

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

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

				DiscardNextRows(0);
				DiscardNextResults(0);

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

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

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

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

Kit MSTPCC451\WEBCLNT\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
                                char            m_szSPPrefix[3] ;       // stored procedures prefix

                                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);
```

Kit MSTPCC451\WEBCLNT\db_odbc_dll\src\tpcc_odbc.cpp

```
/*              FILE:                           TPCC_ODBC.CPP
 *                                              Microsoft TPC-C Kit Ver. 4.42.000
 *                                              Copyright Microsoft, 2002
 *       All Rights Reserved
 *
 *                                              Version 4.10.000 audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *              PURPOSE:        Implements ODBC calls for TPC-C txns.
 *              Contact:        Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                              4.42.000 - changed w_id fields from short to long to support >32K warehouses
 *                              4.20.000 - updated rev number to match kit
 *                              4.10.001 - not deleting error class in catch handler on deadlock retry;
 *                                                      not a functional bug, but a memory leak
 */

#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"

#define MAX_SP_NAME_LEN                         256             //maximum length of a stored procedure name with parameters

// 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
        LPCWSTR szSPPrefix )                            // prefix to append to the stored procedure names
{
        return new CTPCC_ODBC( szServer, szUser, szPassword, szHost, szDatabase, szSPPrefix );
}


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
        LPCWSTR szSPPrefix                                                      // prefix to append to the stored procedure names
        )
{
        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;

        wcsncpy(m_szSPPrefix, szSPPrefix, sizeof(m_szSPPrefix)/sizeof(m_szSPPrefix[0]));


        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()
{
                int rc;
                wchar_t                 szName[MAX_SP_NAME_LEN];

                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_SLONG, SQL_INTEGER, 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);

                //Prepare Stock Level statement
                //rc = SQLPrepareW(m_hstmt, (SQLWCHAR*)L"{call tpcc_stocklevel(?,?,?)}", SQL_NTS);
                _snwprintf(szName, sizeof(szName)/sizeof(szName[0]), L"exec %stpcc_stocklevel ?,?,?", m_szSPPrefix);
                rc = SQLPrepareW(m_hstmt, (SQLWCHAR*)szName, SQL_NTS);
                if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                                ThrowError(CODBCERR::ePrepare);
}

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

                m_hstmt = m_hstmtStockLevel;

                while (TRUE)
                {
                                try
                                {
                                        rc = SQLExecute(m_hstmt);
                                        if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                                                ThrowError(CODBCERR::eExecute);

                                        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()
{
	int rc;
	wchar_t		szName[MAX_SP_NAME_LEN];

	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_SLONG, SQL_INTEGER, 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_SLONG, SQL_INTEGER, 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);

	//Prepare the New Order statement
	//rc = SQLPrepareW(m_hstmt, (SQLWCHAR*)L"{call tpcc_neworder(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,"
	//								L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)}",
SQL_NTS);
	_snwprintf(szName, sizeof(szName)/sizeof(szName[0]), L"exec %stpcc_neworder ?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,"
									L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?",
m_szSPPrefix);
	rc = SQLPrepareW(m_hstmt, (SQLWCHAR*)szName, SQL_NTS);
	if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
		ThrowError(CODBCERR::ePrepare);
}

void CTPCC_ODBC::NewOrder()
{
	int						i;
	RETCODE					rc;
	int						iTryCount = 0;
									// 0         1         2
									// 012345678901234567890123456789
	//wchar_t					szSqlTemplate[] = L"{call tpcc_neworder(?,?,?,?,?,"
	//								 L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,"
	//								 L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,"
	//								 L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)}";
	//
	m_hstmt = m_hstmtNewOrder;

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

	// clip statement buffer based on number of parameters
	// fixed part is 29 chars and variable part is 6 chars per line item
	//i = 29 + m_txn.NewOrder.o_ol_cnt*6;
	//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 = SQLExecute(m_hstmt);
			if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
				ThrowError(CODBCERR::eExecute);

			// 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()
{
                        int rc;
                        wchar_t         szName[MAX_SP_NAME_LEN];

                        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_SLONG, SQL_INTEGER, 0, 0, &m_txn.Payment.w_id, 0, NULL) != SQL_SUCCESS
                                || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 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);

                        //Prepare Payment statement
                        //rc = SQLPrepareW(m_hstmt, (SQLWCHAR*)L"{call tpcc_payment(?,?,?,?,?,?,?)}", SQL_NTS);
                        _snwprintf(szName, sizeof(szName)/sizeof(szName[0]), L"exec %stpcc_payment ?,?,?,?,?,?,?", m_szSPPrefix);
                        rc = SQLPrepareW(m_hstmt, (SQLWCHAR*)szName, SQL_NTS);
                        if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                                ThrowError(CODBCERR::ePrepare);
}

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 = SQLExecute(m_hstmt);
                                        if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                                                ThrowError(CODBCERR::eExecute);

                                        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()
{
                        int rc;
                        wchar_t         szName[MAX_SP_NAME_LEN];

                        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_SLONG, SQL_INTEGER, 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_SLONG, &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);

		//rc = SQLPrepareW(m_hstmt, (SQLWCHAR*)L"{call tpcc_orderstatus(?,?,?,?)}", SQL_NTS);
		_snwprintf(szName, sizeof(szName)/sizeof(szName[0]), L"exec %stpcc_orderstatus ?,?,?,?", m_szSPPrefix);
		rc = SQLPrepareW(m_hstmt, (SQLWCHAR*)szName, SQL_NTS);
		if ( rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO )
				ThrowError(CODBCERR::ePrepare);
}

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 = SQLExecute(m_hstmt);
						if ( ((rc == SQL_SUCCESS_WITH_INFO) && (m_RowsFetched != 0)) || (rc == SQL_ERROR) )
								ThrowError(CODBCERR::eExecute);

						// 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()
{
		int rc;
		wchar_t                 szName[MAX_SP_NAME_LEN];

		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_SLONG, SQL_INTEGER, 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);
		}

		//prepare Delivery statement
		//rc = SQLPrepareW(m_hstmt, (SQLWCHAR*)L"{call tpcc_delivery(?,?)}", SQL_NTS);
		_snwprintf(szName, sizeof(szName)/sizeof(szName[0]), L"exec %stpcc_delivery ?,?", m_szSPPrefix);
		rc = SQLPrepareW(m_hstmt, (SQLWCHAR*)szName, SQL_NTS);
		if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
				ThrowError(CODBCERR::ePrepare);
}

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

		m_hstmt = m_hstmtDelivery;

		while (TRUE)
		{
				try
				{
						rc = SQLExecute(m_hstmt);
```

```
                                        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);
}
```

Kit MSTPCC451\WEBCLNT\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;

                wchar_t                         m_szSPPrefix[32];  // stored procedures prefix

                // 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, LPCWSTR szSPPrefix);
                        ~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, LPCWSTR szSPPrefix );

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

Kit MSTPCC451\WEBCLNT\install\src\install.c

```
/*              FILE:                           INSTALL.C
 *                                                      Microsoft TPC-C Kit Ver. 4.51.000
 *                                                      Copyright Microsoft, 2003
 *       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
 *    4.50.000 - added IIS6 configuration options
 *                              4.51.000 - added routines to copy Visual Studio runtime module (MSVCR70.DLL)
 *                                                      to SystemRoot\System32
 */

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

static          int                     iIISMajorVersion;
static          int                     iNumberOfProcessors;

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, char *szWindowsPath);
static          void                    ReadRegistrySettings(void);
static          void                    WriteRegistrySettings(char *szDllPath);
static          BOOL                    RegisterDLL(char *szFileName);
static          int                             CopyFiles(HWND hDlg, char *szDllPath, char *szWindowsPath);
static          BOOL                    GetInstallPath(char *szDllPath);
static          BOOL                    GetWindowsInstallPath(char *szWindowsPath);
static          void                    GetVersionInfo(char *szDLLPath, char *szExePath);
static          BOOL                    CheckWWWebService(void);
static          BOOL                    StartWWWebService(void);
static          BOOL                    StopWWWebService(void);
static          void                    UpdateDialog(HWND hDlg);
static          void                    ConfigureIIS6(HWND hwnd, HWND hDlg);

SYSTEM_INFO siSysInfo;

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			szWindowsPath[256];
	static char			szExePath[256];

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

			if ( GetWindowsInstallPath(szWindowsPath) )
			{
				MessageBox(hwnd, "Error: Cannot determine Windows System Root.", NULL, MB_ICONSTOP | MB_OK);
				EndDialog(hwnd, FALSE);
				return TRUE;
			}

			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;
			iMaxPoolThreads = iPoolThreadLimit;
			iThreadTimeout = 86400;
			iListenBackLog = 15;
			iAcceptExOutstanding = 40;

			ReadTPCCRegistrySettings( &Reg );
			ReadRegistrySettings();

			// copy the hardware information to the SYSTEM_INFO structure
			GetSystemInfo(&siSysInfo);
			// store the number of processors on this system
			iNumberOfProcessors = siSysInfo.dwNumberOfProcessors;

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

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

                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);

                // check to see if the web services are running
                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);
                }

                // write binaries to inetpub\wwwroot
                rc = CopyFiles(hDlg, szDllPath, szWindowsPath);
                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;
                }

                // while we have the web services shutdown, check to see if this
                // is IIS6.  If it is, then call ConfigureIIS6
                if ( iIISMajorVersion == 6)
                {
                        ConfigureIIS6(hwnd, 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();
        }

        // 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, "SOFTWARE\\Microsoft\\InetStp", 0, KEY_READ, &hKey) == ERROR_SUCCESS )
        {
                        size = sizeof(iIISMajorVersion);
                        if ( RegQueryValueEx(hKey, "MajorVersion", 0, &type, (char *)&iIISMajorVersion, &size) == ERROR_SUCCESS )
                                        if ( !iIISMajorVersion )
                                                        iIISMajorVersion = 5;
        }

        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE, "SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Parameters", 0, KEY_READ, &hKey) == ERROR_SUCCESS )
        {
                        if ( iIISMajorVersion == 6)
                        {
                                        // since IIS6 handles the pool thread parameters differently, we need to fill in the dialog
                                        // with the MaxPoolThreads rather than the PoolThreadLimit
                                        // for ease of coding, we are just going to stuff the value into iPoolThreadLimit
        size = sizeof(iPoolThreadLimit);
                                        if ( RegQueryValueEx(hKey, "MaxPoolThreads", 0, &type, (char *)&iPoolThreadLimit, &size) == ERROR_SUCCESS )
                                        if ( !iPoolThreadLimit )
                                                        iPoolThreadLimit =  iMaxPhysicalMemory * 2;
                        }
                        else
                        {
                                        size = sizeof(iPoolThreadLimit);
                                        if ( RegQueryValueEx(hKey, "MaxPoolThreads", 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);
        }
        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE, "SYSTEM\\CurrentControlSet\\Services\\HTTP\\Parameters", 0, KEY_READ, &hKey) == ERROR_SUCCESS )
        {
                        size = sizeof(iUriEnableCache);
                        if ( RegQueryValueEx(hKey, "UriEnableCache", 0, &type, (char *)&iUriEnableCache, &size) == ERROR_SUCCESS )
                                        if ( !iUriEnableCache )
                                                        iUriEnableCache = 0;

                        size = sizeof(iUriScavengerPeriod);
                        if ( RegQueryValueEx(hKey, "UriScavengerPeriod", 0, &type, (char *)&iUriScavengerPeriod, &size) == ERROR_SUCCESS )
                                        if ( !iUriScavengerPeriod )
                                                        iUriScavengerPeriod = 10800;

                        size = sizeof(iMaxConnections);
                        if ( RegQueryValueEx(hKey, "MaxConnections", 0, &type, (char *)&iMaxConnections, &size) == ERROR_SUCCESS )
                                        if ( !iMaxConnections )
                                                        iMaxConnections = 100000;

                        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 )
        {
                // if this is IIS6, then we need to treat the PoolThreadLimit differently
                // if IIS6, then PoolThreadLimit is the maximum number of threads for the entire system.
                // IIS6 added MaxPoolThreads which controls the number of threads per processor.  For IIS6
                // we will set MaxPoolThreads to the value the user proivided in the dialog and then set
                // PoolThreadLimit to MaxPoolThreads * number of processors on this system
                if ( iIISMajorVersion == 6 )
                {
                        iPoolThreadLimit = iMaxPoolThreads * iNumberOfProcessors;
                        RegSetValueEx(hKey, "PoolThreadLimit", 0, REG_DWORD, (char *)&iPoolThreadLimit, sizeof(iPoolThreadLimit));
                        RegSetValueEx(hKey, "MaxPoolThreads", 0, REG_DWORD, (char *)&iMaxPoolThreads, sizeof(iMaxPoolThreads));
                }
                else
                {
                        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, char *szWindowsPath)
{
        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 MSVCR70.DLL
        strcpy( szLastFileName, "msvcr70.dll" );
        if (!FileFromResource( "MSVCRT70", IDR_MSVCRT701, szWindowsPath, 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);

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


                // Registry key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SystemRoot is used to find the
                // system root to install the VC70 DLL.

                szWindowsPath[0] =  0;
                bRc = TRUE;
                if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion", 0, KEY_ALL_ACCESS, &hKey) == ERROR_SUCCESS )
                {
                        sv = sizeof(szData);
                        iRc = RegQueryValueEx( hKey, "SystemRoot", NULL, NULL, szData, &sv );
                        if (iRc == ERROR_SUCCESS)
                        {
                                bRc = FALSE;
                                strcpy(szWindowsPath, szData);
                                len = strlen(szWindowsPath);
                                if ( szWindowsPath[len-1] != '\\' )
                                {
                                        szWindowsPath[len] = '\\';
                                        szWindowsPath[len+1] = 0;
                                }
                                // now append the path to SYSTEM32
                                strcat(szWindowsPath, "SYSTEM32\\");
                        }

                        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);
        schService = OpenService(schSCManager, TEXT("IISADMIN"), 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;
}

static void ConfigureIIS6(HWND hwnd, HWND hDlg)
{
        int                             irc;
        char            szErrTxt[128];
        FILE            *fErrorFile;

        SetDlgItemText(hDlg, IDC_STATUS, "Configuring IIS6...");
        //SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        irc = system("IIS6_CONFIG.CMD");

        // since the return code from the command file is always 1,
        // check to see if the file iis6_config.err exists
        // if it does, then something hosed
        fErrorFile = fopen("IIS6_CONFIG.err","r");
        if ( fErrorFile != NULL )
        {
                ShowWindow(hwnd, SW_SHOWNA);
                DestroyWindow(hDlg);
```

```
                                        strcpy( szErrTxt, "IIS6 configuration error." );
                                        strcat( szErrTxt, "Check iis6_config.err" );
                                        MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP | MB_OK);
                                        EndDialog(hwnd, 0);
                                        return;
                        }
}
```

Kit MSTPCC451\WEBCLNT\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
//
```

Kit MSTPCC451\WEBCLNT\install\src\install.rc

```
// Microsoft Visual C++ 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_SETFONT | DS_MODALFRAME | DS_CENTER | WS_MINIMIZEBOX | WS_POPUP |
    WS_CAPTION | WS_SYSMENU
CAPTION "TPC-C Web Client Installation Utility"
FONT 8, "MS Sans Serif", 0, 0, 0x1
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_SETFONT | 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  0, 0, 91, 40
STYLE DS_SYSMODAL | DS_SETFONT | 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  0, 0, 291, 202
STYLE DS_SETFONT | 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
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
BEGIN
    "resource.h\0"
END

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

3 TEXTINCLUDE
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                    "icon1.ico"
IDI_ICON2               ICON                    "icon2.ico"

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

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

/////////////////////////////////////////////////////////////////////////////
//
// Version
//

VS_VERSION_INFO VERSIONINFO
 FILEVERSION 0,4,50,0
 PRODUCTVERSION 0,4,50,0
 FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
 FILEFLAGS 0x1L
#else
 FILEFLAGS 0x0L
#endif
 FILEOS 0x40004L
 FILETYPE 0x1L
 FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904b0"
        BEGIN
            VALUE "Comments", "TPC-C Web Client Installer"
            VALUE "CompanyName", "Microsoft"
            VALUE "FileDescription", "install"
            VALUE "FileVersion", "0, 4, 20, 0"
            VALUE "InternalName", "install"
            VALUE "LegalCopyright", "Copyright © 1999"
            VALUE "OriginalFilename", "install.exe"
            VALUE "ProductName", "Microsoft install"
            VALUE "ProductVersion", "0, 4, 20, 0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END


/////////////////////////////////////////////////////////////////////////////
//
```

```
// LICENSE
//

IDR_LICENSE1            LICENSE                 "license.txt"

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

/////////////////////////////////////////////////////////////////////////////
//
// MSVCRT70
//

IDR_MSVCRT701           MSVCRT70                "C:\\WINDOWS\\system32\\msvcr70.dll"
#endif    // English (U.S.) resources
/////////////////////////////////////////////////////////////////////////////


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


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

Kit MSTPCC451\WEBCLNT\install\src\install_com.cpp

```
/*                      FILE:                                   INSTALL_COM.CPP
 *                                                                      Microsoft TPC-C Kit Ver. 4.51.000
 *                                                                      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;
}
```

Kit MSTPCC451\WEBCLNT\install\src\RESOURCE.H

```
//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ 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 IDR_MSVCRT701                   130
#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        131
#define _APS_NEXT_COMMAND_VALUE         40001
#define _APS_NEXT_CONTROL_VALUE         1031
#define _APS_NEXT_SYMED_VALUE           101
#endif
#endif
```

Kit MSTPCC451\WEBCLNT\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
```

Kit MSTPCC451\WEBCLNT\isapi_dll\src\tpcc.cpp

```
/*                      FILE:                                   TPCC.C
 *                                                              Microsoft TPC-C Kit Ver. 4.20.000
 *                                                              Copyright Microsoft, 1999
 *      All Rights Reserved
 *
 *                                                              Version 4.10.000 audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *                      PURPOSE:        Main module for TPCC.DLL which is an ISAPI service dll.
 *                      Contact:        Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                                      4.20.000 - reworked error handling; added options for COM and Encina txn monitors
 */

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

#include <sqltypes.h>

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

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

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

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

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

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


#define         LEN_ERR_STRING                  256

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

char                            szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

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

// The WEBCLIENT_VERSION string specifies the version level of this web client interface.
// The RTE must be synchronized with the interface level on login, otherwise the login
// will fail.  This is a sanity check to catch problems resulting from mismatched versions
// of the RTE and web client.
#define WEBCLIENT_VERSION "410"

static          CRITICAL_SECTION                                TermCriticalSection;

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

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

// For deferred Delivery txns:

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

HANDLE                                                                  hWorkerSemaphore        = INVALID_HANDLE_VALUE;
HANDLE                                                                  hDoneEvent                              = INVALID_HANDLE_VALUE;
HANDLE                                                                  *pDeliHandles           = NULL;

// configuration settings from registry
TPCCREGISTRYDATA                        Reg;


DWORD                                                                   dwNumDeliveryThreads = 4;
CRITICAL_SECTION                        DelBuffCriticalSection;                 //critical section for delivery transactions cache
DELIVERY_TRANSACTION            *pDelBuff                               = NULL;
DWORD                                                                   dwDelBuffSize           = 100;                  // size of circular buffer for delivery txns
DWORD                                                                   dwDelBuffFreeCount;                             // number of buffers free
DWORD                                                                   dwDelBuffBusyIndex      = 0;                    // index position of entry waiting to be delivered
DWORD                                                                   dwDelBuffFreeIndex      = 0;                    // index position of unused entry

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

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

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

// debugging...
// DebugBreak();

                        try
                        {
                                switch( ul_reason_for_call )
                                {
```

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

		DisableThreadLibraryCalls((HMODULE)hModule);
		InitializeCriticalSection(&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
		StartCAP();
#endif

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

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

					throw new CWEBCLNT_ERR( ERR_INVALID_TERMID );
				}

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

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


			switch(iCmd)
			{
			case 0:
				WelcomeForm(pECB, szBuffer);
				break;
			case 1:
				switch( FormId )
				{
					case WELCOME_FORM:
					case MAIN_MENU_FORM:
						break;
					case NEW_ORDER_FORM:
						ProcessNewOrderForm(pECB, TermId, szBuffer);
						break;
					case PAYMENT_FORM:
						ProcessPaymentForm(pECB, TermId, szBuffer);
						break;
					case DELIVERY_FORM:
						ProcessDeliveryForm(pECB, TermId, szBuffer);
						break;
					case ORDER_STATUS_FORM:
						ProcessOrderStatusForm(pECB, TermId, szBuffer);
						break;
```

```
                                                case STOCK_LEVEL_FORM:
                                                        ProcessStockLevelForm(pECB, TermId, szBuffer);
                                                        break;
                                        }
                                        break;
                                case 2:
                                        // new-order selected from menu; display new-order input form
                                        MakeNewOrderForm(TermId, NULL, INPUT_FORM, szBuffer);
                                        break;
                                case 3:
                                        // payment selected from menu; display payment input form
                                        MakePaymentForm(TermId, NULL, INPUT_FORM, szBuffer);
                                        break;
                                case 4:
                                        // delivery selected from menu; display delivery input form
                                        MakeDeliveryForm(TermId, NULL, INPUT_FORM, szBuffer);
                                        break;
                                case 5:
                                        // order-status selected from menu; display order-status input form
                                        MakeOrderStatusForm(TermId, NULL, INPUT_FORM, szBuffer);
                                        break;
                                case 6:
                                        // stock-level selected from menu; display stock-level input form
                                        MakeStockLevelForm(TermId, NULL, INPUT_FORM, szBuffer);
                                        break;
                                case 7:
                                        // ExitCmd
                                        TermDelete(TermId);
                                        WelcomeForm(pECB, szBuffer);
                                        break;
                                case 8:
                                        SubmitCmd(pECB, szBuffer);
                                        break;
                                case 9:
                                        // menu
                                        MakeMainMenuForm(TermId, Term.pClientData[TermId].iSyncId, szBuffer);
                                        break;
                                case 10:
                                        // CMD=Clear
                                        // resets all connections; should only be used when no other connections are active
                                        TermDeleteAll();
                                        TermInit();
                                        WelcomeForm(pECB, szBuffer);
                                        break;
                                case 11:        // CMD=Stats
                                        StatsCmd(pECB, szBuffer);
                                        break;
                                }
                        }
                        catch (CBaseErr *e)
                        {
                                ErrorForm( pECB, e->ErrorType(), e->ErrorNum(), TermId, iSyncId, e->ErrorText(), szBuffer );
                                delete e;
                        }
                        catch (...)
                        {
                                ErrorForm( pECB, ERR_TYPE_WEBDLL, 0, TermId, iSyncId, "Error: Unhandled exception in Web Client.", szBuffer );
                        }

#ifdef ICECAP
                        StopCAP();
#endif


                        lpbSize = strlen(szBuffer);
                        wsprintf(szHeader1,
                                                "Content-Type: text/html\r\n"
                                                "Content-Length: %d\r\n"
                                                "Connection: Keep-Alive\r\n\r\n" , lpbSize);
                        strcat( szHeader1, szBuffer );

                        (*pECB->ServerSupportFunction)(pECB->ConnID, HSE_REQ_SEND_RESPONSE_HEADER, szHeader, (LPDWORD) &dwSize, (LPDWORD)szHeader1);

                        //finish up and keep connection
                        pECB->dwHttpStatusCode = 200;
                        return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}


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

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

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

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

      (VOID) DeregisterEventSource(hEventSource);
   }
}


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

/*static*/ void DeliveryWorkerThread(void *ptr)
{
                        CTPCC_BASE                      *pTxn = NULL;

                        DELIVERY_TRANSACTION                            delivery;
                        PDELIVERY_DATA                                          pDeliveryData;
                        TXN_RECORD_TPCC_DELIV_DEF       txnDeliRec;

                        DWORD                                                                   index;
                        HANDLE                                                                  handles[2];

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

                        assert(txnDelilog != NULL);

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

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

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

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

                                // make a local copy of current entry from delivery buffer and increment buffer index
                                EnterCriticalSection(&DelBuffCriticalSection);
                                delivery = *(pDelBuff+dwDelBuffBusyIndex);
                                dwDelBuffFreeCount++;
                                dwDelBuffBusyIndex++;
                                if (dwDelBuffBusyIndex == dwDelBuffSize)                // wrap-around if at end of buffer
                                        dwDelBuffBusyIndex = 0;

                                LeaveCriticalSection(&DelBuffCriticalSection);

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

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

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

                                //log txn
                                txnDeliRec.TxnStatus = ERR_SUCCESS;
                                for (int i=0; i<10; i++)
                                        txnDeliRec.o_id[i] = pDeliveryData->o_id[i];
                                txnDeliRec.DeltaT4 = (int)(Get64BitTime(&trans_end) - txnDeliRec.TxnStartT0);
                                txnDeliRec.DeltaTxnExec = (int)(Get64BitTime(&trans_end) - Get64BitTime(&trans_start));

                                if (txnDelilog != NULL)
                                        txnDelilog->WriteToLog(&txnDeliRec);
                        }
                }
                catch (CBaseErr *e)
                {
                        char szTmp[1024];
                        wsprintf( szTmp, "Error in Delivery Txn thread.  %s", e->ErrorText() );
                        WriteMessageToEventLog( szTmp );

                        // log the error txn
                        txnDeliRec.TxnStatus = e->ErrorType();
                        if (txnDelilog != NULL)
                                txnDelilog->WriteToLog(&txnDeliRec);

                        delete e;
                }
                catch (...)
                {
                        // unhandled exception; shouldn't happen; not much we can do...
                        WriteMessageToEventLog(TEXT("Unhandled exception caught in DeliveryWorkerThread."));
                }
        }

ErrorExit:
        delete pTxn;
        _endthread();
}

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

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

        EnterCriticalSection(&DelBuffCriticalSection);
        if (dwDelBuffFreeCount > 0)
        {
                bError = FALSE;
                (pDelBuff+dwDelBuffFreeIndex)->w_id              = w_id;
                (pDelBuff+dwDelBuffFreeIndex)->o_carrier_id      = o_carrier_id;
                GetLocalTime(&(pDelBuff+dwDelBuffFreeIndex)->queue);

                dwDelBuffFreeCount--;
                dwDelBuffFreeIndex++;
                if (dwDelBuffFreeIndex == dwDelBuffSize)
                        dwDelBuffFreeIndex = 0;                 // wrap-around if at end of buffer
        }
        else
                // No free buffers.  Return an error, which indicates that the delivery buffer is full.
                // Most likely, the number of delivery worker threads needs to be increased to keep up
                // with the txn rate.
                bError = TRUE;
        LeaveCriticalSection(&DelBuffCriticalSection);

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

        return bError;
}

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

void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd, int *pFormId, int *pTermId, int *pSyncId)
{
        char *ptr = pECB->lpszQueryString;
        char szBuffer[25];
        int i;

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

        *pCmd    = 0;                                   // default is the login screen
        *pTermId = 0;

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

        // parse FORMID, TERMID, and SYNCID
```

```
		*pFormId = GetIntKeyValue(&ptr, "FORMID", NO_ERR, NO_ERR);
		*pTermId = GetIntKeyValue(&ptr, "TERMID", NO_ERR, NO_ERR);
		*pSyncId = GetIntKeyValue(&ptr, "SYNCID", NO_ERR, NO_ERR);

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

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

/* FUNCTION: void WelcomeForm
 *
 */

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

		//welcome to tpc-c html form buffer, this is first form client sees.
		strcpy( szBuffer,	"<HTML><HEAD><TITLE>TPC-C Web Client</TITLE></HEAD><BODY>"
									"<B><BIG>Microsoft TPC-C Web Client (ver 4.20)</BIG></B> <BR> <BR>"
									"<font face=\"Courier New\"><PRE>"
									"Compiled: " __DATE__ " " __TIME__ " <BR>"
									"Source:  " __FILE__ " (" __TIMESTAMP__ ") <BR>"
									"</PRE></font>"
									"<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
									"<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"
									"<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
									"<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"1\">"
									"<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"0\">"
									"<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"0\">"
									"<INPUT TYPE=\"hidden\" NAME=\"VERSION\" VALUE=\"" WEBCLIENT_VERSION "\">"
		);

		sprintf( szTmp,		"Configuration Settings: <BR><font face=\"Courier New\" color=\"blue\"><PRE>"
											"Txn Monitor          = <B>%s</B><BR>"
											"Database protocol    = <B>%s</B><BR>"
											"Max Connections      = <B>%d</B><BR>"
											"# of Delivery Threads  = <B>%d</B><BR>"
											"Max Pending Deliveries = <B>%d</B><BR>"
		, szTxnMonNames[Reg.eTxnMon], szDBNames[Reg.eDB_Protocol],
		Reg.dwMaxConnections, dwNumDeliveryThreads, dwDelBuffSize );
		strcat( szBuffer, szTmp);

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

		if (Reg.eTxnMon == None)
			// connection options may be specified when not using a txn monitor
			sprintf( szTmp,		"Please enter your database options for this connection:<BR>"
									"<font face=\"Courier New\" color=\"blue\"><PRE>"
									"DB Server   = <INPUT NAME=\"db_server\" SIZE=20 VALUE=\"%s\"><BR>"
									"DB User ID  = <INPUT NAME=\"db_user\" SIZE=20 VALUE=\"%s\"><BR>"
									"DB Password = <INPUT NAME=\"db_passwd\" SIZE=20 VALUE=\"%s\"><BR>"
									"DB Name     = <INPUT NAME=\"db_name\" SIZE=20 VALUE=\"%s\"><BR>"
									"</PRE></font>"
			, Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
		else
			// if using a txn monitor, connection options are determined from registry; can't
			// set per user.  show options fyi
			sprintf( szTmp,		"Database options which will be used by the transaction monitor:<BR>"
									"<font face=\"Courier New\" color=\"blue\"><PRE>"
									"DB Server           = <B>%s</B><BR>"
									"DB User ID          = <B>%s</B><BR>"
									"DB Password         = <B>%s</B><BR>"
									"DB Name             = <B>%s</B><BR>"
									"</PRE></font>"
			, Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
		strcat( szBuffer, szTmp);

		sprintf( szTmp,		"Please enter your Warehouse and District for this session:<BR>"
									"<font face=\"Courier New\" color=\"blue\"><PRE>" );
		strcat( szBuffer, szTmp);
		strcat( szBuffer,	"Warehouse ID = <INPUT NAME=\"w_id\" SIZE=6><BR>"
									"District ID  = <INPUT NAME=\"d_id\" SIZE=2><BR>"
									"</PRE></font><HR>"
									"<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Submit\">"
									"</FORM></BODY></HTML>");
}

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

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

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

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

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

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

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


		iNewTerm = TermAdd();

		Term.pClientData[iNewTerm].w_id = w_id;
		Term.pClientData[iNewTerm].d_id = d_id;

		try
		{
			if (Reg.eTxnMon == TUXEDO)
				Term.pClientData[iNewTerm].pTxn = pCTPCC_TUXEDO_new();
			else if (Reg.eTxnMon == ENCINA)
				Term.pClientData[iNewTerm].pTxn = pCTPCC_ENCINA_new();
```

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

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

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

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

                EnterCriticalSection(&TermCriticalSection);

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

                LeaveCriticalSection(&TermCriticalSection);

                wsprintf( szBuffer,
                                "<HTML><HEAD><TITLE>TPC-C Web Client Stats</TITLE></HEAD>"
                                "<BODY><B><BIG> Total Active Connections: %d </BIG></B><BR></BODY></HTML>"
                                , iTotal );
}

char *CWEBCLNT_ERR::ErrorText()
{
                static SERRORMSG errorMsgs[] =
                {
                        {       ERR_COMMAND_UNDEFINED,                                  "Command undefined."
                                                                                                },
                        {       ERR_D_ID_INVALID,                                       "Invalid District ID Must be 1 to 10."
                },
                        {       ERR_DELIVERY_CARRIER_ID_RANGE,                  "Delivery Carrier ID out of range must be 1 - 10."
                                },
                        {       ERR_DELIVERY_CARRIER_INVALID,           "Delivery Carrier ID invalid must be numeric 1 - 10."
                },
                        {       ERR_DELIVERY_MISSING_OCD_KEY,                   "Delivery missing Carrier ID key \"OCD\"\"."
                                        },
                        {       ERR_DELIVERY_THREAD_FAILED,                     "Could not start delivery worker thread."
                                        },
                        {       ERR_GETPROCADDR_FAILED,                                 "Could not map proc in DLL.  GetProcAddr error.  DLL="
                                },
                        {       ERR_HTML_ILL_FORMED,                                    "Required key field is missing from HTML string."
                                        },
                        {       ERR_INVALID_SYNC_CONNECTION,            "Invalid Terminal Sync ID."
                        },
                        {       ERR_INVALID_TERMID,                                     "Invalid Terminal ID."
                                                                                                },
                        {       ERR_LOADDLL_FAILED,                                     "Load of DLL failed.  DLL="
                                                                                                },
                        {       ERR_MAX_CONNECTIONS_EXCEEDED,           "No connections available.  Max Connections is probably too low."  },
                        {       ERR_MISSING_REGISTRY_ENTRIES,           "Required registry entries are missing.  Rerun INSTALL to correct."             },
                        {       ERR_NEWORDER_CUSTOMER_INVALID,          "New Order customer id invalid data type, range = 1 to 3000."
                },
                        {       ERR_NEWORDER_CUSTOMER_KEY,                      "New Order missing Customer key \"CID\"\"."
                                },
                        {       ERR_NEWORDER_DISTRICT_INVALID,          "New Order District ID Invalid range 1 - 10."
                                        },
                        {       ERR_NEWORDER_FORM_MISSING_DID,          "New Order missing District key \"DID\"\"."
                                        },
                        {       ERR_NEWORDER_ITEMID_INVALID,            "New Order Item Id is wrong data type, must be numeric."                 },
                        {       ERR_NEWORDER_ITEMID_RANGE,                      "New Order Item Id is out of range. Range = 1 to 999999."
                },
                        {       ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,      "New Order Item_Id field entered without a corresponding Supp_W."               },
                        {       ERR_NEWORDER_MISSING_IID_KEY,                   "New Order missing Item Id key \"IID\"\"."
                                                                                                },
                        {       ERR_NEWORDER_MISSING_QTY_KEY,                   "New Order Missing Qty key \"Qty##\"\"."
                                },
                        {       ERR_NEWORDER_MISSING_SUPPW_KEY,         "New Order missing Supp_W key \"SP##\"\"."
                                        },
                        {       ERR_NEWORDER_NOITEMS_ENTERED,           "New Order No order lines entered."
                                },
                        {       ERR_NEWORDER_QTY_INVALID,                       "New Order Qty invalid must be numeric range 1 - 99."
                        },
                        {       ERR_NEWORDER_QTY_RANGE,                                 "New Order Qty is out of range. Range = 1 to 99."
                                        },
                        {       ERR_NEWORDER_QTY_WITHOUT_SUPPW,         "New Order Qty field entered without a corresponding Supp_W."
                },
                        {       ERR_NEWORDER_SUPPW_INVALID,             "New Order Supp_W invalid data type must be numeric."
                        },
                        {       ERR_NO_SERVER_SPECIFIED,                        "No Server name specified."
                                },
                        {       ERR_ORDERSTATUS_CID_AND_CLT,            "Order Status Only Customer ID or Last Name may be entered, not both."  },
                        {       ERR_ORDERSTATUS_CID_INVALID,            "Order Status Customer ID invalid, range must be numeric 1 - 3000."             },
                        {       ERR_ORDERSTATUS_CLT_RANGE,                      "Order Status Customer last name longer than 16 characters."
                },
                        {       ERR_ORDERSTATUS_DID_INVALID,            "Order Status District invalid, value must be numeric 1 - 10."                  },
                        {       ERR_ORDERSTATUS_MISSING_CID_CLT,        "Order Status Either Customer ID or Last Name must be entered."                 },
                        {       ERR_ORDERSTATUS_MISSING_CID_KEY,        "Order Status missing Customer key \"CID\"\"."
                                },
                        {       ERR_ORDERSTATUS_MISSING_CLT_KEY,        "Order Status missing Customer Last Name key \"CLT\"\"."
                },
                        {       ERR_ORDERSTATUS_MISSING_DID_KEY,        "Order Status missing District key \"DID\"\"."
                                },
                        {       ERR_PAYMENT_CDI_INVALID,                        "Payment Customer district invalid must be numeric."
                        },
                        {       ERR_PAYMENT_CID_AND_CLT,                        "Payment Only Customer ID or Last Name may be entered, not both."       },
                        {       ERR_PAYMENT_CUSTOMER_INVALID,           "Payment Customer data type invalid, must be numeric."
                                },
                        {       ERR_PAYMENT_CWI_INVALID,                        "Payment Customer Warehouse invalid, must be numeric."
                                },
                        {       ERR_PAYMENT_DISTRICT_INVALID,           "Payment District ID is invalid, must be 1 - 10."
                                },
                        {       ERR_PAYMENT_HAM_INVALID,                        "Payment Amount invalid data type must be numeric."
                                },
                        {       ERR_PAYMENT_HAM_RANGE,                                  "Payment Amount out of range, 0 - 9999.99."
                },
                        {       ERR_PAYMENT_LAST_NAME_TO_LONG,          "Payment Customer last name longer than 16 characters."
                                },
                        {       ERR_PAYMENT_MISSING_CDI_KEY,            "Payment missing Customer district key \"CDI\"\"."
                                },
                        {       ERR_PAYMENT_MISSING_CID_CLT,            "Payment Either Customer ID or Last Name must be entered."
                },
                        {       ERR_PAYMENT_MISSING_CID_KEY,            "Payment missing Customer Key \"CID\"\"."
                                        },
                        {       ERR_PAYMENT_MISSING_CLT_KEY,            "Payment missing Customer Last Name key \"CLT\"\"."
                                },
                        {       ERR_PAYMENT_MISSING_CWI_KEY,            "Payment missing Customer Warehouse key \"CWI\"\"."
                                },
                        {       ERR_PAYMENT_MISSING_DID_KEY,            "Payment missing District Key \"DID\"\"."
                                },
                        {       ERR_PAYMENT_MISSING_HAM_KEY,            "Payment missing Amount key \"HAM\"\"."
                },
                        {       ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,   "Stock Level; missing Threshold key \"TT\"\"."
                                },
                        {       ERR_STOCKLEVEL_THRESHOLD_INVALID,       "Stock Level; Threshold value must be in the range = 1 - 99."                   },
                        {       ERR_STOCKLEVEL_THRESHOLD_RANGE,                 "Stock Level Threshold out of range, range must be 1 - 99."
                },
                        {       ERR_VERSION_MISMATCH,                                   "Invalid version field.  RTE and Web Client are probably out of sync." },
                        {       ERR_W_ID_INVALID,                                       "Invalid Warehouse ID."
                                                                                                },
```

```
		{		0,
		""
										}
		};

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

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

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

}

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

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

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

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

		*pQueryString = ptr;
		return;

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

/* FUNCTION: GetIntKeyValue
 *
 * PURPOSE:	This function parses a http formatted string for a specific key value.
 *
 * ARGUMENTS:	char				*pQueryString	http string from client browser
 *					char				*pKey				key value to look for
 *					WEBERROR		NoKeyErr		error value to throw if key not found
 *					WEBERROR		NotIntErr		error value to throw if value not numeric
 *
 * RETURNS:		integer
 *
 * ERROR:		if (the pKey value is not found) then
 *						if (NoKeyErr != NO_ERR)
 *							throw CWEBCLNT_ERR(err)
 *						else
 *							return 0
 *					else if (non-numeric char found) then
 *						if (NotIntErr != NO_ERR) then
 *							throw CWEBCLNT_ERR(err)
 *						else
 *							return 0
 *
 * COMMENTS:	http keys are formatted either KEY=value& or KEY=value\0. This DLL formats
 *					TPC-C input fields in such a manner that the keys can be extracted in the
 *					above manner.
 */

int GetIntKeyValue(char **pQueryString, char *pKey, WEBERROR NoKeyErr, WEBERROR NotIntErr)
{
		char *ptr0;
		char *ptr;

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

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

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

		*pQueryString = ptr;
		return atoi(ptr0);

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

/* FUNCTION: TermInit
 *
 * PURPOSE:	This function initializes the client terminal structure; it is called when the TPCC.DLL
 *				is first loaded by the inet service.
 *
 */
```

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

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

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

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

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

		LeaveCriticalSection(&TermCriticalSection);
}

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

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

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

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

		LeaveCriticalSection(&TermCriticalSection);
}

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

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

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

		EnterCriticalSection(&TermCriticalSection);
		if (Term.iFreeList != 0)
		{	// position is available
			iNewTerm = Term.iFreeList;
			Term.iFreeList = Term.pClientData[iNewTerm].iNextFree;
			Term.pClientData[iNewTerm].iNextFree = -1;  // indicates this position is in use
		}
		else
		{	// no open slots, so find the slot that hasn't been used in the longest time and reuse it
			for(iNewTerm=1, i=1, iTickCount=0x7FFFFFFF; i<Reg.dwMaxConnections; i++)
			{
				if (iTickCount > Term.pClientData[i].iTickCount)
				{
					iTickCount = Term.pClientData[i].iTickCount;
					iNewTerm = i;
				}
			}
			// if oldest term is less than one minute old, it probably means that more connections
			// are being attempted than were specified as "Max Connections" at install.  In this case,
			// do not bump existing connection; instead, return error to requestor.
			if ((GetTickCount() - iTickCount) < 60000)
			{
				LeaveCriticalSection(&TermCriticalSection);
				throw new CWEBCLNT_ERR( ERR_MAX_CONNECTIONS_EXCEEDED );
			}
		}

		Term.pClientData[iNewTerm].iTickCount = GetTickCount();
		Term.pClientData[iNewTerm].iSyncId = Term.iMasterSyncId++;
		Term.pClientData[iNewTerm].pTxn = NULL;

		LeaveCriticalSection(&TermCriticalSection);
		return iNewTerm;
}

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

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

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

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

			LeaveCriticalSection(&TermCriticalSection);
		}
}


/* FUNCTION: MakeErrorForm
 */

void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int iErrorNum, int iTermId, int iSyncId, char *szErrorText, char *szBuffer )
{
		wsprintf(szBuffer,
			"<HTML><HEAD><TITLE>TPC-C Error</TITLE></HEAD><BODY>"
			"<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
			"<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">"
			"<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"%d\">"
			"<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
			"<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
			"<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
			"<BOLD>An Error Occurred</BOLD><BR><BR>"
			"%s"
			"<BR><BR><HR>"
			"<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
```

```
                                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Payment..\">"
                                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Delivery..\">"
                                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Exit..\">"
                                        "</FORM></BODY></HTML>"
                                        , iType, iErrorNum, MAIN_MENU_FORM, iTermId, iSyncId, szErrorText );
}

/* FUNCTION: MakeMainMenuForm
 */

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

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

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

                c = wsprintf(szForm,
                                        "<HTML><HEAD><TITLE>TPC-C Stock Level</TITLE></HEAD><FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
                                        "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"
                                        "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
                                        "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
                                        "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
                                        "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
                                        "<PRE><font face=\"Courier\">                                   Stock-Level<BR>"
                                        "Warehouse: %6.6d   District: %2.2d<BR> <BR>",
                                        STOCK_LEVEL_FORM, iTermId, Term.pClientData[iTermId].iSyncId,
                                        Term.pClientData[iTermId].w_id, Term.pClientData[iTermId].d_id);

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

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

void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA *pNewOrderData, BOOL bInput, char *szForm)
{
                int                                     i, c;
                BOOL                    bValid;
                static                  char szBR[] = " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>";

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

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

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

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

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

                                        if ( bValid )
                                        {
                                                c += wsprintf(szForm+c, "%2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d",
```

```
                                                pNewOrderData->o_entry_d.day,
                                                pNewOrderData->o_entry_d.month,
                                                pNewOrderData->o_entry_d.year,
                                                pNewOrderData->o_entry_d.hour,
                                                pNewOrderData->o_entry_d.minute,
                                                pNewOrderData->o_entry_d.second);
                        }

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

                        if ( bValid )
                        {
                                        c += sprintf(szForm+c,
                                                                                        "%%Disc: %5.2f          <BR>"
                                                                                        "Order Number: %8.8d  Number of Lines: %2.2d       W_tax: %5.2f  D_tax: %5.2f  <BR> <BR>"
                                                                                        " Supp_W  Item_Id  Item Name                Qty  Stock  B/G  Price    Amount<BR>",
                                                100.0*pNewOrderData->c_discount,
                                                pNewOrderData->o_id,
                                                pNewOrderData->o_ol_cnt,
                                                100.0 * pNewOrderData->w_tax,
                                                100.0 * pNewOrderData->d_tax);

                                        for(i=0; i<pNewOrderData->o_ol_cnt; i++)
                                        {
                                                c += sprintf(szForm+c, "%6.6d   %6.6d   %-24s  %2.2d    %3.3d    %1.1s   $%6.2f  $%7.2f  <BR>",
                                                                pNewOrderData->OL[i].ol_supply_w_id,
                                                                pNewOrderData->OL[i].ol_i_id,
                                                                pNewOrderData->OL[i].ol_i_name,
                                                                pNewOrderData->OL[i].ol_quantity,
                                                                pNewOrderData->OL[i].ol_stock,
                                                                pNewOrderData->OL[i].ol_brand_generic,
                                                                pNewOrderData->OL[i].ol_i_price,
                                                                pNewOrderData->OL[i].ol_amount );
                                        }
                        }
                        else
                        {
                                        c += wsprintf(szForm+c,
                                                        "%Disc:<BR>"
                                                        "Order Number: %8.8d  Number of Lines:          W_tax:       D_tax:<BR> <BR>"
                                                        " Supp_W  Item_Id  Item Name              Qty  Stock  B/G  Price    Amount<BR>"
                                                        , pNewOrderData->o_id);

                                        i = 0;
                        }

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

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

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

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

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

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

        c += sprintf(szForm+c,

                                                "        %-20s                  %%Disc:  %5.2f<BR>",
                                                pPaymentData->c_street_2, 100.0*pPaymentData->c_discount);

                                c += wsprintf(szForm+c,
                                                "        %-20s %-2s %5.5s-%4.4s       Phone:  %6.6s-%3.3s-%3.3s-%4.4s<BR> <BR>",
                                                pPaymentData->c_city, pPaymentData->c_state, pPaymentData->c_zip, pPaymentData->c_zip+5,
                                                pPaymentData->c_phone, pPaymentData->c_phone+6, pPaymentData->c_phone+9, pPaymentData->c_phone+12 );

                                c += sprintf(szForm+c,
```

```
                                        "Amount Paid:          $%7.2f      New Cust-Balance: $%14.2f<BR>"
                                        "Credit Limit:   $%13.2f<BR> <BR>"
                                        , pPaymentData->h_amount, pPaymentData->c_balance
                                        , pPaymentData->c_credit_lim
                                        );

                        if ( pPaymentData->c_credit[0] == 'B' && pPaymentData->c_credit[1] == 'C' )
                                        c += wsprintf(szForm+c,
                                                                        "Cust-Data: %-50.50s<BR>          %-50.50s<BR>          %-50.50s<BR>          %-50.50s<BR>",
                                                                        pPaymentData->c_data, pPaymentData->c_data+50, pPaymentData->c_data+100, pPaymentData->c_data+150 );
                        else
                                        strcpy(szForm+c, "Cust-Data: <BR> <BR> <BR> <BR>");

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

}

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

void MakeOrderStatusForm(int iTermId, ORDER_STATUS_DATA *pOrderStatusData, BOOL bInput, char *szForm)
{
                int                                     i, c;
                static char szBR[] = " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>";

                c = wsprintf(szForm,
                                "<HTML><HEAD><TITLE>TPC-C Order-Status</TITLE></HEAD><BODY>"
                                "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
                                "<PRE><font face=\"Courier\">                                   Order-Status<BR>"
                                "Warehouse: %6.6d   ",
                                ORDER_STATUS_FORM, iTermId, Term.pClientData[iTermId].iSyncId, Term.pClientData[iTermId].w_id);

                if ( bInput )
                {
                                strcpy(szForm+c,
                                                "District: <INPUT NAME=\"DID\" SIZE=1><BR>"
                                                "Customer: <INPUT NAME=\"CID\" SIZE=4>   Name:                  <INPUT NAME=\"CLT\" SIZE=23><BR>"
                                                "Cust-Balance:<BR> <BR>"
                                                "Order-Number:            Entry-Date:                  Carrier-Number:<BR>"
                                                "Supply-W     Item-Id    Qty     Amount      Delivery-Date<BR> <BR> <BR> <BR> <BR>"
                                                " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR></font></PRE>"
                                                "<HR><INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\"><INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
                                                "</BODY></FORM></HTML>" );
                }
                else
                {
                                c += wsprintf(szForm+c,
                                                "District: %2.2d<BR>"
                                                "Customer: %4.4d   Name: %-16s %-2s %-16s<BR>",
                                                pOrderStatusData->d_id, pOrderStatusData->c_id,
                                                pOrderStatusData->c_first, pOrderStatusData->c_middle, pOrderStatusData->c_last);

                                c += sprintf(szForm+c, "Cust-Balance: $%9.2f<BR> <BR>",
                                                pOrderStatusData->c_balance);

                                c += wsprintf(szForm+c,
                                                "Order-Number: %8.8d   Entry-Date: %2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d   Carrier-Number: %2.2d<BR>"
                                                "Supply-W     Item-Id    Qty     Amount      Delivery-Date<BR>",
                                                pOrderStatusData->o_id,
                                                pOrderStatusData->o_entry_d.day,
                                                pOrderStatusData->o_entry_d.month,
                                                pOrderStatusData->o_entry_d.year,
                                                pOrderStatusData->o_entry_d.hour,
                                                pOrderStatusData->o_entry_d.minute,
                                                pOrderStatusData->o_entry_d.second,
                                                pOrderStatusData->o_carrier_id);

                                for(i=0; i< pOrderStatusData->o_ol_cnt; i++)
                                {
                                                c += sprintf(szForm+c, "  %6.6d     %6.6d     %2.2d     $%8.2f      %2.2d-%2.2d-%4.4d<BR>",
                                                                pOrderStatusData->OL[i].ol_supply_w_id,
                                                                pOrderStatusData->OL[i].ol_i_id,
                                                                pOrderStatusData->OL[i].ol_quantity,
                                                                pOrderStatusData->OL[i].ol_amount,
                                                                pOrderStatusData->OL[i].ol_delivery_d.day,
                                                                pOrderStatusData->OL[i].ol_delivery_d.month,
                                                                pOrderStatusData->OL[i].ol_delivery_d.year);
                                }

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

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

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

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

                c = wsprintf(szForm,
                                "<HTML><HEAD><TITLE>TPC-C Delivery</TITLE></HEAD><BODY>"
                                "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
                                "<PRE><font face=\"Courier\">                                   Delivery<BR>"
                                "Warehouse: %6.6d<BR> <BR>",
                                (!bInput && (pDeliveryData->exec_status_code != eOK)) ? ERR_TYPE_DELIVERY_POST : 0,
                                DELIVERY_FORM, iTermId, Term.pClientData[iTermId].iSyncId, Term.pClientData[iTermId].w_id);

                if ( bInput )
                {
                                strcpy( szForm+c,
                                                "Carrier Number: <INPUT NAME=\"OCD\" SIZE=1><BR> <BR>"
                                                "Execution Status: <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"
                                                " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <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_DATApDelivery;

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

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

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

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

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

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

                        PSTOCK_LEVEL_DATA               pStockLevel;

                        pStockLevel = Term.pClientData[iTermId].pTxn->BuffAddr_StockLevel();
                        ZeroMemory( pStockLevel, sizeof(STOCK_LEVEL_DATA) );

                        pStockLevel->w_id = Term.pClientData[iTermId].w_id;
                        pStockLevel->d_id = Term.pClientData[iTermId].d_id;
```

```
		pStockLevel->threshold = GetIntKeyValue(&ptr, "TT*", ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY, ERR_STOCKLEVEL_THRESHOLD_INVALID);
		if ( pStockLevel->threshold >= 100 || pStockLevel->threshold < 0 )
			throw new CWEBCLNT_ERR( ERR_STOCKLEVEL_THRESHOLD_RANGE );

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

		pStockLevel = Term.pClientData[iTermId].pTxn->BuffAddr_StockLevel();
		MakeStockLevelForm(iTermId, pStockLevel, OUTPUT_FORM, szBuffer);
}

/* FUNCTION: GetNewOrderData
 *
 * PURPOSE:	This function extracts and validates the new order form data from an http command string.
 *
 * ARGUMENTS:	LPSTR		lpszQueryString		client browser http command string
 *		NEW_ORDER_DATA	*pNewOrderData		pointer to new order data structure
 *
 */

void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA *pNewOrderData)
{
		char		szTmp[26];
		int			i;
		short		items;
		int			ol_i_id, ol_quantity;
		char		*ptr = lpszQueryString;

		static char szSP[MAX_OL_NEW_ORDER_ITEMS][6] =
			{ "SP00*", "SP01*", "SP02*", "SP03*", "SP04*",
			  "SP05*", "SP06*", "SP07*", "SP08*", "SP09*",
			  "SP10*", "SP11*", "SP12*", "SP13*", "SP14*" };
		static char szIID[MAX_OL_NEW_ORDER_ITEMS][7] =
			{ "IID00*", "IID01*", "IID02*", "IID03*", "IID04*",
			  "IID05*", "IID06*", "IID07*", "IID08*", "IID09*",
			  "IID10*", "IID11*", "IID12*", "IID13*", "IID14*" };
		static char szQty[MAX_OL_NEW_ORDER_ITEMS][7] =
			{ "Qty00*", "Qty01*", "Qty02*", "Qty03*", "Qty04*",
			  "Qty05*", "Qty06*", "Qty07*", "Qty08*", "Qty09*",
			  "Qty10*", "Qty11*", "Qty12*", "Qty13*", "Qty14*" };

		pNewOrderData->d_id = GetIntKeyValue(&ptr, "DID*", ERR_NEWORDER_FORM_MISSING_DID, ERR_NEWORDER_DISTRICT_INVALID);
		pNewOrderData->c_id = GetIntKeyValue(&ptr, "CID*", ERR_NEWORDER_CUSTOMER_KEY, ERR_NEWORDER_CUSTOMER_INVALID);

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

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

				ol_quantity = pNewOrderData->OL[items].ol_quantity =
					GetIntKeyValue(&ptr, szQty[i], ERR_NEWORDER_MISSING_QTY_KEY, ERR_NEWORDER_QTY_INVALID);
				if ( ol_quantity > 99 || ol_quantity < 1 )
					throw new CWEBCLNT_ERR( ERR_NEWORDER_QTY_RANGE );

				items++;
			}
			else
			{	// nothing entered for supply warehouse, so item id and qty must also be blank
				GetKeyValue(&ptr, szIID[i], szTmp, sizeof(szTmp), ERR_NEWORDER_MISSING_IID_KEY);
				if ( szTmp[0] )
					throw new CWEBCLNT_ERR( ERR_NEWORDER_ITEMID_WITHOUT_SUPPW );

				GetKeyValue(&ptr, szQty[i], szTmp, sizeof(szTmp), ERR_NEWORDER_MISSING_QTY_KEY);
				if ( szTmp[0] )
					throw new CWEBCLNT_ERR( ERR_NEWORDER_QTY_WITHOUT_SUPPW );
			}
		}
		if ( items == 0 )
			throw new CWEBCLNT_ERR( ERR_NEWORDER_NOITEMS_ENTERED );

		pNewOrderData->o_ol_cnt = items;
}

/* FUNCTION: GetPaymentData
 *
 * PURPOSE:	This function extracts and validates the payment form data from an http command string.
 *
 * ARGUMENTS:	LPSTR		lpszQueryString		client browser http command string
 *		PAYMENT_DATA	*pPaymentData		pointer to payment data structure
 */

void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA *pPaymentData)
{
		char		szTmp[26];
		char		*ptr = lpszQueryString;
		BOOL		bCustIdBlank;

		pPaymentData->d_id = GetIntKeyValue(&ptr, "DID*", ERR_PAYMENT_MISSING_DID_KEY, ERR_PAYMENT_DISTRICT_INVALID);

		GetKeyValue(&ptr, "CID*", szTmp, sizeof(szTmp), ERR_PAYMENT_MISSING_CID_KEY);
		if ( szTmp[0] == 0 )
		{
			bCustIdBlank = TRUE;
			pPaymentData->c_id = 0;
		}
		else
		{	// parse customer id and verify that last name was NOT entered
			bCustIdBlank = FALSE;
			if ( !IsNumeric(szTmp) )
				throw new CWEBCLNT_ERR( ERR_PAYMENT_CUSTOMER_INVALID );
			pPaymentData->c_id = atoi(szTmp);
		}

		pPaymentData->c_w_id = GetIntKeyValue(&ptr, "CWI*", ERR_PAYMENT_MISSING_CWI_KEY, ERR_PAYMENT_CWI_INVALID);
		pPaymentData->c_d_id = GetIntKeyValue(&ptr, "CDI*", ERR_PAYMENT_MISSING_CDI_KEY, ERR_PAYMENT_CDI_INVALID);

		if ( bCustIdBlank )
		{	// customer id is blank, so last name must be entered
			GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp), ERR_PAYMENT_MISSING_CLT_KEY);
			if ( szTmp[0] == 0 )
				throw new CWEBCLNT_ERR( ERR_PAYMENT_MISSING_CID_CLT );

			_strupr( szTmp );
			if ( strlen(pPaymentData->c_last) > LAST_NAME_LEN )
				throw new CWEBCLNT_ERR( ERR_PAYMENT_LAST_NAME_TO_LONG );
			strcpy(pPaymentData->c_last, szTmp);
		}
		else
		{	// parse customer id and verify that last name was NOT entered
			GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp), ERR_PAYMENT_MISSING_CLT_KEY);
			if ( szTmp[0] != 0 )
				throw new CWEBCLNT_ERR( ERR_PAYMENT_CID_AND_CLT );
		}

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

/* FUNCTION: GetOrderStatusData
 *
 * PURPOSE:	This function extracts and validates the payment form data from an http command string.
 *
 */
void GetOrderStatusData(LPSTR lpszQueryString, ORDER_STATUS_DATA *pOrderStatusData)
{
		char		szTmp[26];
		char		*ptr = lpszQueryString;

		pOrderStatusData->d_id = GetIntKeyValue(&ptr, "DID*", ERR_ORDERSTATUS_MISSING_DID_KEY, ERR_ORDERSTATUS_DID_INVALID);
```

```
                GetKeyValue(&ptr, "CID*", szTmp, sizeof(szTmp), ERR_ORDERSTATUS_MISSING_CID_KEY);
                if ( szTmp[0] == 0 )
                {
                        // customer id is blank, so last name must be entered
                        pOrderStatusData->c_id = 0;
                        GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp), ERR_ORDERSTATUS_MISSING_CLT_KEY);
                        if ( szTmp[0] == 0 )
                                throw new CWEBCLNT_ERR( ERR_ORDERSTATUS_MISSING_CID_CLT );

                        _strupr( szTmp );
                        if ( strlen(pOrderStatusData->c_last) > LAST_NAME_LEN )
                                throw new CWEBCLNT_ERR( ERR_ORDERSTATUS_CLT_RANGE );
                        strcpy(pOrderStatusData->c_last, szTmp);
                }
                else
                {       // parse customer id and verify that last name was NOT entered
                        if ( !IsNumeric(szTmp) )
                                throw new CWEBCLNT_ERR( ERR_ORDERSTATUS_CID_INVALID );
                        pOrderStatusData->c_id = atoi(szTmp);
                        GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp), ERR_ORDERSTATUS_MISSING_CLT_KEY);
                        if ( szTmp[0] != 0 )
                                throw new CWEBCLNT_ERR( ERR_ORDERSTATUS_CID_AND_CLT );
                }

}

/* FUNCTION: BOOL IsNumeric(char *ptr)
 *
 * PURPOSE:      This function determines if a string is numeric. It fails if any characters other
 *                              than numeric and null terminator are present.
 *
 * ARGUMENTS:   char                                    *ptr            pointer to string to check.
 *
 * RETURNS:                     BOOL            FALSE           if string is not all numeric
 *                                                                                      TRUE            if string contains only numeric characters i.e. '0' - '9'
 */

BOOL IsNumeric(char *ptr)
{
                if ( *ptr == 0 )
                        return FALSE;

                while( *ptr && isdigit(*ptr) )
                        ptr++;
                return ( !*ptr );
}

/* FUNCTION: BOOL IsDecimal(char *ptr)
 *
 * PURPOSE:      This function determines if a string is a non-negative decimal value.
 *      It fails if any characters other than a series of numbers followed by
 *                              a decimal point, another series of numbers, and a null terminator are present.
 *
 *
 * ARGUMENTS:   char                                    *ptr            pointer to string to check.
 *
 * RETURNS:                     BOOL            FALSE           if string is not a valid non-negative decimal value
 *                                                                                      TRUE            if string is OK
 */

BOOL IsDecimal(char *ptr)
{
                char *dotptr;
                BOOL  bValid;

                if ( *ptr == 0 )
                        return FALSE;

                // find decimal point
                dotptr = strchr( ptr, '.' );
                if (dotptr == NULL)
                        // no decimal point, so just check for numeric
                        return IsNumeric(ptr);
                *dotptr = 0;  // temporarily replace decimal with a terminator

                if ( *ptr != 0 )
                        bValid = IsNumeric(ptr);
                // string starts with decimal point
                else if (*(dotptr+1) == 0)
                        return FALSE;  // nothing but a decimal point is bad
                else
                        bValid = TRUE;

                if (*(dotptr+1) != 0)
                        // check text after decimal point
                        bValid &= IsNumeric(dotptr+1);

                *dotptr = '.';  // replace decimal point
                return bValid;
}
```

Kit MSTPCC451\WEBCLNT\isapi_dll\src\tpcc.def

```
LIBRARY TPCC.DLL

EXPORTS

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

Kit MSTPCC451\WEBCLNT\isapi_dll\src\tpcc.h

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

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

#define TP_MAX_RETRIES                                                                  50


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

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


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

                int                                             iSyncId;                                //syncronization id
                int                                             iTickCount;                             //time of last access;

                CTPCC_BASE      *pTxn;

} CLIENTDATA, *PCLIENTDATA;


//This structure is used to define the operational interface for terminal id support
```

```
typedef struct _TERM
{
        int             iNumEntries;            //total allocated terminal array entries
        int             iFreeList;              //next available terminal array element or -1 if none
        int             iMasterSyncId;          //syncronization id
        CLIENTDATA      *pClientData;           //pointer to allocated client data
} TERM;

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


        enum WEBERROR
        {
                NO_ERR,
                ERR_COMMAND_UNDEFINED,
                ERR_D_ID_INVALID,
                ERR_DELIVERY_CARRIER_ID_RANGE,
                ERR_DELIVERY_CARRIER_INVALID,
                ERR_DELIVERY_MISSING_OCD_KEY,
                ERR_DELIVERY_THREAD_FAILED,
                ERR_GETPROCADDR_FAILED,
                ERR_HTML_ILL_FORMED,
                ERR_INVALID_SYNC_CONNECTION,
                ERR_INVALID_TERMID,
                ERR_LOADDLL_FAILED,
                ERR_MAX_CONNECTIONS_EXCEEDED,
                ERR_MEM_ALLOC_FAILED,
                ERR_MISSING_REGISTRY_ENTRIES,
                ERR_NEWORDER_CUSTOMER_INVALID,
                ERR_NEWORDER_CUSTOMER_KEY,
                ERR_NEWORDER_DISTRICT_INVALID,
                ERR_NEWORDER_FORM_MISSING_DID,
                ERR_NEWORDER_ITEMID_INVALID,
                ERR_NEWORDER_ITEMID_RANGE,
                ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
                ERR_NEWORDER_MISSING_IID_KEY,
                ERR_NEWORDER_MISSING_QTY_KEY,
                ERR_NEWORDER_MISSING_SUPPW_KEY,
                ERR_NEWORDER_NOITEMS_ENTERED,
                ERR_NEWORDER_QTY_INVALID,
                ERR_NEWORDER_QTY_RANGE,
                ERR_NEWORDER_QTY_WITHOUT_SUPPW,
                ERR_NEWORDER_SUPPW_INVALID,
                ERR_NO_SERVER_SPECIFIED,
                ERR_ORDERSTATUS_CID_AND_CLT,
                ERR_ORDERSTATUS_CID_INVALID,
                ERR_ORDERSTATUS_CLT_RANGE,
                ERR_ORDERSTATUS_DID_INVALID,
                ERR_ORDERSTATUS_MISSING_CID_CLT,
                ERR_ORDERSTATUS_MISSING_CID_KEY,
                ERR_ORDERSTATUS_MISSING_CLT_KEY,
                ERR_ORDERSTATUS_MISSING_DID_KEY,
                ERR_PAYMENT_CDI_INVALID,
                ERR_PAYMENT_CID_AND_CLT,
                ERR_PAYMENT_CUSTOMER_INVALID,
                ERR_PAYMENT_CWI_INVALID,
                ERR_PAYMENT_DISTRICT_INVALID,
                ERR_PAYMENT_HAM_INVALID,
                ERR_PAYMENT_HAM_RANGE,
                ERR_PAYMENT_LAST_NAME_TO_LONG,
                ERR_PAYMENT_MISSING_CDI_KEY,
                ERR_PAYMENT_MISSING_CID_CLT,
                ERR_PAYMENT_MISSING_CID_KEY,
                ERR_PAYMENT_MISSING_CLT,
                ERR_PAYMENT_MISSING_CLT_KEY,
                ERR_PAYMENT_MISSING_CWI_KEY,
                ERR_PAYMENT_MISSING_DID_KEY,
                ERR_PAYMENT_MISSING_HAM_KEY,
                ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
                ERR_STOCKLEVEL_THRESHOLD_INVALID,
                ERR_STOCKLEVEL_THRESHOLD_RANGE,
                ERR_VERSION_MISMATCH,
                ERR_W_ID_INVALID
        };

class CWEBCLNT_ERR : public CBaseErr
{
        public:

                CWEBCLNT_ERR(WEBERROR Err)
                {
                        m_Error = Err;
                        m_szTextDetail = NULL;
                        m_SystemErr = 0;
                        m_szErrorText = NULL;
                };

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

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

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

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

};

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

//function prototypes

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call, LPVOID lpReserved);
void WriteMessageToEventLog(LPTSTR lpszMsg);
void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd, int *pFormId, int *pTermId, int *pSyncId);
void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId);
void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId);
void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int iError, int iErrorType, char *szMsg, int iTermId);
void GetKeyValue(char **pQueryString, char *pKey, char *pValue, int iMax, WEBERROR err);
int  GetIntKeyValue(char **pQueryString, char *pKey, WEBERROR NoKeyErr, WEBERROR NotIntErr);
void TermInit(void);
void TermDeleteAll(void);
int  TermAdd(void);
void TermDelete(int id);
void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int iErrorNum, int iTermId, int iSyncId, char *szErrorText, char *szBuffer );
void MakeMainMenuForm(int iTermId, int iSyncId, char *szForm);
void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA *pStockLevelData, BOOL bInput, char *szForm);
void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA *pNewOrderData, BOOL bInput, char *szForm);
void MakePaymentForm(int iTermId, PAYMENT_DATA *pPaymentData, BOOL bInput, char *szForm);
void MakeOrderStatusForm(int iTermId, ORDER_STATUS_DATA *pOrderStatusData, BOOL bInput, char *szForm);
void MakeDeliveryForm(int iTermId, DELIVERY_DATA *pDeliveryData, BOOL bInput, char *szForm);
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK                *pECB, int iTermId, char *szBuffer);
void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char *szBuffer);
void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char *szBuffer);
void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char *szBuffer);
void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char *szBuffer);
void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA *pNewOrderData);
void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA *pPaymentData);
```

```
void GetOrderStatusData(LPSTR lpszQueryString, ORDER_STATUS_DATA *pOrderStatusData);
BOOL PostDeliveryInfo(long w_id, short o_carrier_id);
BOOL IsNumeric(char *ptr);
BOOL IsDecimal(char *ptr);
void DeliveryWorkerThread(void *ptr);
```

Kit MSTPCC451\WEBCLNT\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
```

Kit MSTPCC451\WEBCLNT\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 );
}
```

Kit MSTPCC451\WEBCLNT\tm_com_dll\src\tpcc_com.h

```
/*		FILE:			TPCC_COM.H
 *
 *
 *	All Rights Reserved
 *
 *
 *
 *		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);
```

Kit MSTPCC451\WEBCLNT\tpcc_com_all\src\Methods.h

```
/*                      FILE:                   METHODS.H
 *                                                      Microsoft TPC-C Kit Ver. 4.20.000
 *                                                      Copyright Microsoft, 1999
 *          All Rights Reserved
 *
 *                                                      not yet audited
 *
 *                      PURPOSE:        Header file for COM components.
 *
 *  Change history:
 *                      4.20.000 - first version
 */


enum COMPONENT_ERROR
{
                        ERR_MISSING_REGISTRY_ENTRIES = 1,
                        ERR_LOADDLL_FAILED,
                        ERR_GETPROCADDR_FAILED,
                        ERR_UNKNOWN_DB_PROTOCOL
};

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

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

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

                                        COMPONENT_ERROR         m_Error;
                                        char                                            *m_szTextDetail;
                                        char                                            *m_szErrorText;
                                        DWORD                                           m_SystemErr;

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


static void WriteMessageToEventLog(LPTSTR lpszMsg);


/////////////////////////////////////////////////////////////////////////////
// CTPCC_Common
class CTPCC_Common :
                        public ITPCC,
                        public IObjectControl,
                        public IObjectConstruct,
                        public CComObjectRootEx<CComSingleThreadModel>
{
public:
BEGIN_COM_MAP(CTPCC_Common)
                        COM_INTERFACE_ENTRY(ITPCC)
                        COM_INTERFACE_ENTRY(IObjectControl)
                        COM_INTERFACE_ENTRY(IObjectConstruct)
END_COM_MAP()

                        CTPCC_Common();
                        ~CTPCC_Common();

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

                        HRESULT __stdcall CallSetComplete();

// IObjectControl
                        STDMETHODIMP_(BOOL) CanBePooled() { return m_bCanBePooled; }
                        STDMETHODIMP Activate() { return S_OK; }                // we don't support COM Services transactions (no enlistment)
                        STDMETHODIMP_(void) Deactivate() { /* nothing to do */ }

// IObjectConstruct
                        STDMETHODIMP Construct(IDispatch * pUnk);

                        // helper methods
private:
                        BOOL                                    m_bCanBePooled;
                        CTPCC_BASE      *m_pTxn;

                        struct COM_DATA
                        {
                                        int retval;
                                        int error;
                                        union
                                        {
                                                        NEW_ORDER_DATA                                  NewOrder;
                                                        PAYMENT_DATA                    Payment;
                                                        DELIVERY_DATA                   Delivery;
                                                        STOCK_LEVEL_DATA                StockLevel;
                                                        ORDER_STATUS_DATA               OrderStatus;
                                        } u;
                        };

};

/////////////////////////////////////////////////////////////////////////////
// CTPCC
class CTPCC :
                        public CTPCC_Common,
                        public CComCoClass<CTPCC, &CLSID_TPCC>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_TPCC)
```

```
BEGIN_COM_MAP(CTPCC)
                        //COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx<CComSingleThreadModel>)
                        COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
                        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

};


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

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

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

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

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

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


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

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

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


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

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

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

Kit MSTPCC451\WEBCLNT\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
```

Kit MSTPCC451\WEBCLNT\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, Reg.szSPPrefix );
                        else if (Reg.eDB_Protocol == DBLIB)
                                m_pTxn = pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, szMyComputerName, Reg.szDbName );
                }
                catch (CBaseErr *e)
                {
                        WriteMessageToEventLog(e->ErrorText());
                        delete e;
                        return E_FAIL;
                }
                catch (...)
                {
                        WriteMessageToEventLog(TEXT("Unhandled exception in object ::Construct"));
                        return E_FAIL;
                }

                return S_OK;
}

HRESULT CTPCC_Common::NewOrder(VARIANT txn_in, VARIANT* txn_out)
{
                PNEW_ORDER_DATA                 pNewOrder;
                COM_DATA                        *pData;
                try
                {
                        pData = (COM_DATA*)txn_in.parray->pvData;
                        pNewOrder = m_pTxn->BuffAddr_NewOrder();
```

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

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

                VariantInit(txn_out);
                txn_out->vt = VT_SAFEARRAY;
                txn_out->parray = SafeArrayCreateVector(VT_UI1,
                                                                        txn_in.parray->rgsabound->cElements,
                                                                        txn_in.parray->rgsabound->cElements);
                pData = (COM_DATA*) txn_out->parray->pvData;

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

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

                pData->retval = e->ErrorType();
                pData->error = e->ErrorNum();
                delete e;
                return E_FAIL;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception."));
                pData->retval = ERR_TYPE_LOGIC;
                pData->error = 0;
                m_bCanBePooled = FALSE;
                return E_FAIL;
        }
}

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

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

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

                VariantInit(txn_out);
                txn_out->vt = VT_SAFEARRAY;
                txn_out->parray = SafeArrayCreateVector( VT_UI1,
                                                                        txn_in.parray->rgsabound->cElements,
                                                                        txn_in.parray->rgsabound->cElements);
                pData = (COM_DATA*) txn_out->parray->pvData;

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

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

                pData->retval = e->ErrorType();
                pData->error = e->ErrorNum();
                delete e;
                return E_FAIL;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception."));
                pData->retval = ERR_TYPE_LOGIC;
                pData->error = 0;
                m_bCanBePooled = FALSE;
                return E_FAIL;
        }
}

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

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

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

                m_pTxn->StockLevel();

                VariantInit(txn_out);
                txn_out->vt = VT_SAFEARRAY;
                txn_out->parray = SafeArrayCreateVector( VT_UI1,
                                                                        txn_in.parray->rgsabound->cElements,
                                                                        txn_in.parray->rgsabound->cElements);
                pData = (COM_DATA*)txn_out->parray->pvData;

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

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

                pData->retval = e->ErrorType();
                pData->error = e->ErrorNum();
                delete e;
                return E_FAIL;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception."));
                pData->retval = ERR_TYPE_LOGIC;
                pData->error = 0;
                m_bCanBePooled = FALSE;
                return E_FAIL;
        }
}

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

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

                m_pTxn->OrderStatus();

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

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

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

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

                        pData->retval = e->ErrorType();
                        pData->error = e->ErrorNum();
                        delete e;
                        return E_FAIL;
                }
                catch (...)
                {
                        WriteMessageToEventLog(TEXT("Unhandled exception."));
                        pData->retval = ERR_TYPE_LOGIC;
                        pData->error = 0;
                        m_bCanBePooled = FALSE;
                        return E_FAIL;
                }
}
```

Kit MSTPCC451\WEBCLNT\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
```

Kit MSTPCC451\WEBCLNT\tpcc_com_all\src\tpcc_com_all.h

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

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


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


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

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

#ifndef __tpcc_com_all_h__
#define __tpcc_com_all_h__

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

/* Forward Declarations */

#ifndef __TPCC_FWD_DEFINED__
#define __TPCC_FWD_DEFINED__

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

#endif                  /* __TPCC_FWD_DEFINED__ */


#ifndef __NewOrder_FWD_DEFINED__
#define __NewOrder_FWD_DEFINED__

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

#endif                  /* __NewOrder_FWD_DEFINED__ */


#ifndef __OrderStatus_FWD_DEFINED__
#define __OrderStatus_FWD_DEFINED__

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

#endif                  /* __OrderStatus_FWD_DEFINED__ */


#ifndef __Payment_FWD_DEFINED__
#define __Payment_FWD_DEFINED__

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

#endif                  /* __Payment_FWD_DEFINED__ */


#ifndef __StockLevel_FWD_DEFINED__
#define __StockLevel_FWD_DEFINED__

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

#endif                  /* __StockLevel_FWD_DEFINED__ */


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

#ifdef __cplusplus
extern "C"{
#endif
```

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

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




extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_s_ifspec;


#ifndef __TPCCLib_LIBRARY_DEFINED__
#define __TPCCLib_LIBRARY_DEFINED__

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


EXTERN_C const IID LIBID_TPCCLib;

EXTERN_C const CLSID CLSID_TPCC;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_NewOrder;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_OrderStatus;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_Payment;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_StockLevel;

#ifdef __cplusplus

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

/* Additional Prototypes for ALL interfaces */

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

Kit MSTPCC451\WEBCLNT\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;
        };

};
```

Kit MSTPCC451\WEBCLNT\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
```

Kit MSTPCC451\WEBCLNT\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'
                                }
                        }
                }
}
```

Kit MSTPCC451\WEBCLNT\tpcc_com_all\src\tpcc_com_all_i.c

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

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

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


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

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

#ifdef __cplusplus
extern "C"{
#endif


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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID, LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID, CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


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


MIDL_DEFINE_GUID(CLSID, CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);


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


MIDL_DEFINE_GUID(CLSID, CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif


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

Kit MSTPCC451\WEBCLNT\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'
                                }
                        }
                }
}
```

Kit MSTPCC451\WEBCLNT\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'
				}
			}
		}
}
```

Kit MSTPCC451\WEBCLNT\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'
				}
			}
		}
}
```

Kit MSTPCC451\WEBCLNT\tpcc_com_all\src\tpcc_com_ps.h

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

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


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


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

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

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

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

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

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

/* Forward Declarations */

#ifndef __ITPCC_FWD_DEFINED__
#define __ITPCC_FWD_DEFINED__
typedef interface ITPCC ITPCC;
#endif 		/* __ITPCC_FWD_DEFINED__ */


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

#ifdef __cplusplus
extern "C"{
#endif

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

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



extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

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


EXTERN_C const IID IID_ITPCC;

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

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

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

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

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

        virtual HRESULT __stdcall OrderStatus(
            /* [in] */ VARIANT txn_in,
```

```
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall CallSetComplete( void) = 0;

};

#else                   /* C style interface */

typedef struct ITPCCVtbl
{
    BEGIN_INTERFACE

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

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

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

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

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

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

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

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

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

    END_INTERFACE
} ITPCCVtbl;

interface ITPCC
{
    CONST_VTBL struct ITPCCVtbl *lpVtbl;
};


#ifdef COBJMACROS


#define ITPCC_QueryInterface(This,riid,ppvObject)           \
    (This)->lpVtbl -> QueryInterface(This,riid,ppvObject)

#define ITPCC_AddRef(This)              \
    (This)->lpVtbl -> AddRef(This)

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


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

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

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

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

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

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

#endif /* COBJMACROS */


#endif                  /* C style interface */


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


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


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


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


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


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


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


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


HRESULT __stdcall ITPCC_OrderStatus_Proxy(
```

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


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


HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC * This);


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


#endif                  /* __ITPCC_INTERFACE_DEFINED__ */


/* Additional Prototypes for ALL interfaces */

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

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

Kit MSTPCC451\WEBCLNT\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'
                                                }
                                }
                }
}
```

Kit MSTPCC451\WEBCLNT\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 */
```

Kit MSTPCC451\WEBCLNT\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
```

Kit MSTPCC451\WEBCLNT\tpcc_com_ps\src\tpcc_com_ps.h

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

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


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


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

#include "rpc.h"
```

```
#include "rpcndr.h"

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

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

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

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

/* Forward Declarations */

#ifndef __ITPCC_FWD_DEFINED__
#define __ITPCC_FWD_DEFINED__
typedef interface ITPCC ITPCC;
#endif                  /* __ITPCC_FWD_DEFINED__ */


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

#ifdef __cplusplus
extern "C"{
#endif

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

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



extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

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


EXTERN_C const IID IID_ITPCC;

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

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

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

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

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

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

        virtual HRESULT __stdcall CallSetComplete( void) = 0;

    };

#else                   /* C style interface */

    typedef struct ITPCCVtbl
    {
        BEGIN_INTERFACE

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

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

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

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

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

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

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

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

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

        END_INTERFACE
    } ITPCCVtbl;

    interface ITPCC
    {
        CONST_VTBL struct ITPCCVtbl *lpVtbl;
    };


#ifdef COBJMACROS


#define ITPCC_QueryInterface(This,riid,ppvObject)               \
    (This)->lpVtbl -> QueryInterface(This,riid,ppvObject)

#define ITPCC_AddRef(This)              \
    (This)->lpVtbl -> AddRef(This)

#define ITPCC_Release(This)             \
    (This)->lpVtbl -> Release(This)
```

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

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

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

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

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

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

#endif /* COBJMACROS */


#endif                  /* C style interface */


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


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


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


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


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


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


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


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


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


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


HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC * This);


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


#endif                  /* __ITPCC_INTERFACE_DEFINED__ */


/* Additional Prototypes for ALL interfaces */

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

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

Kit MSTPCC451\WEBCLNT\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
```

Kit MSTPCC451\WEBCLNT\tpcc_com_ps\src\tpcc_com_ps_i.c

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

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

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


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

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

#ifdef __cplusplus
extern "C"{
#endif


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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID, IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif


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

Kit MSTPCC451\WEBCLNT\tpcc_com_ps\src\tpcc_com_ps_p.c

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

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


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

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


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

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


#include "tpcc_com_ps.h"

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

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

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


static RPC_SYNTAX_IDENTIFIER  _RpcTransferSyntax =
{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,0x00,0x2B,0x10,0x48,0x60}},{2,0}};


extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;


extern const MIDL_STUB_DESC Object_StubDesc;


extern const MIDL_SERVER_INFO ITPCC_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo;


extern const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

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

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this stub because it uses these features:
#error   -Oif or -Oicf, [wire_marshal] or [user_marshal] attribute.
#error However, your C/C++ compilation flags indicate you intend to run this app on earlier systems.
#error This app will die there with the RPC_X_WRONG_STUB_VERSION error.
#endif


static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
    {
        0,
        {

                    /* Procedure NewOrder */

                                                        0x33,                   /* FC_AUTO_HANDLE */
                                                        0x6c,                   /* Old Flags:  object, Oi2 */
/*  2 */            NdrFcLong( 0x0 ),   /* 0 */
/*  6 */            NdrFcShort( 0x3 ),  /* 3 */
/*  8 */            NdrFcShort( 0x1c ),/* x86 Stack size/offset = 28 */
/* 10 */            NdrFcShort( 0x0 ),  /* 0 */
/* 12 */            NdrFcShort( 0x8 ),  /* 8 */
/* 14 */            0x7,                    /* Oi2 Flags:  srv must size, clt must size, has return, */
                                                        0x3,                    /* 3 */

                    /* Parameter txn_in */

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

                    /* Parameter txn_out */

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

                    /* Return value */

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

                    /* Procedure Payment */

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

                    /* Parameter txn_in */

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

                    /* Parameter txn_out */

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

                    /* Return value */

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

                    /* Procedure Delivery */

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

                    /* Parameter txn_in */

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

                    /* Parameter txn_out */

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

                    /* Return value */

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

```
/* 100 */      0x8,                                    /* FC_LONG */
                                                       0x0,                                    /* 0 */

               /* Procedure StockLevel */

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

               /* Parameter txn_in */

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

               /* Parameter txn_out */

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

               /* Return value */

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

               /* Procedure OrderStatus */

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

               /* Parameter txn_in */

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

               /* Parameter txn_out */

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

               /* Return value */

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

               /* Procedure CallSetComplete */

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

               /* Return value */

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

                                                       0x0

    }
  };

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

/*  2 */
                                                       0x12, 0x0,              /* FC_UP */
/*  4 */       NdrFcShort( 0x3ca ),                    /* Offset= 970 (974) */
/*  6 */
                                                       0x2b,                                   /* FC_NON_ENCAPSULATED_UNION */
                                                       0x9,                                    /* FC_ULONG */
/*  8 */       0x7,                                    /* Corr desc: FC_USHORT */
                                                       0x0,                                    /*  */
/* 10 */       NdrFcShort( 0xfff8 ),                   /* -8 */
/* 12 */       NdrFcShort( 0x2 ),  /* Offset= 2 (14) */
/* 14 */       NdrFcShort( 0x10 ),/* 16 */
/* 16 */       NdrFcShort( 0x2f ),  /* 47 */
/* 18 */       NdrFcLong( 0x14 ),  /* 20 */
/* 22 */       NdrFcShort( 0x800b ),                   /* Simple arm type: FC_HYPER */
/* 24 */       NdrFcLong( 0x3 ),   /* 3 */
/* 28 */       NdrFcShort( 0x8008 ),                   /* Simple arm type: FC_LONG */
/* 30 */       NdrFcLong( 0x11 ),  /* 17 */
/* 34 */       NdrFcShort( 0x8001 ),                   /* Simple arm type: FC_BYTE */
/* 36 */       NdrFcLong( 0x2 ),   /* 2 */
/* 40 */       NdrFcShort( 0x8006 ),                   /* Simple arm type: FC_SHORT */
/* 42 */       NdrFcLong( 0x4 ),   /* 4 */
/* 46 */       NdrFcShort( 0x800a ),                   /* Simple arm type: FC_FLOAT */
/* 48 */       NdrFcLong( 0x5 ),   /* 5 */
/* 52 */       NdrFcShort( 0x800c ),                   /* Simple arm type: FC_DOUBLE */
/* 54 */       NdrFcLong( 0xb ),   /* 11 */
/* 58 */       NdrFcShort( 0x8006 ),                   /* Simple arm type: FC_SHORT */
/* 60 */       NdrFcLong( 0xa ),   /* 10 */
/* 64 */       NdrFcShort( 0x8008 ),                   /* Simple arm type: FC_LONG */
/* 66 */       NdrFcLong( 0x6 ),   /* 6 */
/* 70 */       NdrFcShort( 0xe8 ),/* Offset= 232 (302) */
/* 72 */       NdrFcLong( 0x7 ),   /* 7 */
/* 76 */       NdrFcShort( 0x800c ),                   /* Simple arm type: FC_DOUBLE */
/* 78 */       NdrFcLong( 0x8 ),   /* 8 */
/* 82 */       NdrFcShort( 0xe2 ),/* Offset= 226 (308) */
/* 84 */       NdrFcLong( 0xd ),   /* 13 */
/* 88 */       NdrFcShort( 0xf4 ),/* Offset= 244 (332) */
/* 90 */       NdrFcLong( 0x9 ),   /* 9 */
/* 94 */       NdrFcShort( 0x100 ),                    /* Offset= 256 (350) */
/* 96 */       NdrFcLong( 0x2000 ),                    /* 8192 */
/* 100 */      NdrFcShort( 0x10c ),                    /* Offset= 268 (368) */
/* 102 */      NdrFcLong( 0x24 ),  /* 36 */
/* 106 */      NdrFcShort( 0x31a ),                    /* Offset= 794 (900) */
/* 108 */      NdrFcLong( 0x4024 ),                    /* 16420 */
/* 112 */      NdrFcShort( 0x314 ),                    /* Offset= 788 (900) */
/* 114 */      NdrFcLong( 0x4011 ),                    /* 16401 */
/* 118 */      NdrFcShort( 0x312 ),                    /* Offset= 786 (904) */
/* 120 */      NdrFcLong( 0x4002 ),                    /* 16386 */
/* 124 */      NdrFcShort( 0x310 ),                    /* Offset= 784 (908) */
/* 126 */      NdrFcLong( 0x4003 ),                    /* 16387 */
/* 130 */      NdrFcShort( 0x30e ),                    /* Offset= 782 (912) */
/* 132 */      NdrFcLong( 0x4014 ),                    /* 16404 */
/* 136 */      NdrFcShort( 0x30c ),                    /* Offset= 780 (916) */
/* 138 */      NdrFcLong( 0x4004 ),                    /* 16388 */
/* 142 */      NdrFcShort( 0x30a ),                    /* Offset= 778 (920) */
/* 144 */      NdrFcLong( 0x4005 ),                    /* 16389 */
/* 148 */      NdrFcShort( 0x308 ),                    /* Offset= 776 (924) */
/* 150 */      NdrFcLong( 0x400b ),                    /* 16395 */
/* 154 */      NdrFcShort( 0x2f2 ),                    /* Offset= 754 (908) */
/* 156 */      NdrFcLong( 0x400a ),                    /* 16394 */
```

```
/* 160 */      NdrFcShort( 0x2f0 ),                /* Offset= 752 (912) */
/* 162 */      NdrFcLong( 0x4006 ),                /* 16390 */
/* 166 */      NdrFcShort( 0x2fa ),                /* Offset= 762 (928) */
/* 168 */      NdrFcLong( 0x4007 ),                /* 16391 */
/* 172 */      NdrFcShort( 0x2f0 ),                /* Offset= 752 (924) */
/* 174 */      NdrFcLong( 0x4008 ),                /* 16392 */
/* 178 */      NdrFcShort( 0x2f2 ),                /* Offset= 754 (932) */
/* 180 */      NdrFcLong( 0x400d ),                /* 16397 */
/* 184 */      NdrFcShort( 0x2f0 ),                /* Offset= 752 (936) */
/* 186 */      NdrFcLong( 0x4009 ),                /* 16393 */
/* 190 */      NdrFcShort( 0x2ee ),                /* Offset= 750 (940) */
/* 192 */      NdrFcLong( 0x6000 ),                /* 24576 */
/* 196 */      NdrFcShort( 0x2ec ),                /* Offset= 748 (944) */
/* 198 */      NdrFcLong( 0x400c ),                /* 16396 */
/* 202 */      NdrFcShort( 0x2ea ),                /* Offset= 746 (948) */
/* 204 */      NdrFcLong( 0x10 ), /* 16 */
/* 208 */      NdrFcShort( 0x8002 ),               /* Simple arm type: FC_CHAR */
/* 210 */      NdrFcLong( 0x12 ), /* 18 */
/* 214 */      NdrFcShort( 0x8006 ),               /* Simple arm type: FC_SHORT */
/* 216 */      NdrFcLong( 0x13 ), /* 19 */
/* 220 */      NdrFcShort( 0x8008 ),               /* Simple arm type: FC_LONG */
/* 222 */      NdrFcLong( 0x15 ), /* 21 */
/* 226 */      NdrFcShort( 0x800b ),               /* Simple arm type: FC_HYPER */
/* 228 */      NdrFcLong( 0x16 ), /* 22 */
/* 232 */      NdrFcShort( 0x8008 ),               /* Simple arm type: FC_LONG */
/* 234 */      NdrFcLong( 0x17 ), /* 23 */
/* 238 */      NdrFcShort( 0x8008 ),               /* Simple arm type: FC_LONG */
/* 240 */      NdrFcLong( 0xe ),  /* 14 */
/* 244 */      NdrFcShort( 0x2c8 ),                /* Offset= 712 (956) */
/* 246 */      NdrFcLong( 0x400e ),                /* 16398 */
/* 250 */      NdrFcShort( 0x2cc ),                /* Offset= 716 (966) */
/* 252 */      NdrFcLong( 0x4010 ),                /* 16400 */
/* 256 */      NdrFcShort( 0x2ca ),                /* Offset= 714 (970) */
/* 258 */      NdrFcLong( 0x4012 ),                /* 16402 */
/* 262 */      NdrFcShort( 0x286 ),                /* Offset= 646 (908) */
/* 264 */      NdrFcLong( 0x4013 ),                /* 16403 */
/* 268 */      NdrFcShort( 0x284 ),                /* Offset= 644 (912) */
/* 270 */      NdrFcLong( 0x4015 ),                /* 16405 */
/* 274 */      NdrFcShort( 0x282 ),                /* Offset= 642 (916) */
/* 276 */      NdrFcLong( 0x4016 ),                /* 16406 */
/* 280 */      NdrFcShort( 0x278 ),                /* Offset= 632 (912) */
/* 282 */      NdrFcLong( 0x4017 ),                /* 16407 */
/* 286 */      NdrFcShort( 0x272 ),                /* Offset= 626 (912) */
/* 288 */      NdrFcLong( 0x0 ),  /* 0 */
/* 292 */      NdrFcShort( 0x0 ),  /* Offset= 0 (292) */
/* 294 */      NdrFcLong( 0x1 ),  /* 1 */
/* 298 */      NdrFcShort( 0x0 ),  /* Offset= 0 (298) */
/* 300 */      NdrFcShort( 0xffffffff ),           /* Offset= -1 (299) */
/* 302 */
                                                   0x15,                                /* FC_STRUCT */
                                                   0x7,                                 /* 7 */
/* 304 */      NdrFcShort( 0x8 ),  /* 8 */
/* 306 */      0xb,                                /* FC_HYPER */
                                                   0x5b,                                /* FC_END */
/* 308 */
                                                   0x12, 0x0,          /* FC_UP */
/* 310 */      NdrFcShort( 0xc ),  /* Offset= 12 (322) */
/* 312 */
                                                   0x1b,                                /* FC_CARRAY */
                                                   0x1,                                 /* 1 */
/* 314 */      NdrFcShort( 0x2 ),  /* 2 */
/* 316 */      0x9,                                /* Corr desc: FC_ULONG */
                                                   0x0,                                 /*  */
/* 318 */      NdrFcShort( 0xfffc ),               /* -4 */
/* 320 */      0x6,                                /* FC_SHORT */
                                                   0x5b,                                /* FC_END */
/* 322 */
                                                   0x17,                                /* FC_CSTRUCT */
                                                   0x3,                                 /* 3 */
/* 324 */      NdrFcShort( 0x8 ),  /* 8 */
/* 326 */      NdrFcShort( 0xfffffff2 ),           /* Offset= -14 (312) */
/* 328 */      0x8,                                /* FC_LONG */
                                                   0x8,                                 /* FC_LONG */
/* 330 */      0x5c,                               /* FC_PAD */
                                                   0x5b,                                /* FC_END */
/* 332 */
                                                   0x2f,                                /* FC_IP */
                                                   0x5a,                                /* FC_CONSTANT_IID */
/* 334 */      NdrFcLong( 0x0 ),  /* 0 */
/* 338 */      NdrFcShort( 0x0 ),  /* 0 */
/* 340 */      NdrFcShort( 0x0 ),  /* 0 */
/* 342 */      0xc0,                               /* 192 */
                                                   0x0,                                 /* 0 */
/* 344 */      0x0,                                /* 0 */
                                                   0x0,                                 /* 0 */
/* 346 */      0x0,                                /* 0 */
                                                   0x0,                                 /* 0 */
/* 348 */      0x0,                                /* 0 */
                                                   0x46,                                /* 70 */
/* 350 */
                                                   0x2f,                                /* FC_IP */
                                                   0x5a,                                /* FC_CONSTANT_IID */
/* 352 */      NdrFcLong( 0x20400 ),               /* 132096 */
/* 356 */      NdrFcShort( 0x0 ),  /* 0 */
/* 358 */      NdrFcShort( 0x0 ),  /* 0 */
/* 360 */      0xc0,                               /* 192 */
                                                   0x0,                                 /* 0 */
/* 362 */      0x0,                                /* 0 */
                                                   0x0,                                 /* 0 */
/* 364 */      0x0,                                /* 0 */
                                                   0x0,                                 /* 0 */
/* 366 */      0x0,                                /* 0 */
                                                   0x46,                                /* 70 */
/* 368 */
                                                   0x12, 0x10,         /* FC_UP [pointer_deref] */
/* 370 */      NdrFcShort( 0x2 ),  /* Offset= 2 (372) */
/* 372 */
                                                   0x12, 0x0,          /* FC_UP */
/* 374 */      NdrFcShort( 0x1fc ),                /* Offset= 508 (882) */
/* 376 */
                                                   0x2a,                                /* FC_ENCAPSULATED_UNION */
                                                   0x49,                                /* 73 */
/* 378 */      NdrFcShort( 0x18 ),/* 24 */
/* 380 */      NdrFcShort( 0xa ),  /* 10 */
/* 382 */      NdrFcLong( 0x8 ),  /* 8 */
/* 386 */      NdrFcShort( 0x58 ),/* Offset= 88 (474) */
/* 388 */      NdrFcLong( 0xd ),  /* 13 */
/* 392 */      NdrFcShort( 0x78 ),/* Offset= 120 (512) */
/* 394 */      NdrFcLong( 0x9 ),  /* 9 */
/* 398 */      NdrFcShort( 0x94 ),/* Offset= 148 (546) */
/* 400 */      NdrFcLong( 0xc ),  /* 12 */
/* 404 */      NdrFcShort( 0xbc ),/* Offset= 188 (592) */
/* 406 */      NdrFcLong( 0x24 ), /* 36 */
/* 410 */      NdrFcShort( 0x114 ),                /* Offset= 276 (686) */
/* 412 */      NdrFcLong( 0x800d ),                /* 32781 */
/* 416 */      NdrFcShort( 0x130 ),                /* Offset= 304 (720) */
/* 418 */      NdrFcLong( 0x10 ), /* 16 */
/* 422 */      NdrFcShort( 0x148 ),                /* Offset= 328 (750) */
/* 424 */      NdrFcLong( 0x2 ),  /* 2 */
/* 428 */      NdrFcShort( 0x160 ),                /* Offset= 352 (780) */
/* 430 */      NdrFcLong( 0x3 ),  /* 3 */
/* 434 */      NdrFcShort( 0x178 ),                /* Offset= 376 (810) */
/* 436 */      NdrFcLong( 0x14 ), /* 20 */
/* 440 */      NdrFcShort( 0x190 ),                /* Offset= 400 (840) */
/* 442 */      NdrFcShort( 0xffffffff ),           /* Offset= -1 (441) */
/* 444 */
                                                   0x1b,                                /* FC_CARRAY */
                                                   0x3,                                 /* 3 */
/* 446 */      NdrFcShort( 0x4 ),  /* 4 */
/* 448 */      0x19,                               /* Corr desc:  field pointer, FC_ULONG */
                                                   0x0,                                 /*  */
/* 450 */      NdrFcShort( 0x0 ),  /* 0 */
/* 452 */
                                                   0x4b,                                /* FC_PP */
                                                   0x5c,                                /* FC_PAD */
/* 454 */
                                                   0x48,                                /* FC_VARIABLE_REPEAT */
                                                   0x49,                                /* FC_FIXED_OFFSET */
/* 456 */      NdrFcShort( 0x4 ),  /* 4 */
/* 458 */      NdrFcShort( 0x0 ),  /* 0 */
/* 460 */      NdrFcShort( 0x1 ),  /* 1 */
/* 462 */      NdrFcShort( 0x0 ),  /* 0 */
```

```
/* 464 */      NdrFcShort( 0x0 ),   /* 0 */
/* 466 */      0x12, 0x0,           /* FC_UP */
/* 468 */      NdrFcShort( 0xffffff6e ),          /* Offset= -146 (322) */
/* 470 */
                                                  0x5b,                /* FC_END */

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

                                                  0x8,                 /* FC_LONG */
/* 492 */      0x8,                               /* FC_LONG */
                                                  0x5b,                /* FC_END */
/* 494 */
                                                  0x21,                /* FC_BOGUS_ARRAY */
                                                  0x3,                 /* 3 */
/* 496 */      NdrFcShort( 0x0 ),   /* 0 */
/* 498 */      0x19,                              /* Corr desc:  field pointer, FC_ULONG */
                                                  0x0,                 /*  */
/* 500 */      NdrFcShort( 0x0 ),   /* 0 */
/* 502 */      NdrFcLong( 0xffffffff ),           /* -1 */
/* 506 */      0x4c,                              /* FC_EMBEDDED_COMPLEX */
                                                  0x0,                 /* 0 */
/* 508 */      NdrFcShort( 0xffffff50 ),          /* Offset= -176 (332) */
/* 510 */      0x5c,                              /* FC_PAD */
                                                  0x5b,                /* FC_END */
/* 512 */
                                                  0x1a,                /* FC_BOGUS_STRUCT */
                                                  0x3,                 /* 3 */
/* 514 */      NdrFcShort( 0x8 ),   /* 8 */
/* 516 */      NdrFcShort( 0x0 ),   /* 0 */
/* 518 */      NdrFcShort( 0x6 ),   /* Offset= 6 (524) */
/* 520 */      0x8,                               /* FC_LONG */
                                                  0x36,                /* FC_POINTER */
/* 522 */      0x5c,                              /* FC_PAD */
                                                  0x5b,                /* FC_END */
/* 524 */
                                                  0x11, 0x0,           /* FC_RP */
/* 526 */      NdrFcShort( 0xfffffe0 ),           /* Offset= -32 (494) */
/* 528 */
                                                  0x21,                /* FC_BOGUS_ARRAY */
                                                  0x3,                 /* 3 */
/* 530 */      NdrFcShort( 0x0 ),   /* 0 */
/* 532 */      0x19,                              /* Corr desc:  field pointer, FC_ULONG */
                                                  0x0,                 /*  */
/* 534 */      NdrFcShort( 0x0 ),   /* 0 */
/* 536 */      NdrFcLong( 0xffffffff ),           /* -1 */
/* 540 */      0x4c,                              /* FC_EMBEDDED_COMPLEX */
                                                  0x0,                 /* 0 */
/* 542 */      NdrFcShort( 0xfffffc40 ),          /* Offset= -192 (350) */
/* 544 */      0x5c,                              /* FC_PAD */
                                                  0x5b,                /* FC_END */
/* 546 */
                                                  0x1a,                /* FC_BOGUS_STRUCT */
                                                  0x3,                 /* 3 */
/* 548 */      NdrFcShort( 0x8 ),   /* 8 */
/* 550 */      NdrFcShort( 0x0 ),   /* 0 */
/* 552 */      NdrFcShort( 0x6 ),   /* Offset= 6 (558) */
/* 554 */      0x8,                               /* FC_LONG */
                                                  0x36,                /* FC_POINTER */
/* 556 */      0x5c,                              /* FC_PAD */
                                                  0x5b,                /* FC_END */
/* 558 */
                                                  0x11, 0x0,           /* FC_RP */
/* 560 */      NdrFcShort( 0xfffffe0 ),           /* Offset= -32 (528) */
/* 562 */
                                                  0x1b,                /* FC_CARRAY */
                                                  0x3,                 /* 3 */
/* 564 */      NdrFcShort( 0x4 ),   /* 4 */
/* 566 */      0x19,                              /* Corr desc:  field pointer, FC_ULONG */
                                                  0x0,                 /*  */
/* 568 */      NdrFcShort( 0x0 ),   /* 0 */
/* 570 */
                                                  0x4b,                /* FC_PP */
                                                  0x5c,                /* FC_PAD */
/* 572 */
                                                  0x48,                /* FC_VARIABLE_REPEAT */
                                                  0x49,                /* FC_FIXED_OFFSET */
/* 574 */      NdrFcShort( 0x4 ),   /* 4 */
/* 576 */      NdrFcShort( 0x0 ),   /* 0 */
/* 578 */      NdrFcShort( 0x1 ),   /* 1 */
/* 580 */      NdrFcShort( 0x0 ),   /* 0 */
/* 582 */      NdrFcShort( 0x0 ),   /* 0 */
/* 584 */      0x12, 0x0,           /* FC_UP */
/* 586 */      NdrFcShort( 0x184 ),               /* Offset= 388 (974) */
/* 588 */
                                                  0x5b,                /* FC_END */

                                                  0x8,                 /* FC_LONG */
/* 590 */      0x5c,                              /* FC_PAD */
                                                  0x5b,                /* FC_END */
/* 592 */
                                                  0x1a,                /* FC_BOGUS_STRUCT */
                                                  0x3,                 /* 3 */
/* 594 */      NdrFcShort( 0x8 ),   /* 8 */
/* 596 */      NdrFcShort( 0x0 ),   /* 0 */
/* 598 */      NdrFcShort( 0x6 ),   /* Offset= 6 (604) */
/* 600 */      0x8,                               /* FC_LONG */
                                                  0x36,                /* FC_POINTER */
/* 602 */      0x5c,                              /* FC_PAD */
                                                  0x5b,                /* FC_END */
/* 604 */
                                                  0x11, 0x0,           /* FC_RP */
/* 606 */      NdrFcShort( 0xfffffd4 ),           /* Offset= -44 (562) */
/* 608 */
                                                  0x2f,                /* FC_IP */
                                                  0x5a,                /* FC_CONSTANT_IID */
/* 610 */      NdrFcLong( 0x2f ),   /* 47 */
/* 614 */      NdrFcShort( 0x0 ),   /* 0 */
/* 616 */      NdrFcShort( 0x0 ),   /* 0 */
/* 618 */      0xc0,                              /* 192 */
                                                  0x0,                 /* 0 */
/* 620 */      0x0,                               /* 0 */
                                                  0x0,                 /* 0 */
/* 622 */      0x0,                               /* 0 */
                                                  0x0,                 /* 0 */
/* 624 */      0x0,                               /* 0 */
                                                  0x46,                /* 70 */
/* 626 */
                                                  0x1b,                /* FC_CARRAY */
                                                  0x0,                 /* 0 */
/* 628 */      NdrFcShort( 0x1 ),   /* 1 */
/* 630 */      0x19,                              /* Corr desc:  field pointer, FC_ULONG */
                                                  0x0,                 /*  */
/* 632 */      NdrFcShort( 0x4 ),   /* 4 */
/* 634 */      0x1,                               /* FC_BYTE */
                                                  0x5b,                /* FC_END */
/* 636 */
                                                  0x1a,                /* FC_BOGUS_STRUCT */
                                                  0x3,                 /* 3 */
/* 638 */      NdrFcShort( 0x10 ),  /* 16 */
/* 640 */      NdrFcShort( 0x0 ),   /* 0 */
/* 642 */      NdrFcShort( 0xa ),   /* Offset= 10 (652) */
/* 644 */      0x8,                               /* FC_LONG */
                                                  0x8,                 /* FC_LONG */
/* 646 */      0x4c,                              /* FC_EMBEDDED_COMPLEX */
```

```
                                        0x0,                                    /* 0 */
/* 648 */       NdrFcShort( 0xffffffd8 ),       /* Offset= -40 (608) */
/* 650 */       0x36,                           /* FC_POINTER */
                                        0x5b,                                   /* FC_END */
/* 652 */
                                        0x12, 0x0,              /* FC_UP */
/* 654 */       NdrFcShort( 0xffffffe4 ),       /* Offset= -28 (626) */
/* 656 */
                                        0x1b,                                   /* FC_CARRAY */
                                        0x3,                                    /* 3 */
/* 658 */       NdrFcShort( 0x4 ),  /* 4 */
/* 660 */       0x19,                           /* Corr desc:  field pointer, FC_ULONG */
                                        0x0,                                    /*  */
/* 662 */       NdrFcShort( 0x0 ),  /* 0 */
/* 664 */
                                        0x4b,                                   /* FC_PP */
                                        0x5c,                                   /* FC_PAD */
/* 666 */
                                        0x48,                                   /* FC_VARIABLE_REPEAT */
                                        0x49,                                   /* FC_FIXED_OFFSET */
/* 668 */       NdrFcShort( 0x4 ),  /* 4 */
/* 670 */       NdrFcShort( 0x0 ),  /* 0 */
/* 672 */       NdrFcShort( 0x1 ),  /* 1 */
/* 674 */       NdrFcShort( 0x0 ),  /* 0 */
/* 676 */       NdrFcShort( 0x0 ),  /* 0 */
/* 678 */       0x12, 0x0,              /* FC_UP */
/* 680 */       NdrFcShort( 0xffffffd4 ),       /* Offset= -44 (636) */
/* 682 */
                                        0x5b,                                   /* FC_END */

                                        0x8,                                    /* FC_LONG */
/* 684 */       0x5c,                           /* FC_PAD */
                                        0x5b,                                   /* FC_END */
/* 686 */
                                        0x1a,                                   /* FC_BOGUS_STRUCT */
                                        0x3,                                    /* 3 */
/* 688 */       NdrFcShort( 0x8 ),  /* 8 */
/* 690 */       NdrFcShort( 0x0 ),  /* 0 */
/* 692 */       NdrFcShort( 0x6 ),  /* Offset= 6 (698) */
/* 694 */       0x8,                            /* FC_LONG */
                                        0x36,                                   /* FC_POINTER */
/* 696 */       0x5c,                           /* FC_PAD */
                                        0x5b,                                   /* FC_END */
/* 698 */
                                        0x11, 0x0,              /* FC_RP */
/* 700 */       NdrFcShort( 0xffffffd4 ),       /* Offset= -44 (656) */
/* 702 */
                                        0x1d,                                   /* FC_SMFARRAY */
                                        0x0,                                    /* 0 */
/* 704 */       NdrFcShort( 0x8 ),  /* 8 */
/* 706 */       0x1,                            /* FC_BYTE */
                                        0x5b,                                   /* FC_END */
/* 708 */
                                        0x15,                                   /* FC_STRUCT */
                                        0x3,                                    /* 3 */
/* 710 */       NdrFcShort( 0x10 ),/* 16 */
/* 712 */       0x8,                            /* FC_LONG */
                                        0x6,                                    /* FC_SHORT */
/* 714 */       0x6,                            /* FC_SHORT */
                                        0x4c,                                   /* FC_EMBEDDED_COMPLEX */
/* 716 */       0x0,                            /* 0 */
                                        NdrFcShort( 0xfffffff1 ),       /* Offset= -15 (702) */
                                        0x5b,                                   /* FC_END */
/* 720 */
                                        0x1a,                                   /* FC_BOGUS_STRUCT */
                                        0x3,                                    /* 3 */
/* 722 */       NdrFcShort( 0x18 ),/* 24 */
/* 724 */       NdrFcShort( 0x0 ),  /* 0 */
/* 726 */       NdrFcShort( 0xa ),  /* Offset= 10 (736) */
/* 728 */       0x8,                            /* FC_LONG */
                                        0x36,                                   /* FC_POINTER */
/* 730 */       0x4c,                           /* FC_EMBEDDED_COMPLEX */
                                        0x0,                                    /* 0 */
/* 732 */       NdrFcShort( 0xffffffe8 ),       /* Offset= -24 (708) */
/* 734 */       0x5c,                           /* FC_PAD */
                                        0x5b,                                   /* FC_END */
/* 736 */
                                        0x11, 0x0,              /* FC_RP */
/* 738 */       NdrFcShort( 0xfffffe0c ),       /* Offset= -244 (494) */
/* 740 */
                                        0x1b,                                   /* FC_CARRAY */
                                        0x0,                                    /* 0 */
/* 742 */       NdrFcShort( 0x1 ),  /* 1 */
/* 744 */       0x19,                           /* Corr desc:  field pointer, FC_ULONG */
                                        0x0,                                    /*  */
/* 746 */       NdrFcShort( 0x0 ),  /* 0 */
/* 748 */       0x1,                            /* FC_BYTE */
                                        0x5b,                                   /* FC_END */
/* 750 */
                                        0x16,                                   /* FC_PSTRUCT */
                                        0x3,                                    /* 3 */
/* 752 */       NdrFcShort( 0x8 ),  /* 8 */
/* 754 */
                                        0x4b,                                   /* FC_PP */
                                        0x5c,                                   /* FC_PAD */
/* 756 */
                                        0x46,                                   /* FC_NO_REPEAT */
                                        0x5c,                                   /* FC_PAD */
/* 758 */       NdrFcShort( 0x4 ),  /* 4 */
/* 760 */       NdrFcShort( 0x4 ),  /* 4 */
/* 762 */       0x12, 0x0,              /* FC_UP */
/* 764 */       NdrFcShort( 0xffffffe8 ),       /* Offset= -24 (740) */
/* 766 */
                                        0x5b,                                   /* FC_END */

                                        0x8,                                    /* FC_LONG */
/* 768 */       0x8,                            /* FC_LONG */
                                        0x5b,                                   /* FC_END */
/* 770 */
                                        0x1b,                                   /* FC_CARRAY */
                                        0x1,                                    /* 1 */
/* 772 */       NdrFcShort( 0x2 ),  /* 2 */
/* 774 */       0x19,                           /* Corr desc:  field pointer, FC_ULONG */
                                        0x0,                                    /*  */
/* 776 */       NdrFcShort( 0x0 ),  /* 0 */
/* 778 */       0x6,                            /* FC_SHORT */
                                        0x5b,                                   /* FC_END */
/* 780 */
                                        0x16,                                   /* FC_PSTRUCT */
                                        0x3,                                    /* 3 */
/* 782 */       NdrFcShort( 0x8 ),  /* 8 */
/* 784 */
                                        0x4b,                                   /* FC_PP */
                                        0x5c,                                   /* FC_PAD */
/* 786 */
                                        0x46,                                   /* FC_NO_REPEAT */
                                        0x5c,                                   /* FC_PAD */
/* 788 */       NdrFcShort( 0x4 ),  /* 4 */
/* 790 */       NdrFcShort( 0x4 ),  /* 4 */
/* 792 */       0x12, 0x0,              /* FC_UP */
/* 794 */       NdrFcShort( 0xffffffe8 ),       /* Offset= -24 (770) */
/* 796 */
                                        0x5b,                                   /* FC_END */

                                        0x8,                                    /* FC_LONG */
/* 798 */       0x8,                            /* FC_LONG */
                                        0x5b,                                   /* FC_END */
/* 800 */
                                        0x1b,                                   /* FC_CARRAY */
                                        0x3,                                    /* 3 */
/* 802 */       NdrFcShort( 0x4 ),  /* 4 */
/* 804 */       0x19,                           /* Corr desc:  field pointer, FC_ULONG */
                                        0x0,                                    /*  */
/* 806 */       NdrFcShort( 0x0 ),  /* 0 */
/* 808 */       0x8,                            /* FC_LONG */
                                        0x5b,                                   /* FC_END */
/* 810 */
                                        0x16,                                   /* FC_PSTRUCT */
                                        0x3,                                    /* 3 */
/* 812 */       NdrFcShort( 0x8 ),  /* 8 */
/* 814 */
                                        0x4b,                                   /* FC_PP */
```

```
                                        0x5c,                   /* FC_PAD */
/* 816 */
                                        0x46,                   /* FC_NO_REPEAT */
                                        0x5c,                   /* FC_PAD */
/* 818 */       NdrFcShort( 0x4 ),  /* 4 */
/* 820 */       NdrFcShort( 0x4 ),  /* 4 */
/* 822 */       0x12, 0x0,          /* FC_UP */
/* 824 */       NdrFcShort( 0xffffffe8 ),   /* Offset= -24 (800) */
/* 826 */
                                        0x5b,                   /* FC_END */

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

                                        0x8,                    /* FC_LONG */
/* 858 */       0x8,                    /* FC_LONG */
                                        0x5b,                   /* FC_END */
/* 860 */
                                        0x15,                   /* FC_STRUCT */
                                        0x3,                    /* 3 */
/* 862 */       NdrFcShort( 0x8 ),  /* 8 */
/* 864 */       0x8,                    /* FC_LONG */
                                        0x8,                    /* FC_LONG */
/* 866 */       0x5c,                   /* FC_PAD */
                                        0x5b,                   /* FC_END */
/* 868 */
                                        0x1b,                   /* FC_CARRAY */
                                        0x3,                    /* 3 */
/* 870 */       NdrFcShort( 0x8 ),  /* 8 */
/* 872 */       0x7,                    /* Corr desc: FC_USHORT */
                                        0x0,                    /*  */
/* 874 */       NdrFcShort( 0xffd8 ),   /* -40 */
/* 876 */       0x4c,                   /* FC_EMBEDDED_COMPLEX */
                                        0x0,                    /* 0 */
/* 878 */       NdrFcShort( 0xffffffee ),   /* Offset= -18 (860) */
/* 880 */       0x5c,                   /* FC_PAD */
                                        0x5b,                   /* FC_END */
/* 882 */
                                        0x1a,                   /* FC_BOGUS_STRUCT */
                                        0x3,                    /* 3 */
/* 884 */       NdrFcShort( 0x28 ),/* 40 */
/* 886 */       NdrFcShort( 0xffffffee ),   /* Offset= -18 (868) */
/* 888 */       NdrFcShort( 0x0 ),  /* Offset= 0 (888) */
/* 890 */       0x6,                    /* FC_SHORT */
                                        0x6,                    /* FC_SHORT */
/* 892 */       0x8,                    /* FC_LONG */
                                        0x8,                    /* FC_LONG */
/* 894 */       0x4c,                   /* FC_EMBEDDED_COMPLEX */
                                        0x0,                    /* 0 */
/* 896 */       NdrFcShort( 0xfffffdf8 ),   /* Offset= -520 (376) */
/* 898 */       0x5c,                   /* FC_PAD */
                                        0x5b,                   /* FC_END */
/* 900 */
                                        0x12, 0x0,          /* FC_UP */
/* 902 */       NdrFcShort( 0xfffffef6 ),   /* Offset= -266 (636) */
/* 904 */
                                        0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 906 */       0x1,                    /* FC_BYTE */
                                        0x5c,                   /* FC_PAD */
/* 908 */
                                        0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 910 */       0x6,                    /* FC_SHORT */
                                        0x5c,                   /* FC_PAD */
/* 912 */
                                        0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 914 */       0x8,                    /* FC_LONG */
                                        0x5c,                   /* FC_PAD */
/* 916 */
                                        0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 918 */       0xb,                    /* FC_HYPER */
                                        0x5c,                   /* FC_PAD */
/* 920 */
                                        0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 922 */       0xa,                    /* FC_FLOAT */
                                        0x5c,                   /* FC_PAD */
/* 924 */
                                        0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 926 */       0xc,                    /* FC_DOUBLE */
                                        0x5c,                   /* FC_PAD */
/* 928 */
                                        0x12, 0x0,          /* FC_UP */
/* 930 */       NdrFcShort( 0xfffffd8c ),   /* Offset= -628 (302) */
/* 932 */
                                        0x12, 0x10,         /* FC_UP [pointer_deref] */
/* 934 */       NdrFcShort( 0xfffffd8e ),   /* Offset= -626 (308) */
/* 936 */
                                        0x12, 0x10,         /* FC_UP [pointer_deref] */
/* 938 */       NdrFcShort( 0xfffffda2 ),   /* Offset= -606 (332) */
/* 940 */
                                        0x12, 0x10,         /* FC_UP [pointer_deref] */
/* 942 */       NdrFcShort( 0xfffffdb0 ),   /* Offset= -592 (350) */
/* 944 */
                                        0x12, 0x10,         /* FC_UP [pointer_deref] */
/* 946 */       NdrFcShort( 0xfffffdbe ),   /* Offset= -578 (368) */
/* 948 */
                                        0x12, 0x10,         /* FC_UP [pointer_deref] */
/* 950 */       NdrFcShort( 0x2 ),  /* Offset= 2 (952) */
/* 952 */
                                        0x12, 0x0,          /* FC_UP */
/* 954 */       NdrFcShort( 0x14 ),/* Offset= 20 (974) */
/* 956 */
                                        0x15,                   /* FC_STRUCT */
                                        0x7,                    /* 7 */
/* 958 */       NdrFcShort( 0x10 ),/* 16 */
/* 960 */       0x6,                    /* FC_SHORT */
                                        0x1,                    /* FC_BYTE */
/* 962 */       0x1,                    /* FC_BYTE */
                                        0x8,                    /* FC_LONG */
/* 964 */       0xb,                    /* FC_HYPER */
                                        0x5b,                   /* FC_END */
/* 966 */
                                        0x12, 0x0,          /* FC_UP */
/* 968 */       NdrFcShort( 0xfffffff4 ),   /* Offset= -12 (956) */
/* 970 */
                                        0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 972 */       0x2,                    /* FC_CHAR */
                                        0x5c,                   /* FC_PAD */
/* 974 */
                                        0x1a,                   /* FC_BOGUS_STRUCT */
                                        0x7,                    /* 7 */
/* 976 */       NdrFcShort( 0x20 ),/* 32 */
/* 978 */       NdrFcShort( 0x0 ),  /* 0 */
/* 980 */       NdrFcShort( 0x0 ),  /* Offset= 0 (980) */
/* 982 */       0x8,                    /* FC_LONG */
                                        0x8,                    /* FC_LONG */
/* 984 */       0x6,                    /* FC_SHORT */
```

```
                                                    0x6,                    /* FC_SHORT */
/* 986 */               0x6,                        /* FC_SHORT */
                                                    0x6,                    /* FC_SHORT */
/* 988 */               0x4c,                       /* FC_EMBEDDED_COMPLEX */
                                                    0x0,                    /* 0 */
/* 990 */               NdrFcShort( 0xfffffc28 ),   /* Offset= -984 (6) */
/* 992 */               0x5c,                       /* FC_PAD */
                                                    0x5b,                   /* FC_END */
/* 994 */               0xb4,                       /* FC_USER_MARSHAL */
                                                    0x83,                   /* 131 */
/* 996 */               NdrFcShort( 0x0 ),  /* 0 */
/* 998 */               NdrFcShort( 0x10 ),/* 16 */
/* 1000 */              NdrFcShort( 0x0 ),  /* 0 */
/* 1002 */              NdrFcShort( 0xfffffc18 ),   /* Offset= -1000 (2) */
/* 1004 */
                                                    0x11, 0x4,              /* FC_RP [alloced_on_stack] */
/* 1006 */              NdrFcShort( 0x6 ),  /* Offset= 6 (1012) */
/* 1008 */
                                                    0x13, 0x0,              /* FC_OP */
/* 1010 */              NdrFcShort( 0xffffffdc ),   /* Offset= -36 (974) */
/* 1012 */              0xb4,                       /* FC_USER_MARSHAL */
                                                    0x83,                   /* 131 */
/* 1014 */              NdrFcShort( 0x0 ),  /* 0 */
/* 1016 */              NdrFcShort( 0x10 ),/* 16 */
/* 1018 */              NdrFcShort( 0x0 ),  /* 0 */
/* 1020 */              NdrFcShort( 0xfffffff4 ),   /* Offset= -12 (1008) */

                                                    0x0

        }
    };

static const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
        {

            {
            VARIANT_UserSize
            ,VARIANT_UserMarshal
            ,VARIANT_UserUnmarshal
            ,VARIANT_UserFree
            }

        };



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


/* 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}} */

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

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


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

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

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

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

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

PCInterfaceName const _tpcc_com_ps_InterfaceNamesList[] =
{
    "ITPCC",
    0
```

```
};


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

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

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

   return 0;
}

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

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

SQL Stored Procedures

Kit MSTPCC451\SETUP\scripts\dml\delivery.sql

```
-------------------------------------------------------------------
--                                                                 --
--                File:           DELIVERY.SQL                                 --
--                                Microsoft TPC-C Benchmark Kit Ver. 4.51      --
--                                Copyright Microsoft, 2003                    --
--                                                                 --
--                                Creates 4 delivery stored procedure          --
--                                préfixées Q0, Q1, Q2 et Q3                   --
--                                                                 --
--                                Interface Level:    4.10.000                 --
--                                                                 --
-------------------------------------------------------------------

use tpcc
go

-------------------------------------------------------------------
-- création de la proc préfixée Q0
-------------------------------------------------------------------

if exists (select name from sysobjects where name = 'Q0tpcc_delivery' )
                drop procedure Q0tpcc_delivery
go

create proc Q0tpcc_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 WITH (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


-----------------------------------------------------------------------
-- création de la proc préfixée Q1
-----------------------------------------------------------------------

if exists (select name from sysobjects where name = 'Q1tpcc_delivery' )
                drop procedure Q1tpcc_delivery
go

create proc Q1tpcc_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 WITH (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

-----------------------------------------------------------------------
-- création de la proc préfixée Q2
-----------------------------------------------------------------------

if exists (select name from sysobjects where name = 'Q2tpcc_delivery' )
                drop procedure Q2tpcc_delivery
go
```

```
create proc Q2tpcc_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 WITH (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


-----------------------------------------------------------------------
-- création de la proc préfixée Q3
-----------------------------------------------------------------------

if exists (select name from sysobjects where name = 'Q3tpcc_delivery' )
                    drop procedure Q3tpcc_delivery
go

create proc Q3tpcc_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 WITH (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
```

Kit MSTPCC451\SETUP\scripts\dml\neword.sql

```
-------------------------------------------------------------------
--                                                                 --
--                      File:           NEWORD.SQL                 --
--                                      Microsoft TPC-C Benchmark Kit Ver. 4.51    --
--                                      Copyright Microsoft, 2003  --
--                                                                 --
--                                      Creates 4 neworder stored procedure        --
--                                      préfixées Q0, Q1, Q2 et Q3                 --
--                                                                 --
--                                      Interface Level:    4.10.000               --
--                                                                 --
-------------------------------------------------------------------

use tpcc
go

-------------------------------------------------------------------
-- création de la proc préfixée Q0
-------------------------------------------------------------------

if exists ( select name from sysobjects where name = 'Q0tpcc_neworder' )
                drop procedure Q0tpcc_neworder
go

create proc Q0tpcc_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 WITH (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 WITH (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 WITH (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



-----------------------------------------------------------------------
-- création de la proc préfixée Q1
-----------------------------------------------------------------------

if exists ( select name from sysobjects where name = 'Q1tpcc_neworder' )
                        drop procedure Q1tpcc_neworder
go

create proc Q1tpcc_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 WITH (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 WITH (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 WITH (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

-----------------------------------------------------------------
-- création de la proc préfixée Q2
-----------------------------------------------------------------

if exists ( select name from sysobjects where name = 'Q2tpcc_neworder' )
                        drop procedure Q2tpcc_neworder
go

create proc Q2tpcc_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 WITH (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 WITH (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 WITH (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


-----------------------------------------------------------------------
-- création de la proc préfixée Q3
-----------------------------------------------------------------------

if exists ( select name from sysobjects where name = 'Q3tpcc_neworder' )
                drop procedure Q3tpcc_neworder
go

create proc Q3tpcc_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 WITH (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 WITH (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 WITH (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
```

Kit MSTPCC451\SETUP\scripts\dml\ordstat.sql

```
-----------------------------------------------------------------------
--                                                                    --
--              File:           ORDSTAT.SQL                           --
--                              Microsoft TPC-C Benchmark Kit Ver. 4.51       --
--                              Copyright Microsoft, 2003             --
--                                                                    --
--                              Creates 4 order status stored procedure       --
--                              préfixées Q0, Q1, Q2 et Q3            --
--                                                                    --
--                              Interface Level:  4.10.000            --
--                                                                    --
-----------------------------------------------------------------------

use tpcc
go

-------------------------------------------------------------------
-- création de la proc préfixée Q0
```

```
-----------------------------------------------------------------------

if exists ( select name from sysobjects where name = 'Q0tpcc_orderstatus' )
                drop procedure  Q0tpcc_orderstatus
go

create proc Q0tpcc_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 WITH (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 WITH (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 WITH (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 WITH (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 WITH (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


-----------------------------------------------------------------------
-- création de la proc préfixée Q1
-----------------------------------------------------------------------

if exists ( select name from sysobjects where name = 'Q1tpcc_orderstatus' )
                drop procedure  Q1tpcc_orderstatus
go

create proc Q1tpcc_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 WITH (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 WITH (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 WITH (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 WITH (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 WITH (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

-----------------------------------------------------------------------
-- création de la proc préfixée Q2
-----------------------------------------------------------------------

if exists ( select name from sysobjects where name = 'Q2tpcc_orderstatus' )
                        drop procedure  Q2tpcc_orderstatus
go

create proc Q2tpcc_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 WITH (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 WITH (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 WITH (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 WITH (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 WITH (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


-----------------------------------------------------------------------
-- création de la proc préfixée Q3
-----------------------------------------------------------------------

if exists ( select name from sysobjects where name = 'Q3tpcc_orderstatus' )
                drop procedure  Q3tpcc_orderstatus
go

create proc Q3tpcc_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 WITH (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 WITH (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 WITH (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 WITH (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 WITH (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
```

Kit MSTPCC451\SETUP\scripts\dml\payment.sql

```
-----------------------------------------------------------------------
--                                                                     --
--              File:           PAYMENT.SQL                            --
--                              Microsoft TPC-C Benchmark Kit Ver. 4.51 --
--                              Copyright Microsoft, 2003              --
--                                                                     --
--                              Creates 4 payment stored procedure     --
--                              préfixées Q0, Q1, Q2 et Q3             --
--                                                                     --
--                              Interface Level:    4.10.000           --
--                                                                     --
-----------------------------------------------------------------------

use tpcc
go

-----------------------------------------------------------------------
-- création de la proc préfixée Q0
-----------------------------------------------------------------------

if exists (select name from sysobjects where name = 'Q0tpcc_payment' )
        drop procedure Q0tpcc_payment
go

create proc Q0tpcc_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),
                @c_data                                 char(42),
                @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 WITH (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 WITH (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
                                set             c_data          = @c_data + substring(c_data,1,458),
                                                @screen_data = @c_data + substring(c_data,1,158)
                                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



-----------------------------------------------------------------------
-- création de la proc préfixée Q1
-----------------------------------------------------------------------

if exists (select name from sysobjects where name = 'Q1tpcc_payment' )
                drop procedure Q1tpcc_payment
go

create proc Q1tpcc_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),
                @c_data                                 char(42),
                @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 WITH (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 WITH (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
                                set             c_data          = @c_data + substring(c_data,1,458),
                                                @screen_data = @c_data + substring(c_data,1,158)
                                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
-------------------------------------------------------------------------
-- création de la proc préfixée Q2
-------------------------------------------------------------------------

if exists (select name from sysobjects where name = 'Q2tpcc_payment' )
                drop procedure Q2tpcc_payment
go

create proc Q2tpcc_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),
                @c_data                                 char(42),
                @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 WITH (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 WITH (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
                                set                 c_data              = @c_data + substring(c_data,1,458),
                                                    @screen_data = @c_data + substring(c_data,1,158)
                                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


-------------------------------------------------------------------------
-- création de la proc préfixée Q3
-------------------------------------------------------------------------

if exists (select name from sysobjects where name = 'Q3tpcc_payment' )
                drop procedure Q3tpcc_payment
go

create proc Q3tpcc_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),
                @c_data                                 char(42),
                @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 WITH (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 WITH (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
                        set             c_data          = @c_data + substring(c_data,1,458),
                                        @screen_data = @c_data + substring(c_data,1,158)
                        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
```

Kit MSTPCC451\SETUP\scripts\dml\stocklev.sql

```
------------------------------------------------------------------------
--                                                                      --
--              File:           STOCKLEV.SQL                            --
--                              Microsoft TPC-C Benchmark Kit Ver. 4.51 --
--                              Copyright Microsoft, 2003               --
--                                                                      --
--              Creates 4 stock level stored procedure                  --
--              préfixées Q0, Q1, Q2 et Q3                              --
--                                                                      --
--              Interface Level:        4.10.000                        --
--                                                                      --
------------------------------------------------------------------------

use tpcc
go

------------------------------------------------------------------------
-- création de la proc préfixée Q0
------------------------------------------------------------------------

if exists (select name from sysobjects where name = 'Q0tpcc_stocklevel' )
                drop procedure Q0tpcc_stocklevel
go

create proc Q0tpcc_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


------------------------------------------------------------------------
-- création de la proc préfixée Q1
------------------------------------------------------------------------

if exists (select name from sysobjects where name = 'Q1tpcc_stocklevel' )
                drop procedure Q1tpcc_stocklevel
go

create proc Q1tpcc_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

------------------------------------------------------------------------
-- création de la proc préfixée Q2
------------------------------------------------------------------------

if exists (select name from sysobjects where name = 'Q2tpcc_stocklevel' )
                drop procedure Q2tpcc_stocklevel
go

create proc Q2tpcc_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


-----------------------------------------------------------------------
-- création de la proc préfixée Q3
-----------------------------------------------------------------------

if exists (select name from sysobjects where name = 'Q3tpcc_stocklevel' )
                drop procedure Q3tpcc_stocklevel
go

create proc Q3tpcc_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
```

Kit MSTPCC451\SETUP\scripts\dml\version.sql

```
-----------------------------------------------------------------------
--                                                                                              --
--              File:           VERSION.SQL                                             --
--                              Microsoft TPC-C Benchmark Kit Ver. 4.51                 --
--                              Copyright Microsoft, 2003                               --
--                                                                                              --
--                              Extracts current version of SQL Server          --
--                                                                                              --
-----------------------------------------------------------------------

use master
go

SELECT          CONVERT(char(20), SERVERPROPERTY('ProductVersion')),
                CONVERT(char(20), SERVERPROPERTY('ProductLevel')),
                CONVERT(char(29), SERVERPROPERTY('Edition'))
go

SELECT          CONVERT(char(30), getdate(), 21)
go
```

Appendix B: Database definition and population code used in the tests.

Build Scripts

Kit MSTPCC451\SETUP\setup.cmd

```
::---------------------------------------------------------------------------------------
::---  FILE:      RUNSQLCFG.CMD
::---             Microsoft TPC-C Kit Ver. 4.51
::---             Copyright Microsoft, 2001, 2002, 2003
::---             All Rights Reserved
::---
::---  PURPOSE:    Calls RunSQLCfg.sql to configure SQL Server
::---
::---  ARGUMENTS:  Optionally, the user can pass the following positional arguments:
::---               Server Name
::---               SQL Server User ID (sa)
::---               SQL Server account password
::---               Number of Warehouses
::---               Build Option
::---                 {full,builddb,objects,objectsfull,bulkload,bulkloadfull,backup}
::---               Database Type
::---                 {normal or scale_down}
::---               Unattended Build
::---                                 {true or false}
::---                   (true will display all confirmation messages (default))
::---                   (false will supress all confirmation messages)
::---
::---             If they are not passed, then the user will be prompted by the VBS file.
::---
::---------------------------------------------------------------------------------------
@cscript SetupScripts\setup.vbs //H:CScript //I %1 %2 %3 %4 %5 %6 %7
```

Kit MSTPCC451\SETUP\SetupScripts\setup.vbs

```
'---------------------------------------------------------------------------------------
'---  FILE:      SETUP.VBS
'---             Microsoft TPC-C Kit Ver. 4.51
'---             Copyright Microsoft, 2001, 2002, 2003
'---             All Rights Reserved
'---
'---  PURPOSE:    This module performs the tasks to create and populate a TPC-C database
'---
'---------------------------------------------------------------------------------------
'---------------------------------------------------
'--- set the kit verison variable for later display
'---------------------------------------------------
Kit_Version = "  4.51"
'----------------------------------------
'--- open an windows scripting object
'----------------------------------------
Set WshShell = CreateObject("WScript.Shell")
'---------------------------------------------------
'--- set up windows scripting argument collection
'---------------------------------------------------
Set ObjArgs = wScript.ARGUMENTS
'------------------------------------------------------------------------
'--- grab the plaftorm, ia64 or x86, from the environment variables
'------------------------------------------------------------------------
Platform = Left(WshShell.ExpandEnvironmentStrings("%PROCESSOR_IDENTIFIER%"), 4)
'-------------------------------------------------------------------------
'--- grab the processor architecture.  This is to determine if the
'--- user is trying to run in 32-bit emulation on a 64-bit machine.
'--- if that is the case, then display a message and exit.
'-------------------------------------------------------------------------
Proc_Architecture = WshShell.ExpandEnvironmentStrings("%PROCESSOR_ARCHITECTURE%")
If Platform = "ia64" And Proc_Architecture = "x86" Then
	wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
	wScript.Echo "!!                                                                     !!"
	wScript.Echo "!! You are attempting to run this SETUP in the 32-bit (WOW) emulation !!"
	wScript.Echo "!! mode on an ia64 system.  Please restart the SETUP in a native      !!"
	wScript.Echo "!! 64-bit environment.                                                 !!"
	wScript.Echo "!!                                                                     !!"
	wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
	wScript.Quit
End If
'------------------------------------------------
'--- before we go any further, make sure that
'--- we are running Windows Scripting Host 5.6
'--- or higher
'------------------------------------------------
If wScript.Version < 5.6 Then
	wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
	wScript.Echo "!!                                                                     !!"
	wScript.Echo "!! You do not have the proper version of the Windows Scripting Host   !!"
	wScript.Echo "!! installed.  Please install the latest Windows Scripting Host from  !!"
	wScript.Echo "!! ..\tools\wsh\scripten.exe and restart setup.                        !!"
	wScript.Echo "!!                                                                     !!"
	wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
	wScript.Quit
End If
'----------------------------------------
'--- display banner message
'----------------------------------------
wScript.Echo "*************************************************************************"
wScript.Echo "*                                                                       *"
wScript.Echo "*  Microsoft TPC-C Benchmark Kit Ver." & Kit_Version & "                            *"
wScript.Echo "*                                                                       *"
wScript.Echo "*  Database Setup                                                       *"
wScript.Echo "*                                                                       *"
wScript.Echo "*************************************************************************"
'------------------------------------------------
'--- define function to check for any error messages
'------------------------------------------------
Function CheckSQLOutput(SQL_Out)
	ErrorFlag = 0
	Set SQL_fso = CreateObject("Scripting.FileSystemObject")
	If SQL_fso.FileExists(SQL_Out) Then
		Set SQL_Out_File = SQL_fso.OpenTextFile(SQL_Out, 1)
		Do While SQL_Out_File.AtEndOfStream <> True
			SQL_Line = SQL_Out_File.ReadLine
			'first check to see if the output contains a message about the login password
			If InStr(SQL_Line, "Login failed") Then
				'display the messages and get out of here
				ErrorFlag = 1
				wScript.Echo "The login for userid 'sa' failed."
				wScript.Echo "Please restart SETUP with the correct password."
			Else
				If InStr(SQL_Line, "Msg") Then
					'find out where the "Msg" indicator is in the line
					LocMsg = InStr(SQL_Line, "Msg")
					'find out where the comma is after the error code
					LocComma = InStr(SQL_Line, ",")
					'now isolate the error code
					ErrorCode = Mid(SQL_Line, (LocMsg + 4), (LocComma - (LocMsg + 4)))
					Select Case ErrorCode
						Case " 170"
							ErrorFlag = 1
							wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
							wScript.Echo "Syntax Error."
							wScript.Echo "SQL Server Error 170."
							wScript.Echo "Check CREATEDB.SQL."
							wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
						Case "1801"
							ErrorFlag = 1
							wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
							wScript.Echo "Database 'tpcc' already exists."
							wScript.Echo "SQL Server Error 1801."
							wScript.Echo "Check CREATEDB.SQL."
							wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
						Case "1802"
```

```
                                ErrorFlag = 1
                                wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                                wScript.Echo "CREATE DATABASE failed."
                                wScript.Echo "SQL Server Error 1802."
                                wScript.Echo "Check CREATEDB.SQL."
                                wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                            Case "1921"
                                ErrorFlag = 1
                                wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                                wScript.Echo "CREATE INDEX failed."
                                wScript.Echo "SQL Server Error 1921."
                                wScript.Echo "Check " & SQL_Out & "."
                                wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                            Case "3013"
                        ErrorFlag = 1
                        wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                                wScript.Echo "BACKUP DATABASE is terminating abnormally."
                                wScript.Echo "SQL Server Error 3013."
                                wScript.Echo "Check the SQL Server error log for more details."
                                wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                            Case "3201"
                        ErrorFlag = 1
                        wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                                wScript.Echo "Cannot open backup device."
                                wScript.Echo "Device error or device off-line."
                                wScript.Echo "SQL Server Error 3201."
                                wScript.Echo "See the SQL Server error log for more details."
                                wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                            Case "5105"
                        ErrorFlag = 1
                        wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                                wScript.Echo "Device Activation Error."
                                wScript.Echo "SQL Server Error 5105."
                                wScript.Echo "Check CREATEDB.SQL."
                                wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                            Case "5170"
                                ErrorFlag = 1
                                wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                                wScript.Echo "Cannot create one or more files because it already exists."
                                wScript.Echo "SQL Server Error 5170."
                                wScript.Echo "Check CREATEDB.SQL."
                                wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                            Case "15010", "15012"
                                ErrorFlag = 0
                            Case "15069"
                                ErrorFlag = 1
                                wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                                wScript.Echo "One or more users are using the database."
                                wScript.Echo "The requested operation cannot be completed."
                                wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                            Case Else
                                ErrorFlag = 1
                                wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                                wScript.Echo "An error occurred."
                                wScript.Echo "SQL Server Error Code: " & ErrorCode & "."
                                wScript.Echo "Check " & SQL_Out & " for more information."
                                wScript.Echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
                        End Select
                    End If
                End If
            Loop
            SQL_Out_File.Close
        End If
        CheckSQLOutput = ErrorFlag
End Function
'-----------------------------------------
'--- end function
'-----------------------------------------
'-------------------------------------------------------------
'--- define function to prompt for user input if necessary
'-------------------------------------------------------------
Function GetUserInput(UserInput)
    Select Case UserInput
        Case "ServerName"
            '--- pre-fill the prompt with the machine name
            TempServerName = WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%")
            '--- prompt the use for the setup particulars
            TempResponse = InputBox("Enter your server name:", "TPC-C Setup", TempServerName)
            Do While TempResponse = ""
                rc = MsgBox("You must enter a valid server name.", 21)
                If rc = 2 Then
                    wScript.Echo ""
                    wScript.Echo "TPC-C Setup cancelled by user."
                    wScript.Quit
                End If
                TempResponse = WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%")
                TempResponse = InputBox("Enter your server name:", "TPC-C Setup", TempServerName)
            Loop
        Case "SQLUserID"
            TempResponse = InputBox("Enter the SQL Server User ID to use:" & _
                                    Chr(13) & "Note: You must use SQL Authentication for TPC-C", "TPC-C Setup", "sa")
            Do While TempResponse = ""
                rc = MsgBox("You must enter a valid SQL Server User ID.", 21)
                If rc = 2 Then
                    wScript.Echo ""
                    wScript.Echo "TPC-C Setup cancelled by user."
                    wScript.Quit
                End If
                TempResponse = InputBox("Enter the SQL Server User ID to use:", "TPC-C Setup", "sa")
            Loop
        Case "saPassword"
            TempResponse = InputBox("Enter the SQL Server password for " & SQLUserID & ":", "TPC-C Setup")
        Case "NumberWarehouses"
            TempResponse = InputBox("Enter the number of warehouses to build:", "TPC-C Setup")
            Do While TempResponse = ""
                rc = MsgBox("You must enter a value for Number of Warehouses.", 21)
                If rc = 2 Then
                    wScript.Echo ""
                    wScript.Echo "TPC-C Setup cancelled by user."
                    wScript.Quit
                End If
                TempResponse = InputBox("Enter the number of warehouses to build:", "TPC-C Setup")
            Loop
        Case "BuildOption"
            TempResponse = InputBox("Build Option:" & Chr(13) & "(full,builddb,objects,objectsfull,bulkload,bulkloadfull,backup)", "TPC-C Setup", "full")
            TempResponse = LCase(TempResponse)
            Flag = 0
            Do While Flag = 0
                Select Case TempResponse
                    Case "full"
                        'TempResponse = "full"
                        Flag = 1
                    Case "builddb"
                        'TempResponse = "builddb"
                        Flag = 1
                    Case "objects"
                        'TempResponse = "objects"
                        Flag = 1
                    Case "objectsfull"
                        'TempResponse = "objectsfull"
                        Flag = 1
                    Case "bulkload"
                        'TempResponse = "bulkload"
                        Flag = 1
                    Case "bulkloadfull"
                        'TempResponse = "bulkloadfull"
                        Flag = 1
                    Case "backup"
                        'TempResponse = "backup"
                        Flag = 1
                    Case Else
                        rc = MsgBox("Invalid Database Build Option.", 21)
                        If rc = 2 Then
                            wScript.Echo ""
                            wScript.Echo "TPC-C Setup cancelled by user."
                            wScript.Quit
                        End If
                        Flag = 0
                        TempResponse = InputBox("Build Option:" & Chr(13) & "(full,builddb,objects,objectsfull,bulkload,bulkloadfull,backup)", , "full")
                        TempResponse = LCase(TempResponse)
                End Select
            Loop
```

```
        Case "DatabaseType"
            TempResponse = InputBox("Database Type:" & Chr(13) & "(normal or scale_down)", "TPC-C Setup", "normal")
            TempResponse = LCase(TempResponse)
            '--- set flag
            Flag = 0
            Do While Flag = 0
                Select Case TempResponse
                    Case "normal"
                        TempResponse = "0"
                        Flag = 1
                    Case "scale_down"
                        TempResponse = "1"
                        Flag = 1
                    Case Else
                        rc = MsgBox("Invalid Database Type.", 21)
                        If rc = 2 Then
                            wScript.Echo ""
                            wScript.Echo "TPC-C Setup cancelled by user."
                            wScript.Quit
                        End If
                        Flag = 0
                        TempResponse = InputBox("Database Type:" & Chr(13) & "(normal or scale_down)", , "normal")
                        TempResponse = LCase(TempResponse)
                End Select
            Loop
            Case "UnattendedBuild"
            TempResponse = InputBox("Unattended Build?:" & Chr(13) & "(true or false)", "TPC-C Setup", "false")
            TempResponse = LCase(TempResponse)
            '--- set flag
            Flag = 0
            Do While TempResponse = ""
                rc = MsgBox("You must enter true or false for Unattended Build.", 21)
                If rc = 2 Then
                    wScript.Echo ""
                    wScript.Echo "TPC-C Setup cancelled by user."
                    wScript.Quit
                End If
                TempResponse = InputBox("Unattended Build?:" & Chr(13) & "(true or false)", "TPC-C Setup", "false")
                TempResponse = LCase(TempResponse)
            Loop
    End Select
    GetUserInput = TempResponse
End Function
'-------------------------------------------
'--- end function
'-------------------------------------------
'-----------------------------------------------------
'--- Initialize an array of the TPC-C table names
'-----------------------------------------------------
Dim TableArray(9)
TableArray(0) = "warehouse"
TableArray(1) = "district"
TableArray(2) = "customer"
TableArray(3) = "history"
TableArray(4) = "new_order"
TableArray(5) = "orders"
TableArray(6) = "order_line"
TableArray(7) = "item"
TableArray(8) = "stock"
TableArray(9) = "tpccldr"
'-----------------------------------------------------------
'--- Initialize an array of the TPC-C build log file names
'-----------------------------------------------------------
Dim LogFileArray(21)
LogFileArray(0) = "version.log"
LogFileArray(1) = "removedb.log"
LogFileArray(2) = "createdb.log"
LogFileArray(3) = "tables.log"
LogFileArray(4) = "dbopt1.log"
LogFileArray(5) = "idxordcl.log"
LogFileArray(6) = "idxitmcl.log"
LogFileArray(7) = "idxwarcl.log"
LogFileArray(8) = "idxcuscl.log"
LogFileArray(9) = "idxnodcl.log"
LogFileArray(10) = "idxdiscl.log"
LogFileArray(11) = "idxstkcl.log"
LogFileArray(12) = "idxodlcl.log"
LogFileArray(13) = "idxcusnc.log"
LogFileArray(14) = "idxhiscl.log"
LogFileArray(15) = "idxordnc.log"
LogFileArray(16) = "bulkload.log"
LogFileArray(17) = "dbopt2.log"
LogFileArray(18) = "nurand_load.log"
LogFileArray(19) = "backupdev.log"
LogFileArray(20) = "backupdev.log"
LogFileArray(21) = "verifyload.log"
'-------------------------------------------
'--- open a file system object
'-------------------------------------------
Set fs = CreateObject("Scripting.FileSystemObject")
'-------------------------------------------
'--- grab the current directory value
'-------------------------------------------
SetupDirectory = WshShell.CurrentDirectory & "\"
'-------------------------------------------
'--- now calculate the other directories
'-------------------------------------------
ACIDDirectory = Left(SetupDirectory, (Len(SetupDirectory) - 6))
ScriptDirectory = SetupDirectory & "SCRIPTS\"
LogDirectory = SetupDirectory & "LOGS\"
'-----------------------------------------------------------------------------
'--- first see if the user passed a "?" as the first parameter
'--- if they did, then show the usage data
'-----------------------------------------------------------------------------
If ObjArgs.Length > 0 Then
                    If ObjArgs(0) = "?"  or ObjArgs(0) = "/?" Then
                        Call ShowUsage
                    End If
End If
'-----------------------------------------------------------------------------
'--- now determine if the user passed us any parameters.
'--- the order should be ServerName, sa Password, Number of Warehouses,
'--- Build Option, and Database Type
'-----------------------------------------------------------------------------
Select Case ObjArgs.Length
    Case 0
        '-------------------------------------------
        '--- get the server name
        '-------------------------------------------
        ServerName = GetUserInput("ServerName")
        If ServerName = "" Then
            ServerName = WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%")
        End If
        '-------------------------------------------
        '--- get the SQL Server user id
        '-------------------------------------------
        SQLUserID = GetUserInput("SQLUserID")
        '-------------------------------------------
        '--- get the sa password
        '-------------------------------------------
        saPassword = GetUserInput("saPassword")
        '-------------------------------------------
        '--- get the number of warehouses
        '-------------------------------------------
        NumberWarehouses = GetUserInput("NumberWarehouses")
        '-------------------------------------------
        '--- get the build option
        '-------------------------------------------
        BuildOption = GetUserInput("BuildOption")
        '-------------------------------------------
        '--- get the database type
        '-------------------------------------------
        DatabaseType = (GetUserInput("DatabaseType"))
        If DatabaseType = "scale_down" Then
            DatabaseType = 1
        Else
            DatabaseType = 0
        End If
        '-------------------------------------------
        '--- unattended install
        '-------------------------------------------
        UnattendedBuild = GetUserInput("UnattendedBuild")
```

```
Case 1
    '-------------------------------------------------------
    '--- assume that the server name was passed correctly
    '-------------------------------------------------------
    '-----------------------------------------
    '--- store the server name
    '-----------------------------------------
    ServerName = ObjArgs(0)
    If ServerName = "" Then
        ServerName = WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%")
    End If
    '-----------------------------------------
    '--- get the SQL Server user id
    '-----------------------------------------
    SQLUserID = GetUserInput("SQLUserID")
    '-----------------------------------------
    '--- get the sa password
    '-----------------------------------------
    saPassword = GetUserInput("saPassword")
    '-----------------------------------------
    '--- get the number of warehouses
    '-----------------------------------------
    NumberWarehouses = GetUserInput("NumberWarehouses")
    '-----------------------------------------
    '--- get the build option
    '-----------------------------------------
    BuildOption = GetUserInput("BuildOption")
    '-----------------------------------------
    '--- get the database type
    '-----------------------------------------
    DatabaseType = (GetUserInput("DatabaseType"))
    If DatabaseType = "scale_down" Then
        DatabaseType = 1
    Else
        DatabaseType = 0
    End If
    '-----------------------------------------
    '--- unattended install
    '-----------------------------------------
    UnattendedBuild = GetUserInput("UnattendedBuild")
Case 2
    '-------------------------------------------------------------------------
    '--- assume that the server name and SQL Server user ID was passed correctly
    '-------------------------------------------------------------------------
    '-----------------------------------------
    '--- store the server name
    '-----------------------------------------
    ServerName = ObjArgs(0)
    If ServerName = "" Then
        ServerName = WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%")
    End If
    '-----------------------------------------
    '--- store the SQL Server user id
    '-----------------------------------------
    SQLUserID = ObjArgs(1)
    '-----------------------------------------
    '--- store the sa password
    '-----------------------------------------
    saPassword = ObjArgs(2)
    '-----------------------------------------
    '--- get the number of warehouses
    '-----------------------------------------
    NumberWarehouses = GetUserInput("NumberWarehouses")
    '-----------------------------------------
    '--- get the build option
    '-----------------------------------------
    BuildOption = GetUserInput("BuildOption")
    '-----------------------------------------
    '--- get the database type
    '-----------------------------------------
    DatabaseType = (GetUserInput("DatabaseType"))
    If DatabaseType = "scale_down" Then
        DatabaseType = 1
    Else
        DatabaseType = 0
    End If
    '-----------------------------------------
    '--- unattended install
    '-----------------------------------------
    UnattendedBuild = GetUserInput("UnattendedBuild")
Case 3
    '-------------------------------------------------------------------------------------------
    '--- assume that the server name, SQL Server user ID, SQL Server password were passed correctly
    '-------------------------------------------------------------------------------------------
    '-----------------------------------------
    '--- store the server name
    '-----------------------------------------
    ServerName = ObjArgs(0)
    If ServerName = "" Then
        ServerName = WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%")
    End If
    '-----------------------------------------
    '--- store the SQL Server user id
    '-----------------------------------------
    SQLUserID = ObjArgs(1)
    '-----------------------------------------
    '--- store the sa password
    '-----------------------------------------
    saPassword = ObjArgs(2)
    '-----------------------------------------
    '--- store the number of warehouses
    '-----------------------------------------
    NumberWarehouses = ObjArgs(3)
    '-----------------------------------------
    '--- get the build option
    '-----------------------------------------
    BuildOption = GetUserInput("BuildOption")
    '-----------------------------------------
    '--- get the database type
    '-----------------------------------------
    DatabaseType = (GetUserInput("DatabaseType"))
    If DatabaseType = "scale_down" Then
        DatabaseType = 1
    Else
        DatabaseType = 0
    End If
    '-----------------------------------------
    '--- unattended install
    '-----------------------------------------
    UnattendedBuild = GetUserInput("UnattendedBuild")
Case 4
    '-------------------------------------------------------------------------------------------------------
    '--- assume that the server name, SQL Server user ID, SQL Server password, and number of warehouses were passed correctly
    '-------------------------------------------------------------------------------------------------------
    '-----------------------------------------
    '--- store the server name
    '-----------------------------------------
    ServerName = ObjArgs(0)
    If ServerName = "" Then
        ServerName = WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%")
    End If
    '-----------------------------------------
    '--- store the SQL Server user id
    '-----------------------------------------
    SQLUserID = ObjArgs(1)
    '-----------------------------------------
    '--- store the sa password
    '-----------------------------------------
    saPassword = ObjArgs(2)
    '-----------------------------------------
    '--- store the number of warehouses
    '-----------------------------------------
    NumberWarehouses = ObjArgs(3)
    '-----------------------------------------
    '--- store the build option
    '-----------------------------------------
    BuildOption = LCase(ObjArgs(4))
    '-----------------------------------------
    '--- get the database type
    '-----------------------------------------
    DatabaseType = (GetUserInput("DatabaseType"))
    If DatabaseType = "scale_down" Then
```

```
            DatabaseType = 1
        Else
            DatabaseType = 0
        End If
        '-----------------------------------------
        '--- unattended install
        '-----------------------------------------
        UnattendedBuild = GetUserInput("UnattendedBuild")
    Case 5
        '-----------------------------------------------------------------------------------------------------------------
        '--- assume that the server name, SQL Server user ID, SQL Server password, and number of warehouses,
        '--- build option were passed correctly
        '-----------------------------------------------------------------------------------------------------------------
        '-----------------------------------------
        '--- store the server name
        '-----------------------------------------
        ServerName = ObjArgs(0)
        If ServerName = "" Then
            ServerName = WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%")
        End If
        '-----------------------------------------
        '--- store the SQL Server user id
        '-----------------------------------------
        SQLUserID = ObjArgs(1)
        '-----------------------------------------
        '--- store the sa password
        '-----------------------------------------
        saPassword = ObjArgs(2)
        '-----------------------------------------
        '--- store the number of warehouses
        '-----------------------------------------
        NumberWarehouses = ObjArgs(3)
        '-----------------------------------------
        '--- store the build option
        '-----------------------------------------
        BuildOption = LCase(ObjArgs(4))
        '-----------------------------------------
        '--- get the database type
        '-----------------------------------------
        DatabaseType = (GetUserInput("DatabaseType"))
        If DatabaseType = "scale_down" Then
            DatabaseType = 1
        Else
            DatabaseType = 0
        End If
        '-----------------------------------------
        '--- unattended install
        '-----------------------------------------
        UnattendedBuild = GetUserInput("UnattendedBuild")
    Case 6
        '-----------------------------------------------------------------------------------------------------------------
        '--- assume that the server name, SQL Server user ID, SQL Server password, and number of warehouses,
        '--- build option, and database type were passed correctly
        '-----------------------------------------------------------------------------------------------------------------
        '-----------------------------------------
        '--- store the server name
        '-----------------------------------------
        ServerName = ObjArgs(0)
        If ServerName = "" Then
            ServerName = WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%")
        End If
        '-----------------------------------------
        '--- store the SQL Server user id
        '-----------------------------------------
        SQLUserID = ObjArgs(1)
        '-----------------------------------------
        '--- store the sa password
        '-----------------------------------------
        saPassword = ObjArgs(2)
        '-----------------------------------------
        '--- store the number of warehouses
        '-----------------------------------------
        NumberWarehouses = ObjArgs(3)
        '-----------------------------------------
        '--- store the build option
        '-----------------------------------------
        BuildOption = LCase(ObjArgs(4))
        '-----------------------------------------
        '--- get the database type
        '-----------------------------------------
        DatabaseType = LCase(ObjArgs(5))
        If DatabaseType = "scale_down" Then
            DatabaseType = 1
        Else
            DatabaseType = 0
        End If
        '-----------------------------------------
        '--- unattended install
        '-----------------------------------------
        UnattendedBuild = GetUserInput("UnattendedBuild")
    Case 7
        '-----------------------------------------------------------------------------------------------------------------
        '--- assume was passed correctly
        '-----------------------------------------------------------------------------------------------------------------
        '-----------------------------------------
        '--- store the server name
        '-----------------------------------------
        ServerName = ObjArgs(0)
        If ServerName = "" Then
            ServerName = WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%")
        End If
        '-----------------------------------------
        '--- store the SQL Server user id
        '-----------------------------------------
        SQLUserID = ObjArgs(1)
        '-----------------------------------------
        '--- store the sa password
        '-----------------------------------------
        saPassword = ObjArgs(2)
        '-----------------------------------------
        '--- store the number of warehouses
        '-----------------------------------------
        NumberWarehouses = ObjArgs(3)
        '-----------------------------------------
        '--- store the build option
        '-----------------------------------------
        BuildOption = LCase(ObjArgs(4))
        '-----------------------------------------
        '--- get the database type
        '-----------------------------------------
        DatabaseType = LCase(ObjArgs(5))
        If DatabaseType = "scale_down" Then
            DatabaseType = 1
        Else
            DatabaseType = 0
        End If
        '-----------------------------------------
        '--- unattended install
        '-----------------------------------------
        UnattendedBuild = LCase(ObjArgs(6))
End Select
'-------------------------------------------------------------
'--- if the user specified a scale down database, then show
'--- them the warning message
'-------------------------------------------------------------
If DatabaseType = 1 Then
    MsgBox "WARNING!" & Chr(13) & "The Scale_Down option is to be used for functional testing only." _
        & Chr(13) & "The use of this option will not produce a valid TPC-C result.", _
        vbExclamation, "Scale-Down Warning"
End If
'-------------------------------------------------------------
'--- before we start to do anything, verify the input
'-------------------------------------------------------------
Select Case BuildOption
    Case "full"
        strBuildOpt = "Full build"
    Case "builddb"
        strBuildOpt = "Build database only"
    Case "objects"
        strBuildOpt = "Install stored procedures only"
    Case "objectsfull"
        strBuildOpt = "Install stored procedures and complete build process"
```

```
        Case "bulkload"
                strBuildOpt = "Load data only"
        Case "bulkloadfull"
                strBuildOpt = "Load data and complete build process"
        Case "backup"
                strBuildOpt = "Backup database"
End Select
If DatabaseType = 1 Then
        strDBType = "Scale Down"
Else
        strDBType = "Normal"
End If
If UnattendedBuild = "false" Then
        rc = MsgBox("The following options will be used:" & Chr(13) & Chr(10) & Chr(13) & Chr(10) _
                                        & "Database Server Name:      " & ServerName & Chr(13) & Chr(10) _
                                        & Chr(13) & Chr(10) _
                                        & "SQL Server User ID:        " & SQLUserID & Chr(13) & Chr(10) _
                                        & Chr(13) & Chr(10) _
                                        & "SQL Server Password:       " & saPassword & Chr(13) & Chr(10) _
                                        & Chr(13) & Chr(10) _
                                        & "Number of Warehouses:    " & NumberWarehouses & Chr(13) & Chr(10) _
                                        & Chr(13) & Chr(10) _
                                        & "Build Option:              " & strBuildOpt & Chr(13) & Chr(10) _
                                        & Chr(13) & Chr(10) _
                                        & "Build Type:                " & strDBType & Chr(13) & Chr(10) _
                                        & "                                                ", 65, "TPC-C Setup Option Verification")
        If rc = 2 Then
                wScript.Echo ""
                wScript.Echo "TPC-C Setup cancelled by user."
                wScript.Quit
        End If
End If
'-----------------------------------------------------------------------
'--- now that we have all the variables filled in, let's get to work
'--- cleanup any old .err files
'-----------------------------------------------------------------------
For i = 0 To 9
        If fs.FileExists(LogPath & TableArray(i) & ".err") Then
                fs.DeleteFile LogPath & TableArray(i) & ".err"
        End If
Next
For i = 0 To 21
        If fs.FileExists(LogPath & LogFileArray(i)) Then
                fs.DeleteFile LogPath & LogFileArray(i)
        End If
Next
'-----------------------------------------------------------------------
'--- now grab the version of SQL Server you are running this against
'-----------------------------------------------------------------------
Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ScriptDirectory & "utility\version.sql -o" & LogDirectory & "version.log")
Do While oExec.Status = 0
   wScript.Sleep 100
Loop
rc = CheckSQLOutput(LogDirectory & "version.log")
If rc <> 0 Then
        wScript.Quit
End If
'-----------------------------------------
'--- okay, let's do it
'-----------------------------------------
If (BuildOption = "full" _
        Or BuildOption = "builddb") Then
        wScript.Echo "Removing any existing TPCC database and backup devices..."
        Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ScriptDirectory & NumberWarehouses & ".war\database\removedb.sql -o" & LogDirectory & "removedb.log")
        Do While oExec.Status = 0
                wScript.Sleep 100
        Loop
        rc = CheckSQLOutput(LogDirectory & "removedb.log")
        If rc <> 0 Then
                wScript.Quit
        End If
        wScript.Echo "Building database files and database..."
        Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ScriptDirectory & NumberWarehouses & ".war\database\createdb.sql -o" & LogDirectory & "createdb.log")
        Do While oExec.Status = 0
                wScript.Sleep 100
        Loop
        rc = CheckSQLOutput(LogDirectory & "createdb.log")
        If rc <> 0 Then
                wScript.Quit
        End If
End If
'-----------------------------------------
'--- build tables and stored procedures
'-----------------------------------------
If (BuildOption = "full" _
        Or BuildOption = "objects" _
     Or BuildOption = "objectsfull") Then
        wScript.Echo "Creating TPC-C database tables..."
        Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ScriptDirectory & NumberWarehouses & ".war\ddl\tables.sql -o" & LogDirectory & "tables.log")
        Do While oExec.Status = 0
                wScript.Sleep 100
        Loop
        rc = CheckSQLOutput(LogDirectory & "tables.log")
        If rc <> 0 Then
                wScript.Quit
        End If
        wScript.Echo "Creating database objects..."
        Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ScriptDirectory & "dml\neword.sql -o" & LogDirectory & "sp_neword.log")
        Do While oExec.Status = 0
                wScript.Sleep 100
        Loop
        rc = CheckSQLOutput(LogDirectory & "sp_neword.log")
        If rc <> 0 Then
                wScript.Quit
        End If
        Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ScriptDirectory & "dml\payment.sql -o" & LogDirectory & "sp_payment.log")
        Do While oExec.Status = 0
                wScript.Sleep 100
        Loop
        rc = CheckSQLOutput(LogDirectory & "sp_payment.log")
        If rc <> 0 Then
                wScript.Quit
        End If
        Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ScriptDirectory & "dml\ordstat.sql -o" & LogDirectory & "sp_ordstat.log")
        Do While oExec.Status = 0
                wScript.Sleep 100
        Loop
        rc = CheckSQLOutput(LogDirectory & "sp_ordstat.log")
        If rc <> 0 Then
                wScript.Quit
        End If
        Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ScriptDirectory & "dml\delivery.sql -o" & LogDirectory & "sp_delivery.log")
        Do While oExec.Status = 0
                wScript.Sleep 100
        Loop
        rc = CheckSQLOutput(LogDirectory & "sp_delivery.log")
        If rc <> 0 Then
                wScript.Quit
        End If
        Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ScriptDirectory & "dml\stocklev.sql -o" & LogDirectory & "sp_stocklev.log")
        Do While oExec.Status = 0
                wScript.Sleep 100
        Loop
        rc = CheckSQLOutput(LogDirectory & "sp_stocklev.log")
        If rc <> 0 Then
                wScript.Quit
        End If
        Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ScriptDirectory & "dml\version.sql -o" & LogDirectory & "sp_version.log")
        Do While oExec.Status = 0
                wScript.Sleep 100
        Loop
        rc = CheckSQLOutput(LogDirectory & "sp_version.log")
        If rc <> 0 Then
                wScript.Quit
        End If
        wScript.Echo "Database object creation complete..."
End If
If (BuildOption = "full" _
        Or BuildOption = "objectsfull" _
     Or BuildOption = "bulkload" _
     Or BuildOption = "bulkloadfull") Then
        wScript.Echo "Setting database options before load..."
```

```
    Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ScriptDirectory & "utility\dbopt1.sql -o" & LogDirectory & "dbopt1.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory & "dbopt1.log")
    If rc <> 0 Then
        wScript.Quit
    End If
    '-------------------------------------------------------------
    '--- before we start tpccldr.exe, check the registry
    '--- to ensure that the Shared Memory Protocol is off.
    '--- if it is on, store the setting so we can return
    '--- the system to the pre-tpccldr state.
    '-------------------------------------------------------------
    SharedMemoryRegKey = WshShell.RegRead("HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SharedMemoryOn")
    If SharedMemoryRegKey = 1 Then
        WshShell.RegWrite "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SharedMemoryOn", 0, "REG_DWORD"
    End If
    If Platform = "ia64" Then
        wScript.Echo "Loading data and creating indexes..."
        wScript.Echo "(This runs in a separate, minimized window.)"
        CMD_String = SetupDirectory & "loader\bin\ia64\tpccldr.exe"
        CMD_String = CMD_String & " -S" & ServerName
        CMD_String = CMD_String & " -U" & SQLUserID
        CMD_String = CMD_String & " -P" & saPassword
        CMD_String = CMD_String & " -W" & NumberWarehouses
        CMD_String = CMD_String & " -f" & LogDirectory & "bulkload.log"
        CMD_String = CMD_String & " -L" & LogDirectory
        CMD_String = CMD_String & " -d" & ScriptDirectory & NumberWarehouses & ".war\ddl"
        CMD_String = CMD_String & " -c" & DatabaseType
    Else
        wScript.Echo "Loading data and creating indexes..."
        wScript.Echo "(This runs in a separate, minimized window.)"
        CMD_String = SetupDirectory & "loader\bin\x86\tpccldr.exe"
        CMD_String = CMD_String & " -S" & ServerName
        CMD_String = CMD_String & " -U" & SQLUserID
        CMD_String = CMD_String & " -P" & saPassword
        CMD_String = CMD_String & " -W" & NumberWarehouses
        CMD_String = CMD_String & " -f" & LogDirectory & "bulkload.log"
        CMD_String = CMD_String & " -L" & LogDirectory
        CMD_String = CMD_String & " -d" & ScriptDirectory & NumberWarehouses & ".war\ddl"
        CMD_String = CMD_String & " -c" & DatabaseType
    End If
    oExec = WshShell.Run(CMD_String, 2, True)
    If oExec <> 0 Then
        wScript.Echo "The TPCCLDR.EXE encountered an error."
        wScript.Echo "Check logs\TPCCLDR.ERR for details."
        wScript.Quit
    End If
    '-------------------------------------------------------------
    '--- now that the loader is finished, put the
    '--- SharedMemoryOn registry key back to its original
    '--- value.
    '-------------------------------------------------------------
    If SharedMemoryRegKey = 1 Then
        WshShell.RegWrite "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SharedMemoryOn", 1, "REG_DWORD"
    End If
    wScript.Echo "Setting database options after load..."
    Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ScriptDirectory & "utility\dbopt2.sql -o" & LogDirectory & "dbopt2.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory & "dbopt2.log")
    If rc <> 0 Then
        wScript.Quit
    End If
    wScript.Echo "Data load and index creation complete."
    '-----------------------------------------
    '--- now parse the index creation logs
    '--- to see if there were any errors
    '--- there.
    '-----------------------------------------
    For i = 5 To 15
        rc = CheckSQLOutput(LogDirectory & LogFileArray(i))
        If rc <> 0 Then
            wScript.Quit
        End If
    Next
    wScript.Echo "Calculating initial database space usage...."
    Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ACIDDirectory & "space\scripts\spused.sql -o" & ACIDDirectory & "space\spused.ver")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ACIDDirectory & "space\scripts\splog.sql -o" & ACIDDirectory & "space\splog.ver")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ACIDDirectory & "space\scripts\spfiles.sql -o" & ACIDDirectory & "space\spfiles.ver")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    '-----------------------------------------
    '--- now that the loader is finished
    '--- check the .err files and if they
    '--- are of zero length, delete them.
    '-----------------------------------------
    Set fsErr = CreateObject("Scripting.FileSystemObject")
    Set fErr = fsErr.GetFolder(LogDirectory)
    Set fcErr = fErr.Files
    For Each f1 In fcErr
        If f1.Type = "ERR File" Then
            If f1.Size = 0 Then
                f1.Delete
            End If
        End If
    Next
    Set fcErr = Nothing
    Set fErr = Nothing
    Set fsErr = Nothing
End If
If (BuildOption = "full" _
   Or BuildOption = "objectsfull" _
   Or BuildOption = "bulkloadfull" _
   Or BuildOption = "backup") Then
    wScript.Echo "Creating Backup Device(s)..."
    Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ScriptDirectory & NumberWarehouses & ".war\database\backupdev.sql -o" & LogDirectory & "backupdev.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory & "backupdev.log")
    If rc <> 0 Then
        wScript.Quit
    End If
    wScript.Echo "Backing up database..."
    Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ScriptDirectory & NumberWarehouses & ".war\database\backup.sql -o" & LogDirectory & "backup.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory & "backup.log")
    If rc <> 0 Then
        wScript.Quit
    End If
    wScript.Echo "Database backup complete."
End If
If (BuildOption = "full" _
   Or BuildOption = "objectsfull" _
   Or BuildOption = "bulkloadfull") Then
    wScript.Echo "Verifying TPC-C database load..."
    Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P" & saPassword & " -S" & ServerName & " -e -i" & ScriptDirectory & "utility\verifytpccload.sql -o" & LogDirectory & "verifyload.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory & "verifyload.log")
    If rc <> 0 Then
        wScript.Quit
    End If
    wScript.Echo "Check logs\verifyload.log to verify database load."
End If
'-----------------------------------------
'--- display banner message
'-----------------------------------------
wScript.Echo "*********************************************************************************************"
```

```
wScript.Echo "*                                                          *"
wScript.Echo "*  Microsoft TPC-C Benchmark Kit Ver." & Kit_Version & "                    *"
wScript.Echo "*                                                          *"
wScript.Echo "*  Database Setup Complete                                 *"
wScript.Echo "*                                                          *"
wScript.Echo "************************************************************"
wScript.Quit
'-------------------------------------------------------
'--- ShowUsage
'-------------------------------------------------------
Function ShowUsage()
	wScript.Echo "************************************************************"
	wScript.Echo "*                                                          *"
	wScript.Echo "*  Microsoft TPC-C Benchmark Kit Ver." & Kit_Version & "                    *"
	wScript.Echo "*                                                          *"
	wScript.Echo "*  Usage:                                                  *"
	wScript.Echo "*     Optionally, you can pass the following positional arguments to SETUP *"
	wScript.Echo "*        - Server Name (can be """" for local host)          *"
	wScript.Echo "*        - SQL Server User ID (recommended you use sa)       *"
	wScript.Echo "*        - SQL Sever Account Password                        *"
	wScript.Echo "*        - Number of Warehouses to Build                     *"
	wScript.Echo "*        - Build Option                                      *"
	wScript.Echo "*          [full,builddb,objects,objectsfull,bulkload,bulkloadfull,backup] *"
	wScript.Echo "*        - Database Type                                     *"
	wScript.Echo "*          [normal or scale_down]                            *"
	wScript.Echo "*        - Unattended Install                                *"
	wScript.Echo "*          [true or false]                                   *"
	wScript.Echo "*                                                          *"
	wScript.Echo "*     If you do not pass the parameters, then you will be prompted for the *"
	wScript.Echo "*     information by the application.                        *"
	wScript.Echo "*                                                          *"
	wScript.Echo "************************************************************"
	wScript.Quit
End Function
```

Kit MSTPCC451\SETUP\scripts\utility\RunSQLCfg.sql

```
--------------------------------------------------------------
--                                                          --
--          File:          RUNSQLCFG.SQL                    --
--                         Microsoft TPC-C Benchmark Kit Ver. 4.51  --
--                         Copyright Microsoft, 2003        --
--                                                          --
--                         Sets suggested runtime server configuration --
--                         parameters                       --
--                                                          --
--------------------------------------------------------------

exec sp_configure 'show advanced option', 1
go

reconfigure with override
go

exec sp_configure 'max server memory',62000
exec sp_configure 'min server memory',60000
exec sp_configure 'network packet size',4096

exec sp_configure 'max degree of parallelism',1

---------------------------------------------------------------------
-- change this value to approximately the number of connected users
---------------------------------------------------------------------
exec sp_configure 'max worker threads',400

---------------------------------------------------------------------
-- increase priority of user threads
---------------------------------------------------------------------
exec sp_configure 'priority boost',1

---------------------------------------------------------------------
-- disable automatic checkpointing
---------------------------------------------------------------------
exec sp_configure 'recovery interval',118

---------------------------------------------------------------------
-- change to a mask appropriate for the number of processors on the server
---------------------------------------------------------------------
exec sp_configure 'affinity mask',0xff

---------------------------------------------------------------------
-- enable fibers
---------------------------------------------------------------------
exec sp_configure 'lightweight pooling',1

go

reconfigure with override
go
```

Kit MSTPCC451\SETUP\scripts\14000.war\ddl\idxcuscl.sql

```
--------------------------------------------------------------
--                                                          --
--  File:  IDXCUSCL.SQL                                     --
--        Microsoft TPC-C Benchmark Kit Ver. 4.51           --
--        Copyright Microsoft, 2003                         --
--                                                          --
--        Creates clustered index on customer table         --
--------------------------------------------------------------

use tpcc
go

declare		@startdate	datetime,
		@enddate	datetime

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

if exists ( select name from sysindexes where name = '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, 21)
select		'Elapsed time (in seconds): ',
			datediff(second, @startdate, @enddate)
go
```

Kit MSTPCC451\SETUP\scripts\14000.war\ddl\idxcusnc.sql

```
--------------------------------------------------------------
--                                                          --
--  File:  IDXCUSNC.SQL                                     --
--        Microsoft TPC-C Benchmark Kit Ver. 4.51           --
--        Copyright Microsoft, 2003                         --
--                                                          --
--        Creates non-clustered index on customer table     --
--------------------------------------------------------------

use tpcc
go

declare		@startdate	datetime,
		@enddate	datetime

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

if exists ( select name from sysindexes where  name = '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)
```

```
                        with SORT_IN_TEMPDB
                        on MSSQL_cs_fg

select                  @enddate = getdate()
select                  'End date: ',
                                convert(varchar(30),@enddate, 21)
select                  'Elapsed time (in seconds): ',
                                datediff(second, @startdate, @enddate)
go
```

Kit MSTPCC451\SETUP\scripts\14000.war\ddl\idxdiscl.sql

```
-------------------------------------------------------------------
--                                                               --
--  File:  IDXDISCL.SQL                                          --
--       Microsoft TPC-C Benchmark Kit Ver. 4.51                 --
--       Copyright Microsoft, 2003                               --
--                                                               --
--       Creates clustered index on district table               --
-------------------------------------------------------------------

use tpcc
go

declare                 @startdate              datetime,
                        @enddate                datetime

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

if exists ( select name from sysindexes where name = '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, 21)
select                  'Elapsed time (in seconds): ',
                                datediff(second, @startdate, @enddate)
go
```

Kit MSTPCC451\SETUP\scripts\14000.war\ddl\idxhiscl.sql

```
-------------------------------------------------------------------
--                                                               --
--  File:  IDXHISCL.SQL                                          --
--       Microsoft TPC-C Benchmark Kit Ver. 4.51                 --
--       Copyright Microsoft, 2003                               --
--                                                               --
--       Creates clustered index on history table                --
--                                                               --
--       CAUTION:  This index is only beneficial for systems     --
--       CAUTION:  with 8 or more processors.                    --
--       CAUTION:  It may negatively impact performance on       --
--       CAUTION:  systems with less than 8 processors.          --
--                                                               --
-------------------------------------------------------------------

use tpcc
go

declare                 @startdate              datetime,
                        @enddate                datetime

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

if exists ( select name from sysindexes where  name = '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, 21)
select                  'Elapsed time (in seconds): ',
                                datediff(second, @startdate, @enddate)
go
```

Kit MSTPCC451\SETUP\scripts\14000.war\ddl\idxitmcl.sql

```
-------------------------------------------------------------------
--                                                               --
--  File:  IDXITMCL.SQL                                          --
--       Microsoft TPC-C Benchmark Kit Ver. 4.51                 --
--       Copyright Microsoft, 2003                               --
--                                                               --
--       Creates clustered index on item table                   --
-------------------------------------------------------------------

use tpcc
go

declare                 @startdate              datetime,
                        @enddate                datetime

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

if exists ( select name from sysindexes where  name = '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, 21)
select                  'Elapsed time (in seconds): ',
                                datediff(second, @startdate, @enddate)
go
```

Kit MSTPCC451\SETUP\scripts\14000.war\ddl\idxnodcl.sql

```
-------------------------------------------------------------------
--                                                               --
--  File:  IDXNODCL.SQL                                          --
--       Microsoft TPC-C Benchmark Kit Ver. 4.51                 --
--       Copyright Microsoft, 2003                               --
--                                                               --
--       Creates clustered index on new_order table              --
-------------------------------------------------------------------

use tpcc
go

declare                 @startdate              datetime,
                        @enddate                datetime

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

if exists ( select name from sysindexes where  name = '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)
  with pad_index, fillfactor=90
  on MSSQL_misc_fg
```

```
select          @enddate = getdate()
select          'End date: ',
                        convert(varchar(30),@enddate, 21)
select          'Elapsed time (in seconds): ',
                        datediff(second, @startdate, @enddate)
go
```

Kit MSTPCC451\SETUP\scripts\14000.war\ddl\idxodlcl.sql

```
------------------------------------------------------------------
--                                                              --
--  File:  IDXODLCL.SQL                                         --
--      Microsoft TPC-C Benchmark Kit Ver. 4.51                 --
--      Copyright Microsoft, 2003                               --
--                                                              --
--      Creates clustered index on order_line table             --
------------------------------------------------------------------

use tpcc
go

declare         @startdate      datetime,
                @enddate        datetime

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

if exists ( select name from sysindexes where  name = '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, 21)
select          'Elapsed time (in seconds): ',
                        datediff(second, @startdate, @enddate)
go
```

Kit MSTPCC451\SETUP\scripts\14000.war\ddl\idxordcl.sql

```
------------------------------------------------------------------
--                                                              --
--  File:  IDXORDCL.SQL                                         --
--      Microsoft TPC-C Benchmark Kit Ver. 4.51                 --
--      Copyright Microsoft, 2003                               --
--                                                              --
--      Creates clustered index on orders table                 --
------------------------------------------------------------------

use tpcc
go

declare         @startdate      datetime,
                @enddate        datetime

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

if exists ( select name from sysindexes where  name = 'orders_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, 21)
select          'Elapsed time (in seconds): ',
                        datediff(second, @startdate, @enddate)
go
```

Kit MSTPCC451\SETUP\scripts\14000.war\ddl\idxstkcl.sql

```
------------------------------------------------------------------
--                                                              --
--  File:  IDXSTKCL.SQL                                         --
--      Microsoft TPC-C Benchmark Kit Ver. 4.51                 --
--      Copyright Microsoft, 2003                               --
--                                                              --
--      Creates clustered index on stock table                  --
------------------------------------------------------------------

use tpcc
go

declare         @startdate      datetime,
                @enddate        datetime

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

if exists ( select name from sysindexes where  name = '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, 21)
select          'Elapsed time (in seconds): ',
                        datediff(second, @startdate, @enddate)
go
```

Kit MSTPCC451\SETUP\scripts\14000.war\ddl\idxwarcl.sql

```
------------------------------------------------------------------
--                                                              --
--  File:  IDXWARCL.SQL                                         --
--      Microsoft TPC-C Benchmark Kit Ver. 4.51                 --
--      Copyright Microsoft, 2003                               --
--                                                              --
--      Creates clustered index on warehouse table              --
------------------------------------------------------------------

use tpcc
go

declare         @startdate      datetime,
                @enddate        datetime

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

if exists ( select name from sysindexes where name = '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, 21)
select          'Elapsed time (in seconds): ',
                        datediff(second, @startdate, @enddate)
go
```

Kit MSTPCC451\SETUP\scripts\14000.war\ddl\tables.sql

```
-------------------------------------------------------------------------
--                                                                     --
--  File:  TABLES.SQL                                                  --
--         Microsoft TPC-C Benchmark Kit Ver. 4.51                     --
--         Copyright Microsoft, 2003                                   --
--                                                                     --
--         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                                                  text
) on MSSQL_cs_fg
  textimage_on MSSQL_cust_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
```

Kit MSTPCC451\SETUP\scripts\utility\dbopt1.sql

```
-------------------------------------------------------------------
--                                                                 --
--              File:           DBOPT1.SQL                                 --
--                              Microsoft TPC-C Benchmark Kit Ver. 4.51    --
--                              Copyright Microsoft, 2003          --
--                                                                 --
--                              Sets database options for load     --
--                                                                 --
-------------------------------------------------------------------

use master
go

ALTER DATABASE tpcc SET RECOVERY BULK_LOGGED
go

ALTER DATABASE tpcc SET TORN_PAGE_DETECTION OFF
go

use tpcc
go

checkpoint
go
```

Kit MSTPCC451\SETUP\scripts\utility\dbopt2.sql

```
-------------------------------------------------------------------
--                                                                 --
--              File:           DBOPT2.SQL                                 --
--                              Microsoft TPC-C Benchmark Kit Ver. 4.51    --
--                              Copyright Microsoft, 2003          --
--                                                                 --
--                              Sets database options after data load      --
--                                                                 --
-------------------------------------------------------------------

ALTER DATABASE tpcc SET RECOVERY FULL
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
```

Kit MSTPCC451\SETUP\scripts\utility\VerifyTpccLoad.sql

```
-----------------------------------------------------------------------
--                                                                     --
--              File:           VERIFYTPCCLOAD.SQL                     --
--                              Microsoft TPC-C Benchmark Kit Ver. 4.51 --
--                              Copyright Microsoft, 2003              --
--                                                                     --
--                              Performs series of TPCC database checks to --
--                              verify that database load completed correctly --
--                                                                     --
-----------------------------------------------------------------------

print           ' '
select          convert(char(30), getdate(), 21)
print           ' '

use tpcc
go

----------------------
-- Check rows per table
----------------------
print           'WAREHOUSE TABLE'
select          count_big(*)
from            warehouse
go

print           'DISTRICT TABLE = (10 * No of warehouses)'
select          count_big(*)
from            district
go

print           'ITEM TABLE = 100,000'
select          count_big(*)
from            item
go

print           'CUSTOMER TABLE = (30,000 * No of warehouses)'
select          count_big(*)
from            customer
go

print           'ORDERS TABLE = (30,000 * No of warehouses)'
select          count_big(*)
from            orders
go

print           'HISTORY TABLE = (30,000 * No of warehouses)'
select          count_big(*)
from            history
go

print           'STOCK TABLE = (100,000 * No of warehouses)'
select          count_big(*)
from            stock
go

print           'ORDER_LINE TABLE = (300,000 * No of warehouses + some change)'
select          count_big(*)
from            order_line
go

print           'NEW_ORDER TABLE = (9000 * No of warehouses)'
select          count_big(*)
from            new_order
go

----------------
-- Check indices
----------------
print '---------- Index Check ----------'

use tpcc
go

sp_helpindex    customer
go

sp_helpindex    history
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
```

Kit MSTPCC451\SETUP\scripts\14000.war\database\backup.sql

```
-----------------------------------------------------------------------
--                                                                   --
--  File:  BACKUP.SQL                                                --
--      Microsoft TPC-C Benchmark Kit Ver. 4.51                      --
--      Copyright Microsoft, 2003                                    --
--                                                                   --
-----------------------------------------------------------------------

declare         @startdate      datetime,
                @enddate        datetime

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

dump database tpcc to tpccdurb_1,tpccdurb_2,tpccdurb_3,tpccdurb_4,tpccdurb_5,
tpccdurb_6 with init, stats = 1

select          @enddate = getdate()
select          'End date: ',
                                convert(varchar(30),@enddate, 21)
select          'Elapsed time (in seconds): ',
                                datediff(second, @startdate, @enddate)
go
```

Kit MSTPCC451\SETUP\scripts\14000.war\database\backupdev.sql

```
-----------------------------------------------------------------------
--                                                                   --
--  File:  BACKUPDEV.SQL                                             --
--      Microsoft TPC-C Benchmark Kit Ver. 4.51                      --
--      Copyright Microsoft, 2003                                    --
--                                                                   --
-----------------------------------------------------------------------
```

```
use master
go

------------------------
-- create backup devices
------------------------

exec sp_addumpdevice 'disk','tpccdurb_1','C:\MSSQL\TPCC_14000W\back7\tpccdur7.dmp'
go
exec sp_addumpdevice 'disk','tpccdurb_2','C:\MSSQL\TPCC_14000W\back8\tpccdur8.dmp'
go
exec sp_addumpdevice 'disk','tpccdurb_3','C:\MSSQL\TPCC_14000W\back9\tpccdur9.dmp'
go
exec sp_addumpdevice 'disk','tpccdurb_4','C:\MSSQL\TPCC_14000W\back10\tpccdur10.dmp'
go
exec sp_addumpdevice 'disk','tpccdurb_5','C:\MSSQL\TPCC_14000W\back11\tpccdur11.dmp'
go
exec sp_addumpdevice 'disk','tpccdurb_6','C:\MSSQL\TPCC_14000W\back12\tpccdur12.dmp'
go
```

Kit MSTPCC451\SETUP\scripts\14000.war\database\createdb.sql

```
-----------------------------------------------------------------
--                                                                          --
--        File:        CREATEDB.SQL                                         --
--                     Microsoft TPC-C Benchmark Kit Ver. 4.50              --
--                     Copyright Microsoft, 2003                            --
--                                                                          --
--                     Creates 1400 warehouse database                      --
--                                                                          --
-----------------------------------------------------------------

SET ANSI_NULL_DFLT_OFF ON
go

use master
go

-----------------------------------
-- Create temporary table for timing
-----------------------------------
if exists ( select name from sysobjects where name = 'tpcc_timer' )
        drop table tpcc_timer
go

create table tpcc_timer
        (start_date     char(30),
         end_date       char(30))

insert          into tpcc_timer values (0,0)
go

---------------------
-- Store starting time
---------------------
update          tpcc_timer
set             start_date      = (select convert(char(30), getdate(), 21))
go

----------------------------
-- create main database files
----------------------------
CREATE DATABASE tpcc
ON PRIMARY
(               NAME                    = MSSQL_tpcc_root,
                FILENAME        = 'C:\MSSQL_tpcc_root.mdf',
                SIZE                    = 8MB,
                FILEGROWTH      = 0),
FILEGROUP       MSSQL_misc_fg
 (              NAME    = MSSQL_misc1, FILENAME= "C:\MSSQL\TPCC_14000W\misc1\", SIZE     = 37760MB, FILEGROWTH = 0),
 (              NAME    = MSSQL_misc2, FILENAME= "C:\MSSQL\TPCC_14000W\misc2\", SIZE     = 37760MB, FILEGROWTH = 0),
 (              NAME    = MSSQL_misc3, FILENAME= "C:\MSSQL\TPCC_14000W\misc3\", SIZE     = 37760MB, FILEGROWTH = 0),
 (              NAME    = MSSQL_misc4, FILENAME= "C:\MSSQL\TPCC_14000W\misc4\", SIZE     = 37760MB, FILEGROWTH = 0),
 (              NAME    = MSSQL_misc5, FILENAME= "C:\MSSQL\TPCC_14000W\misc5\", SIZE     = 37760MB, FILEGROWTH = 0),
 (              NAME    = MSSQL_misc6, FILENAME= "C:\MSSQL\TPCC_14000W\misc6\", SIZE     = 37760MB, FILEGROWTH = 0),
 (              NAME    = MSSQL_misc7, FILENAME= "C:\MSSQL\TPCC_14000W\misc7\", SIZE     = 37760MB, FILEGROWTH = 0),
 (              NAME    = MSSQL_misc8, FILENAME= "C:\MSSQL\TPCC_14000W\misc8\", SIZE     = 37760MB, FILEGROWTH = 0),
 (              NAME    = MSSQL_misc9, FILENAME= "C:\MSSQL\TPCC_14000W\misc9\", SIZE     = 37760MB, FILEGROWTH = 0),
 (              NAME    = MSSQL_misc10, FILENAME= "C:\MSSQL\TPCC_14000W\misc10\", SIZE   = 37760MB, FILEGROWTH = 0),
 (              NAME    = MSSQL_misc11, FILENAME= "C:\MSSQL\TPCC_14000W\misc11\", SIZE   = 37760MB, FILEGROWTH = 0),
 (              NAME    = MSSQL_misc12, FILENAME= "C:\MSSQL\TPCC_14000W\misc12\", SIZE   = 37760MB, FILEGROWTH = 0),
FILEGROUP       MSSQL_cs_fg
  (             NAME    = MSSQL_cs1, FILENAME= "C:\MSSQL\TPCC_14000W\cs1\", SIZE         = 46080MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cs2, FILENAME= "C:\MSSQL\TPCC_14000W\cs2\", SIZE         = 46080MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cs3, FILENAME= "C:\MSSQL\TPCC_14000W\cs3\", SIZE         = 46080MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cs4, FILENAME= "C:\MSSQL\TPCC_14000W\cs4\", SIZE         = 46080MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cs5, FILENAME= "C:\MSSQL\TPCC_14000W\cs5\", SIZE         = 46080MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cs6, FILENAME= "C:\MSSQL\TPCC_14000W\cs6\", SIZE         = 46080MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cs7, FILENAME= "C:\MSSQL\TPCC_14000W\cs7\", SIZE         = 46080MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cs8, FILENAME= "C:\MSSQL\TPCC_14000W\cs8\", SIZE         = 46080MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cs9, FILENAME= "C:\MSSQL\TPCC_14000W\cs9\", SIZE         = 46080MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cs10, FILENAME= "C:\MSSQL\TPCC_14000W\cs10\", SIZE       = 46080MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cs11, FILENAME= "C:\MSSQL\TPCC_14000W\cs11\", SIZE       = 46080MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cs12, FILENAME= "C:\MSSQL\TPCC_14000W\cs12\", SIZE       = 46080MB, FILEGROWTH = 0),
FILEGROUP       MSSQL_cust_fg
  (             NAME    = MSSQL_cust1, FILENAME= "C:\MSSQL\TPCC_14000W\cust1\", SIZE     = 21120MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cust2, FILENAME= "C:\MSSQL\TPCC_14000W\cust2\", SIZE     = 21120MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cust3, FILENAME= "C:\MSSQL\TPCC_14000W\cust3\", SIZE     = 21120MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cust4, FILENAME= "C:\MSSQL\TPCC_14000W\cust4\", SIZE     = 21120MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cust5, FILENAME= "C:\MSSQL\TPCC_14000W\cust5\", SIZE     = 21120MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cust6, FILENAME= "C:\MSSQL\TPCC_14000W\cust6\", SIZE     = 21120MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cust7, FILENAME= "C:\MSSQL\TPCC_14000W\cust7\", SIZE     = 21120MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cust8, FILENAME= "C:\MSSQL\TPCC_14000W\cust8\", SIZE     = 21120MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cust9, FILENAME= "C:\MSSQL\TPCC_14000W\cust9\", SIZE     = 21120MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cust10, FILENAME= "C:\MSSQL\TPCC_14000W\cust10\", SIZE   = 21120MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cust11, FILENAME= "C:\MSSQL\TPCC_14000W\cust11\", SIZE   = 21120MB, FILEGROWTH = 0)
, (             NAME    = MSSQL_cust12, FILENAME= "C:\MSSQL\TPCC_14000W\cust12\", SIZE   = 21120MB, FILEGROWTH = 0)
LOG ON
(               NAME                    = MSSQL_tpcc_log,
                FILENAME        = "L:",
                SIZE                    = 230000MB,
                FILEGROWTH      = 0)
COLLATE Latin1_General_BIN
go

-------------------
-- Store ending time
-------------------
update          tpcc_timer
set             end_date        = (select convert(char(30), getdate(), 21))
go

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

------------------------
-- remove temporary table
------------------------
if exists ( select name from sysobjects where name = 'tpcc_timer' )
        drop table tpcc_timer
go
```

Kit MSTPCC451\SETUP\scripts\14000.war\database\removedb.sql

```
-----------------------------------------------------------------
--                                              --
--  File:  REMOVEDB.SQL                                 --
--      Microsoft TPC-C Benchmark Kit Ver. 4.51                 --
--      Copyright Microsoft, 2003                       --
--                                              --
-----------------------------------------------------------------

use master
go

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

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

Kit MSTPCC451\SETUP\scripts\14000.war\database\restore.sql

```
-----------------------------------------------------------------
--                                              --
--  File:  RESTORE.SQL                          --
--       Microsoft TPC-C Benchmark Kit Ver. 4.51        --
--       Copyright Microsoft, 2003              --
--                                              --
-----------------------------------------------------------------

declare         @startdate      datetime,
                @enddate        datetime

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

load database tpcc from tpccback with stats = 1

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

go
```

Kit MSTPCC451\SETUP\scripts\14000.war\ddl\idxordnc.sql

```
-----------------------------------------------------------------
--                                              --
--  File:  IDXORDNC.SQL                         --
--       Microsoft TPC-C Benchmark Kit Ver. 4.51        --
--       Copyright Microsoft, 2003              --
--                                              --
--       Creates non-clustered index on orders table    --
-----------------------------------------------------------------

-- use tpcc
-- go

-- declare      @startdate      datetime,
--              @enddate        datetime

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

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

-- create index orders_nc1 on orders(o_w_id, o_d_id, o_c_id, o_id)
--              with pad_index, fillfactor=90,
--              SORT_IN_TEMPDB
--              on MSSQL_misc_fg

-- select       @enddate = getdate()
-- select       'End date: ',
--                      convert(varchar(30),@enddate, 21)
-- select       'Elapsed time (in seconds): ',
--                      datediff(second, @startdate, @enddate)
-- go
```

Loader Source Code

Kit MSTPCC451\SETUP\loader\src\getargs.c

```
//              File:                   GETARGS.C
//                                                      Microsoft TPC-C Kit Ver. 4.51
//                                                      Copyright Microsoft, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
//              Purpose:        Source file for command line processing


// Includes
#include "tpcc.h"

//===========================================================================
//
// Function name: GetArgsLoader
//
//===========================================================================

void GetArgsLoader(int argc, char **argv, TPCCLDR_ARGS *pargs)
{
   int                          i;
   char         *ptr;

#ifdef DEBUG
   printf("[%ld]DBG: Entering GetArgsLoader()\n", (int) GetCurrentThreadId());
#endif

   /* init args struct with some useful values */
   pargs->server                                = SERVER;
   pargs->user                                  = USER;
   pargs->password                              = PASSWORD;
   pargs->database                              = DATABASE;
   pargs->batch                                 = BATCH;
   pargs->num_warehouses                        = UNDEF;
                pargs->tables_all                                       = TRUE;
                pargs->table_item                                       = FALSE;
                pargs->table_warehouse                                  = FALSE;
                pargs->table_customer                                   = FALSE;
                pargs->table_orders                                             = FALSE;
                pargs->loader_res_file                                  = LOADER_RES_FILE;
                pargs->log_path                                                 = LOADER_LOG_PATH;
                pargs->pack_size                                        = DEFLDPACKSIZE;
                pargs->starting_warehouse       = DEF_STARTING_WAREHOUSE;
                pargs->build_index                                      = BUILD_INDEX;
                pargs->index_order                                      = INDEX_ORDER;
                pargs->index_script_path        = INDEX_SCRIPT_PATH;
                pargs->scale_down                                       = SCALE_DOWN;

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

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

                ptr = argv[i];

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

                case 'D':
                                                        pargs->database = ptr+2;
                                                        break;

                case 'P':
                                                        pargs->password = ptr+2;
                                                        break;

                case 'S':
                                                        pargs->server = ptr+2;
                                                        break;
```

```
                                        case 'U':
                                                pargs->user = ptr+2;
                                                break;

                                        case 'b':
                                                pargs->batch = atol(ptr+2);
                                                break;

                                        case 'W':
                                                pargs->num_warehouses = atol(ptr+2);
                                                break;

                                        case 's':
                                                pargs->starting_warehouse = atol(ptr+2);
                                                break;

                                        case 't':
                                                {
                                                        pargs->tables_all = FALSE;
                                                        if (strcmp(ptr+2,"item") == 0)
                                                                pargs->table_item = TRUE;
                                                        else if (strcmp(ptr+2,"warehouse") == 0)
                                                                pargs->table_warehouse = TRUE;
                                                        else if (strcmp(ptr+2,"customer") == 0)
                                                                pargs->table_customer = TRUE;
                                                        else if (strcmp(ptr+2,"orders") == 0)
                                                                pargs->table_orders = TRUE;
                                                        else
                                                        {
                                                        printf("\nUnrecognized command");
                                                        GetArgsLoaderUsage();
                                                        exit(1);
                                                        }

                                                        break;
                                                }

                                        case 'f':
                                                pargs->loader_res_file = ptr+2;
                                                break;

                                        case 'L':
                                                pargs->log_path = ptr+2;
                                                break;

                                        case 'p':
                                                pargs->pack_size = atol(ptr+2);
                                                break;

                                        case 'i':
                                                pargs->build_index = atol(ptr+2);
                                                break;

                                        case 'o':
                                                pargs->index_order = atol(ptr+2);
                                                break;

                                        case 'c':
                                                pargs->scale_down = atol(ptr+2);
                                                break;

                                        case 'd':
                                                pargs->index_script_path = ptr+2;
                                                break;

                                        default:
                                        GetArgsLoaderUsage();
                                        exit(-1);
                                        break;
                                }
        }

        /* check for required args */
        if (pargs->num_warehouses == UNDEF )
        {
                                printf("Number of Warehouses is required\n");
                                exit(-2);
        }

        return;
}

//==============================================================================
//
// Function name: GetArgsLoaderUsage
//
//==============================================================================

void GetArgsLoaderUsage()
{

#ifdef DEBUG
        printf("[%ld]DBG: Entering GetArgsLoaderUsage()\n", (int) GetCurrentThreadId());
#endif


        printf("TPCCLDR:\n\n");
                        printf("Parameter                                       Default\n");
                        printf("-------------------------------------------------------------------\n");
        printf("-W Number of Warehouses to Load                 Required \n");
        printf("-S Server                                       %s\n", SERVER);
        printf("-U Username                                     %s\n", USER);
        printf("-P Password                                     %s\n", PASSWORD);
        printf("-D Database                                     %s\n", DATABASE);
                        printf("-b Batch Size                                   %ld\n", (long) BATCH);
                        printf("-p TDS packet size                              %ld\n", (long) DEFLDPACKSIZE);
                        printf("-L Loader BCP Log Path                          %s\n", LOADER_LOG_PATH);
                        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);
}
```

Kit MSTPCC451\SETUP\loader\src\random.c

```
//                      File:                   RANDOM.C
//                                                      Microsoft TPC-C Kit Ver. 4.51
//                                                      Copyright Microsoft, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
//                      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;
}
```

Kit MSTPCC451\SETUP\loader\src\strings.c

```
//                      File:                                   STRINGS.C
//                                                                      Microsoft TPC-C Kit Ver. 4.51
//                                                                      Copyright Microsoft, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
//                      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;
    }
                        str[len] = 0;

    return len;
}

//====================================================================
//
// Function name: MakeOriginalAlphaString
//
//====================================================================

int MakeOriginalAlphaString(int  x,
                                                                                                        int  y,
                                                                                                                int  z,
                                                                                                        char *str,
                                                                                                int  percent)
{
    int                                 len;
    int                                 val;
    int                                 start;

#ifdef DEBUG
    printf("[%ld]DBG: Entering MakeOriginalAlphaString()\n", (int) GetCurrentThreadId());
#endif

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

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

    // Make Alpha String
    len = MakeAlphaString(x,y, z, str);

    val = RandomNumber(1,100);
    if (val <= percent)
    {
                                        start = RandomNumber(0, len - 8);
                                        strncpy(str + start, "ORIGINAL", 8);
    }


#ifdef DEBUG
    printf("[%ld]DBG: MakeOriginalAlphaString: : %s\n",
                                                                (int) GetCurrentThreadId(), str);
#endif

    return strlen(str);
}


//====================================================================
//
// Function name: MakeNumberString
//
//====================================================================
int MakeNumberString(int  x, int  y, int  z, char *str)
{
                        char tmp[16];

                        //MakeNumberString is always called MakeZipNumberString(16, 16, 16, string)

                        memset(str, '0', 16);
                        itoa(RandomNumber(0, 99999999), tmp, 10);
                        memcpy(str, tmp, strlen(tmp));

                        itoa(RandomNumber(0, 99999999), tmp, 10);
                        memcpy(str+8, tmp, strlen(tmp));

                        str[16] = 0;

    return 16;
}


//====================================================================
//
// Function name: MakeZipNumberString
//
//====================================================================
int MakeZipNumberString(int  x, int  y, int  z, char *str)
{
                        char tmp[16];

                        //MakeZipNumberString is always called MakeZipNumberString(9, 9, 9, string)

                        strcpy(str, "000011111");

                        itoa(RandomNumber(0, 9999), tmp, 10);
                        memcpy(str, tmp, strlen(tmp));

    return 9;
}


//====================================================================
//
// Function name: InitString
//
//====================================================================
void InitString(char *str, int len)
{
#ifdef DEBUG
    printf("[%ld]DBG: Entering InitString()\n", (int) GetCurrentThreadId());
#endif

                        memset(str, ' ', len);
    str[len] = 0;
}


//====================================================================
// Function name: InitAddress
//
// Description:
//
//====================================================================

void InitAddress(char *street_1, char *street_2, char *city, char *state, char *zip)
{
                        memset(street_1, ' ', ADDRESS_LEN+1);
                        memset(street_2, ' ', ADDRESS_LEN+1);
                        memset(city, ' ', ADDRESS_LEN+1);

    street_1[ADDRESS_LEN+1] = 0;
    street_2[ADDRESS_LEN+1] = 0;
    city[ADDRESS_LEN+1]     = 0;

                        memset(state, ' ', STATE_LEN+1);
    state[STATE_LEN+1] = 0;
```

```
                memset(zip, ' ', ZIP_LEN+1);
    zip[ZIP_LEN+1] = 0;
}


//====================================================================
//
// Function name: PaddString
//
//====================================================================

void PaddString(int max, char *name)
{
                int                             len;

                len = strlen(name);
                if ( len < max )
                                memset(name+len, ' ', max - len);
                name[max] = 0;

                return;
}
```

Kit MSTPCC451\SETUP\loader\src\time.c

```
//              File:                   TIME.C
//                                                      Microsoft TPC-C Kit Ver. 4.51
//                                                      Copyright Microsoft, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
//              Purpose:        Source file for time functions


// Includes
#include "tpcc.h"

// Globals
static long start_sec;


//====================================================================
//
// Function name: TimeNow
//
//====================================================================

long TimeNow()
{
  long                          time_now;
  struct        _timeb el_time;

#ifdef DEBUG
    printf("[%ld]DBG: Entering TimeNow()\n", (int) GetCurrentThreadId());
#endif

  _ftime(&el_time);

  time_now = ((el_time.time - start_sec) * 1000) + el_time.millitm;

  return time_now;
}
```

Kit MSTPCC451\SETUP\loader\src\tpcc.h

```
//              File:                   TPCC.H
//                                                      Microsoft TPC-C Kit Ver. 4.51
//                                                      Copyright Microsoft, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
//              Purpose:        Header file for TPC-C database loader


// Build number of TPC Benchmark Kit
#define TPCKIT_VER              "4.51"

// 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>
#include <math.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.450\\SETUP\\LOGS\\load.out"
#define LOADER_LOG_PATH                         "C:\\MSTPCC.450\\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();
```

Kit MSTPCC451\SETUP\loader\src\tpccldr.c

```
//================================================================
//              File:                   TPCCLDR.C
//                                                      Microsoft TPC-C Kit Ver. 4.51
//                                                      Copyright Microsoft, 1996, 1997, 1998, 1999,
//                                                      2000, 2001, 2002, 2003
//              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
#define MAX_SQL_ERRORS                                  10

// Functions declarations
void HandleErrorDBC (SQLHDBC  hdbc1);
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 CheckForCommit_Big();
void OpenConnections();
void BuildIndex();
void FormatDate ();

// Shared memory structures
typedef struct
{
                double                                                  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;
                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;

int                                 total_db_errors;

ORDERS_STRUCT                       orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT                     customer_buf[CUSTOMERS_PER_DISTRICT];
long                                        orders_rows_loaded;
double                                      new_order_rows_loaded;
double                                      order_line_rows_loaded;
long                                        history_rows_loaded;
long                                        customer_rows_loaded;
double                                      stock_rows_loaded;
long                                        district_rows_loaded;
long                                        item_rows_loaded;
long                                        warehouse_rows_loaded;
long                                        main_time_start;
long                                        main_time_end;
long                                        max_items;
long                                        customers_per_district;
long                                        orders_per_district;
long                                        first_new_order;
long                                        last_new_order;

TPCCLDR_ARGS    *aptr, args;

//==============================================================================
//
// Function name: main
//
//==============================================================================
int main(int  argc, char **argv)
{

                DWORD           dwThreadID[MAX_MAIN_THREADS];
                HANDLE          hThread[MAX_MAIN_THREADS];
                FILE            *fLoader;
                char            buffer[255];
                int                                                             i;

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

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

   // process command line arguments
   aptr = &args;
   GetArgsLoader(argc, argv, aptr);

                printf("Build interface is ODBC.\n");

                if (aptr->build_index == 0)
                        printf("Data load only - no index creation.\n");
                else
                        printf("Data load and index creation.\n");

                if (aptr->index_order == 0)
                        printf("Clustered indexes will be created after bulk load.\n");
                else
                        printf("Clustered indexes will be created before bulk load.\n");

                // set database scale values
                if (aptr->scale_down == 1)
                {
                        printf("*** Scaled Down Database ***\n");
                        max_items = MAXITEMS_SCALE_DOWN;
                        customers_per_district = CUSTOMERS_SCALE_DOWN;
                        orders_per_district = ORDERS_SCALE_DOWN;
                        first_new_order = 0;
                        last_new_order  = 30;
                }
                else
                {
                        max_items = MAXITEMS;
                        customers_per_district = CUSTOMERS_PER_DISTRICT;
                        orders_per_district = ORDERS_PER_DISTRICT;
                        first_new_order = 2100;
                        last_new_order  = 3000;
                }

                // open connections to SQL Server
                OpenConnections();

                // open file for loader results
                fLoader = fopen(aptr->loader_res_file, "w");


   if (fLoader == NULL)
                {
                        printf("Error, loader result file open failed.");
                        exit(-1);
                }

   // start loading data
   sprintf(buffer,"TPC-C load started for %ld warehouses.\n",aptr->num_warehouses);

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

                main_time_start = (TimeNow() / MILLI);

                // start parallel load threads
                if (aptr->tables_all || aptr->table_item)
                {
                        fprintf(fLoader, "\nStarting loader threads for: item\n");

                        hThread[0] = CreateThread(NULL,
                                                                                                0,
                                                                                                (LPTHREAD_START_ROUTINE) LoadItem,
                                                                                                NULL,
                                                                                                0,
                                                                                                &dwThreadID[0]);
```

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

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

                        hThread[1] = CreateThread(NULL,
                                                                                                                0,
                                                                                                                (LPTHREAD_START_ROUTINE) LoadWarehouse,
                                                                                                                NULL,
                                                                                                                0,
                                                                                                                &dwThreadID[1]);

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

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

                        hThread[2] = CreateThread(NULL,
                                                                                                                0,
                                                                                                                (LPTHREAD_START_ROUTINE) LoadCustomer,
                                                                                                                NULL,
                                                                                                                0,
                                                                                                                &dwThreadID[2]);

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

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

                        hThread[3] = CreateThread(NULL,
                                                                                                                0,
                                                                                                                (LPTHREAD_START_ROUTINE) LoadOrders,
                                                                                                                NULL,
                                                                                                                0,
                                                                                                                &dwThreadID[3]);

                        if (hThread[3] == NULL)
                        {
                                printf("Error, failed in creating creating main thread = 3.\n");
                                exit(-1);
                        }
                }

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

                main_time_end = (TimeNow() / MILLI);

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

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

                fclose(fLoader);

                SQLFreeEnv(henv);

    exit(0);

                return 0;
}

//====================================================================
//
// Function name: LoadItem
//
//====================================================================
void LoadItem()
{
    long                        i_id;
                long                        i_im_id;
    char                        i_name[I_NAME_LEN+1];
    double                      i_price;
    char                        i_data[I_DATA_LEN+1];
                char                        name[20];
                long                        time_start;
                RETCODE                     rc;
                DBINT                       rcint;
                char                        bcphint[128];
                char                        err_log_path[256];

                // Seed with unique number
                seed(1);

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

                //if build index before load
                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                        BuildIndex("idxitmcl");

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

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

                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");

		// if build index before load...
		if ((aptr->build_index == 1) && (aptr->index_order == 1))
			BuildIndex("idxwarcl");

		InitString(w_name, W_NAME_LEN+1);
		InitAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

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

		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 = (long)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");

                // build index before load
                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                        BuildIndex("idxdiscl");

                InitString(d_name, D_NAME_LEN+1);
                InitAddress(d_street_1, d_street_2, d_city, d_state, d_zip);
                sprintf(name, "%s..%s", aptr->database, "district");

                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);

			// if build index before load...
			if ((aptr->build_index == 1) && (aptr->index_order == 1))
				BuildIndex("idxstkcl");

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

			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);
			rc = 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 = (long)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_Big(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];

                // Seed with unique number
                seed(5);

                printf("Loading customer and history tables...\n");

                // if build index before load...
                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                {
                                BuildIndex("idxcuscl");
                                 // check the number of processors on this system
                                 // if 8 or more processors, then build index on History.
                                 // if less than 8 processors, do not build the index
                                num_procs = atoi(getenv( "NUMBER_OF_PROCESSORS" ));
                                if ( num_procs >= 8 )
                                                BuildIndex("idxhiscl");
                }

                // Initialize bulk copy
                sprintf(name, "%s..%s", aptr->database, "customer");

                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 = (long)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\" > %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()
{
                long    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;

                                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, long 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;
                                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;
                                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)
{
                long                                    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;
                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);
    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;
                                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)
{
                long                                    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");

                // if build index before load...
                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                {
                                        BuildIndex("idxordcl");
                                        BuildIndex("idxnodcl");
                                        BuildIndex("idxodlcl");
                }

                // initialize bulk copy
                sprintf(name, "%s..%s", aptr->database, "orders");

                rc = bcp_init(o_hdbc1, name, NULL, "logs\\orders.err", DB_IN);
                strcpy(err_log_path_ord,aptr->log_path);
                strcat(err_log_path_ord,"orders.err");
                rc = bcp_init(o_hdbc1, name, NULL, err_log_path_ord, DB_IN);
                if (rc != SUCCEED)
                                        HandleErrorDBC(o_hdbc1);

                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                {
                                        sprintf(bcphint, "tablock, order (o_w_id, o_d_id, o_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
                                        rc = bcp_control(o_hdbc1, BCPHINTS, (void*) bcphint);
                                        if (rc != SUCCEED)
                                                HandleErrorDBC(o_hdbc1);
                }

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

                rc = bcp_init(o_hdbc2, name, NULL, "logs\\neword.err", DB_IN);
                strcpy(err_log_path_nord,aptr->log_path);
                strcat(err_log_path_nord,"neword.err");
                rc = bcp_init(o_hdbc2, name, NULL, err_log_path_nord, DB_IN);
                if (rc != SUCCEED)
                                        HandleErrorDBC(o_hdbc2);

                if ((aptr->build_index == 1) && (aptr->index_order == 1))
                {
                                        sprintf(bcphint, "tablock, order (no_w_id, no_d_id, no_o_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 9000));
                                        rc = bcp_control(o_hdbc2, BCPHINTS, (void*) bcphint);
                                        if (rc != SUCCEED)
                                                HandleErrorDBC(o_hdbc2);
                }

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

                rc = bcp_init(o_hdbc3, name, NULL, "logs\\ordline.err", DB_IN);
                strcpy(err_log_path_ordl,aptr->log_path);
                strcat(err_log_path_ordl,"ordline.err");
                rc = bcp_init(o_hdbc3, name, NULL, err_log_path_ordl, DB_IN);
                if (rc != SUCCEED)
                                        HandleErrorDBC(o_hdbc3);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
                {
                                        sprintf(bcphint, "tablock, order (ol_w_id, ol_d_id, ol_o_id, ol_number), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 300000));
                                        rc = bcp_control(o_hdbc3, BCPHINTS, (void*) bcphint);
                                        if (rc != SUCCEED)
                                                HandleErrorDBC(o_hdbc3);
                }

                orders_rows_loaded      = 0;
                new_order_rows_loaded   = 0;
                order_line_rows_loaded  = 0;

                OrdersBufInit();

                orders_time_start.time_start = (TimeNow() / MILLI);
                new_order_time_start.time_start = (TimeNow() / MILLI);
                order_line_time_start.time_start = (TimeNow() / MILLI);

                for (w_id = (long)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(short d_id, long w_id)
{

                        int     cust[ORDERS_PER_DISTRICT+1];
                        long    o_id;
                        long                    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);
    }

                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)
{
                long                                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_Big(o_hdbc2, o_hstmt2, new_order_rows_loaded, "new_order", &new_order_time_start->time_start);
                }

                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)
{
                long                                i;
                long                                j;
    long        o_id;
                short               o_d_id;
                long                                o_w_id;
                double                              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, SQLFLT8, 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_Big(o_hdbc3, o_hstmt3, order_line_rows_loaded, "order_line", &order_line_time_start->time_start);
                                        }
                        }

                        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,
                                                                                                long rows_loaded,
                                                                                                char *table_name,
                                                                        long *time_start)
{

   long                 time_end, time_diff;

   if ( !(rows_loaded % aptr->batch) )
   {
                                        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   : CheckForCommit_Big
//
//================================================================
void CheckForCommit_Big(HDBC hdbc,                                                              HSTMT hstmt,
                                                                                                double rows_loaded,
                                                                                                char *table_name,
                                                                        long *time_start)
{

   long                 time_end, time_diff;

   if ( !(fmod(rows_loaded,aptr->batch) ) )
   {
                                        time_end = (TimeNow() / MILLI);
                                        time_diff = time_end - *time_start;

                                        printf("-> Loaded %ld rows into %s in %ld sec - Total = %.0f (%.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) &&
                         (rc != SQL_SUCCESS_WITH_INFO) )
        {
                        HandleErrorDBC(i_hdbc1);
                        printf("TPC-C Loader aborted!\n");
                        exit(9);
        }

        // 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) &&
                         (rc != SQL_SUCCESS_WITH_INFO) )
        {
                        HandleErrorDBC(w_hdbc1);
                        printf("TPC-C Loader aborted!\n");
                        exit(9);
        }

        // 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) &&
                         (rc != SQL_SUCCESS_WITH_INFO) )
        {
                        HandleErrorDBC(c_hdbc1);
                        printf("TPC-C Loader aborted!\n");
                        exit(9);
        }

        // 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) &&
                         (rc != SQL_SUCCESS_WITH_INFO) )
        {
                        HandleErrorDBC(c_hdbc2);
                        printf("TPC-C Loader aborted!\n");
                        exit(9);
```

```
		}

		// 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) &&
				 (rc != SQL_SUCCESS_WITH_INFO) )
		{
				HandleErrorDBC(o_hdbc1);
				printf("TPC-C Loader aborted!\n");
				exit(9);
		}

		// 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) &&
				 (rc != SQL_SUCCESS_WITH_INFO) )
		{
				HandleErrorDBC(o_hdbc2);
				printf("TPC-C Loader aborted!\n");
				exit(9);
		}

		// 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) &&
				 (rc != SQL_SUCCESS_WITH_INFO) )
		{
				HandleErrorDBC(o_hdbc3);
				printf("TPC-C Loader aborted!\n");
				exit(9);
		}
}

//==============================================================================
//
// 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);
}

//==============================================================================
//
// Function name: HandleErrorDBC
//
//==============================================================================
void HandleErrorDBC (SQLHDBC  hdbc1)
{

		SQLCHAR				SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
		SQLLEN				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==>SQLState: %s\n" , datebuf, timebuf, szLastError, SqlState);

				strcpy(err_log_path,aptr->log_path);
				strcat(err_log_path,"tpccldr.err");
				fp1 = fopen(err_log_path,"a+");
				if (fp1 == NULL)
						printf("ERROR:  Unable to open errorlog file.\n");
				else
				{
						fprintf(fp1, "[%s : %s] %s\nSQLState: %s\n" , datebuf, timebuf, szLastError, SqlState);
						fclose(fp1);
				}

				i++;
		}
}
```

```
//==================================================================
//
// Function   : HandleErrorSTMT
//
//==================================================================
void HandleErrorSTMT (HSTMT  hstmt1)
{
                SQLCHAR                         SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
                SQLLEN                          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 )
                {
                                if (total_db_errors >= MAX_SQL_ERRORS)
                                {
                                                printf(">>>>> Maximum SQL errors of %d exceeded.  Terminating TPCCLDR.<<<<<\n",total_db_errors);
                                                exit(9);
                                }
                                total_db_errors++;

                                sprintf( szLastError , "%s" ,  Msg );

                                _strtime(timebuf);
                                _strdate(datebuf);

                                printf( "[%s : %s] %s\nSQLState: %s\n" , datebuf, timebuf, szLastError, SqlState);

                                strcpy(err_log_path,aptr->log_path);
                                strcat(err_log_path,"tpccldr.err");
                                fp1 = fopen(err_log_path,"a+");
                                if (fp1 == NULL)
                                                printf("ERROR:  Unable to open errorlog file.\n");
                                else
                                {
                                                fprintf(fp1, "[%s : %s] %s\nSQLState: %s\n" , datebuf, timebuf, szLastError, SqlState);
                                                fclose(fp1);
                                }

                                i++;
                }
}

//==================================================================
//
// Function   : FormatDate
//
//==================================================================
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;
}
```

Appendix C: Tunable parameters used in the TPC-C tests.

RTE input Parameters

The following parameters were used with Microsoft BenchCraft RTE:

Mesures\NS5080\CS140000 mseed1.v3.txt

```
Profile:        CS140000
File Path:      C:\logs\CS140000.xml
Version:        0

                Number of Engines: 32

                        Name: DRIVER1
                        Description:
                        Directory: c:\logs\driver1.log
                        Machine: rte01
                        Parameter Set: CS
                        Index: 500000
                        Seed: 42921
                        Configured Users: 4580
                        Pipe Name: DRIVER1516771137
                        Connect Rate: 1000
                        Start Rate: 150
                        Max. Concurrency: -1
                        Concurrency Rate: 10
                        CLIENT_NURAND: 233
                        CPU: 0
                        Additional Options:

                        Name: DRIVER2
                        Description:
                        Directory: c:\logs\driver2.log
                        Machine: rte01
                        Parameter Set: CS
                        Index: 1000000
                        Seed: 42921
                        Configured Users: 4580
                        Pipe Name: DRIVER2516771137
                        Connect Rate: 1000
                        Start Rate: 150
                        Max. Concurrency: -1
                        Concurrency Rate: 10
                        CLIENT_NURAND: 233
                        CPU: 0
                        Additional Options:

                        Name: DRIVER3
                        Description:
                        Directory: c:\logs\driver3.log
                        Machine: rte01
                        Parameter Set: CS
                        Index: 1500000
                        Seed: 42921
                        Configured Users: 4580
                        Pipe Name: DRIVER3516771137
                        Connect Rate: 1000
                        Start Rate: 150
                        Max. Concurrency: -1
                        Concurrency Rate: 10
                        CLIENT_NURAND: 233
                        CPU: 0
                        Additional Options:

                        Name: DRIVER4
                        Description:
                        Directory: c:\logs\driver4.log
                        Machine: rte02
                        Parameter Set: CS
                        Index: 2000000
                        Seed: 42921
                        Configured Users: 4580
                        Pipe Name: DRIVER4516771137
                        Connect Rate: 1000
                        Start Rate: 150
                        Max. Concurrency: -1
                        Concurrency Rate: 10
                        CLIENT_NURAND: 233
                        CPU: 0
                        Additional Options:

                        Name: DRIVER5
                        Description:
                        Directory: c:\logs\driver5.log
                        Machine: rte02
                        Parameter Set: CS
                        Index: 2500000
                        Seed: 42921
                        Configured Users: 4100
                        Pipe Name: DRIVER5516771137
                        Connect Rate: 1000
                        Start Rate: 150
                        Max. Concurrency: -1
                        Concurrency Rate: 10
                        CLIENT_NURAND: 233
                        CPU: 0
                        Additional Options:

                        Name: DRIVER6
                        Description:
                        Directory: c:\logs\driver6.log
                        Machine: rte02
                        Parameter Set: CS
                        Index: 3000000
                        Seed: 42921
                        Configured Users: 4100
                        Pipe Name: DRIVER6516771137
                        Connect Rate: 1000
                        Start Rate: 150
                        Max. Concurrency: -1
                        Concurrency Rate: 10
                        CLIENT_NURAND: 233
                        CPU: 0
                        Additional Options:

                        Name: DRIVER7
                        Description:
                        Directory: c:\logs\driver7.log
                        Machine: rte03
                        Parameter Set: CS
                        Index: 3500000
                        Seed: 42921
                        Configured Users: 4100
                        Pipe Name: DRIVER7516771137
                        Connect Rate: 1000
                        Start Rate: 150
                        Max. Concurrency: -1
                        Concurrency Rate: 10
                        CLIENT_NURAND: 233
                        CPU: 0
                        Additional Options:

                        Name: DRIVER8
                        Description:
                        Directory: c:\logs\driver8.log
                        Machine: rte03
                        Parameter Set: CS
                        Index: 4000000
                        Seed: 42921
                        Configured Users: 4100
                        Pipe Name: DRIVER8516771137
                        Connect Rate: 1000
```

Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER9
Description:
Directory: c:\logs\driver9.log
Machine: rte03
Parameter Set: CS
Index: 4500000
Seed: 42921
Configured Users: 4320
Pipe Name: DRIVER9516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER10
Description:
Directory: c:\logs\driver10.log
Machine: rte04
Parameter Set: CS
Index: 5000000
Seed: 42921
Configured Users: 4320
Pipe Name: DRIVER10516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER11
Description:
Directory: c:\logs\driver11.log
Machine: rte04
Parameter Set: CS
Index: 5500000
Seed: 42921
Configured Users: 4320
Pipe Name: DRIVER11516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER12
Description:
Directory: c:\logs\driver12.log
Machine: rte04
Parameter Set: CS
Index: 6000000
Seed: 42921
Configured Users: 4320
Pipe Name: DRIVER12516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER13
Description:
Directory: c:\logs\driver13.log
Machine: rte05
Parameter Set: CS
Index: 6500000
Seed: 42921
Configured Users: 4500
Pipe Name: DRIVER13516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER14
Description:
Directory: c:\logs\driver14.log
Machine: rte05
Parameter Set: CS
Index: 7000000
Seed: 42921
Configured Users: 4500
Pipe Name: DRIVER14516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER15
Description:
Directory: c:\logs\driver15.log
Machine: rte05
Parameter Set: CS
Index: 7500000
Seed: 42921
Configured Users: 4500
Pipe Name: DRIVER15516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER16
Description:
Directory: c:\logs\driver16.log
Machine: rte06
Parameter Set: CS
Index: 8000000
Seed: 42921
Configured Users: 4500
Pipe Name: DRIVER16516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER17
Description:
Directory: c:\logs\driver17.log
Machine: rte06
Parameter Set: CS
Index: 8500000
Seed: 42921
Configured Users: 4580
Pipe Name: DRIVER17516771137

Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER18
Description:
Directory: c:\logs\driver18.log
Machine: rte06
Parameter Set: CS
Index: 9000000
Seed: 42921
Configured Users: 4580
Pipe Name: DRIVER18516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER19
Description:
Directory: c:\logs\driver19.log
Machine: rte07
Parameter Set: CS
Index: 9500000
Seed: 42921
Configured Users: 4580
Pipe Name: DRIVER19516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER20
Description:
Directory: c:\logs\driver20.log
Machine: rte07
Parameter Set: CS
Index: 10000000
Seed: 42921
Configured Users: 4580
Pipe Name: DRIVER20516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER21
Description:
Directory: c:\logs\driver21.log
Machine: rte07
Parameter Set: CS
Index: 10500000
Seed: 42921
Configured Users: 4100
Pipe Name: DRIVER21516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER22
Description:
Directory: c:\logs\driver22.log
Machine: rte08
Parameter Set: CS
Index: 11000000
Seed: 42921
Configured Users: 4100
Pipe Name: DRIVER22516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER23
Description:
Directory: c:\logs\driver23.log
Machine: rte08
Parameter Set: CS
Index: 11500000
Seed: 42921
Configured Users: 4100
Pipe Name: DRIVER23516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER24
Description:
Directory: c:\logs\driver24.log
Machine: rte08
Parameter Set: CS
Index: 12000000
Seed: 42921
Configured Users: 4100
Pipe Name: DRIVER24516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER25
Description:
Directory: c:\logs\driver25.log
Machine: rte09
Parameter Set: CS
Index: 12500000
Seed: 42921
Configured Users: 4320
Pipe Name: DRIVER25516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER26
Description:
Directory: c:\logs\driver26.log
Machine: rte09
Parameter Set: CS
Index: 13000000
Seed: 42921
Configured Users: 4320

Pipe Name: DRIVER26516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER27
Description:
Directory: c:\logs\driver27.log
Machine: rte09
Parameter Set: CS
Index: 13500000
Seed: 42921
Configured Users: 4320
Pipe Name: DRIVER27516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER28
Description:
Directory: c:\logs\driver28.log
Machine: rte10
Parameter Set: CS
Index: 14000000
Seed: 42921
Configured Users: 4320
Pipe Name: DRIVER28516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER29
Description:
Directory: c:\logs\driver29.log
Machine: rte10
Parameter Set: CS
Index: 14500000
Seed: 42921
Configured Users: 4500
Pipe Name: DRIVER29516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER30
Description:
Directory: c:\logs\driver30.log
Machine: rte10
Parameter Set: CS
Index: 15000000
Seed: 42921
Configured Users: 4500
Pipe Name: DRIVER30516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER31
Description:
Directory: c:\logs\driver31.log
Machine: rte11
Parameter Set: CS
Index: 15500000
Seed: 42921
Configured Users: 4500
Pipe Name: DRIVER31516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: DRIVER32
Description:
Directory: c:\logs\driver32.log
Machine: rte11
Parameter Set: CS
Index: 16000000
Seed: 42921
Configured Users: 4500
Pipe Name: DRIVER32516771137
Connect Rate: 1000
Start Rate: 150
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Number of User groups: 32

Driver Engine: DRIVER1
IIS Server: client05
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1 - 458
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4580
District id: 1
Scale Down: No

Driver Engine: DRIVER2
IIS Server: client05
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 459 - 916
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4580
District id: 1
Scale Down: No

Driver Engine: DRIVER3
IIS Server: client05
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 917 - 1374
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4580

District id: 1
Scale Down: No

Driver Engine: DRIVER4
IIS Server: client05
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1375 - 1832
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4580
District id: 1
Scale Down: No

Driver Engine: DRIVER5
IIS Server: client06
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1833 - 2242
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4100
District id: 1
Scale Down: No

Driver Engine: DRIVER6
IIS Server: client06
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2243 - 2652
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4100
District id: 1
Scale Down: No

Driver Engine: DRIVER7
IIS Server: client06
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2653 - 3062
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4100
District id: 1
Scale Down: No

Driver Engine: DRIVER8
IIS Server: client06
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3063 - 3472
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4100
District id: 1
Scale Down: No

Driver Engine: DRIVER9
IIS Server: client07
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3473 - 3904
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4320
District id: 1
Scale Down: No

Driver Engine: DRIVER10
IIS Server: client07
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3905 - 4336
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4320
District id: 1
Scale Down: No

Driver Engine: DRIVER11
IIS Server: client07
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4337 - 4768
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4320
District id: 1
Scale Down: No

Driver Engine: DRIVER12
IIS Server: client07
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4769 - 5200
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4320
District id: 1
Scale Down: No

Driver Engine: DRIVER13
IIS Server: client08
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5201 - 5650
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4500
District id: 1
Scale Down: No

Driver Engine: DRIVER14
IIS Server: client08
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5651 - 6100
w_id Min Warehouse: 1
w_id Max Warehouse: 14000

Scale: Normal
User Count: 4500
District id: 1
Scale Down: No

Driver Engine: DRIVER15
IIS Server: client08
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 6101 - 6550
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4500
District id: 1
Scale Down: No

Driver Engine: DRIVER16
IIS Server: client08
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 6551 - 7000
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4500
District id: 1
Scale Down: No

Driver Engine: DRIVER17
IIS Server: client09
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 7001 - 7458
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4580
District id: 1
Scale Down: No

Driver Engine: DRIVER18
IIS Server: client09
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 7459 - 7916
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4580
District id: 1
Scale Down: No

Driver Engine: DRIVER19
IIS Server: client09
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 7917 - 8374
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4580
District id: 1
Scale Down: No

Driver Engine: DRIVER20
IIS Server: client09
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 8375 - 8832
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4580
District id: 1
Scale Down: No

Driver Engine: DRIVER21
IIS Server: client10
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 8833 - 9242
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4100
District id: 1
Scale Down: No

Driver Engine: DRIVER22
IIS Server: client10
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 9243 - 9652
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4100
District id: 1
Scale Down: No

Driver Engine: DRIVER23
IIS Server: client10
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 9653 - 10062
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4100
District id: 1
Scale Down: No

Driver Engine: DRIVER24
IIS Server: client10
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 10063 - 10472
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4100
District id: 1
Scale Down: No

Driver Engine: DRIVER25
IIS Server: client11
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 10473 - 10904

w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4320
District id: 1
Scale Down: No

Driver Engine: DRIVER26
IIS Server: client11
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 10905 - 11336
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4320
District id: 1
Scale Down: No

Driver Engine: DRIVER27
IIS Server: client11
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 11337 - 11768
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4320
District id: 1
Scale Down: No

Driver Engine: DRIVER28
IIS Server: client11
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 11769 - 12200
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4320
District id: 1
Scale Down: No

Driver Engine: DRIVER29
IIS Server: client12
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 12201 - 12650
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4500
District id: 1
Scale Down: No

Driver Engine: DRIVER30
IIS Server: client12
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 12651 - 13100
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4500
District id: 1
Scale Down: No

Driver Engine: DRIVER31
IIS Server: client12
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 13101 - 13550
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4500
District id: 1
Scale Down: No

Driver Engine: DRIVER32
IIS Server: client12
SQL Server: tpcc-fame
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 13551 - 14000
w_id Min Warehouse: 1
w_id Max Warehouse: 14000
Scale: Normal
User Count: 4500
District id: 1
Scale Down: No

Number of Parameter Sets: 3

~Default
Default Parameter Set

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay			
New Order	10.00			12.05	18.01	0.10	5.00	0.10	
Payment	10.00			12.05	3.01	0.10	5.00	0.10	
Delivery	1.00	5.05		2.01	0.10	5.00	0.10		
Stock Level	1.00			5.05	2.01	0.10	20.00	0.10	
Order Status	1.00			10.05	2.01	0.10	5.00	0.10	

CS
Temps de reflexion Client-Serveur

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay			
New Order	44.93			12.07	18.02	0.10	5.00	0.00	
Payment	43.01			12.07	3.02	0.10	5.00	0.00	
Delivery	4.02	5.07		2.02	0.10	5.00	0.00		
Stock Level	4.02			5.07	2.02	0.10	20.00	0.00	
Order Status	4.02			10.07	2.02	0.10	5.00	0.00	

Batch
Batch

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay			
New Order	44.92			0.01	0.00	0.00	5.00	0.00	
Payment	43.01			0.00	0.00	0.00	5.00	0.00	
Delivery	4.01	0.00		0.00	0.00	5.00	0.00		
Stock Level	4.04			0.00	0.00	0.00	20.00	0.00	
Order Status	4.01			0.00	0.00	0.00	5.00	0.00	

CLIENT Configuration

Services

Only the following services were activated during the measurement.

Name	Status	Startup Type	Log On As
Alerter	Started	Automatic	LocalSystem
COM+ Event System	Started	Manual	LocalSystem
Computer Browser	Started	Automatic	LocalSystem
Distributed File System	Started	Automatic	LocalSystem
Distributed Link Tracking Client	Started	Automatic	LocalSystem
Distributed Transaction Coordinator	Started	Automatic	LocalSystem
Event Log	Started	Automatic	LocalSystem
IIS Admin Service	Started	Automatic	LocalSystem
IPSEC Policy Agent	Started	Automatic	LocalSystem
License Logging Service	Started	Automatic	LocalSystem
Logical Disk Manager	Started	Automatic	LocalSystem
Messenger	Started	Automatic	LocalSystem
Network Connections	Started	Manual	LocalSystem
Plug and Play	Started	Automatic	LocalSystem
Protected Storage	Started	Automatic	LocalSystem
Remote Access Connection Manager	Started	Manual	LocalSystem
Remote Procedure Call (RPC)	Started	Automatic	LocalSystem
Remote Registry Service	Started	Automatic	LocalSystem
Removable Storage	Started	Automatic	LocalSystem
RunAs Service	Started	Automatic	LocalSystem
Security Accounts Manager	Started	Automatic	LocalSystem
Server	Started	Automatic	LocalSystem
Simple TCP/IP Services	Started	Automatic	LocalSystem
System Event Notification	Started	Automatic	LocalSystem
Task Scheduler	Started	Automatic	LocalSystem
TCP/IP NetBIOS Helper Service	Started	Automatic	LocalSystem
Telephony	Started	Manual	LocalSystem
Windows Management Instrumentation	Started	Automatic	LocalSystem
Windows Management Instrumentation Driver Extensions	Started	Manual	LocalSystem
Workstation	Started	Automatic	LocalSystem
World Wide Web Publishing Service	Started	Automatic	LocalSystem

COM+ Application Configuration

COM+ settings (properties of component TPCC.ALL Txns)
For each 8 frontends.

Pool COM+ TPCC.AllTxns: min=max=60

TPCC Application registry

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPCC]

Path REG_SZ C:\inetpub\WWWRoot
NumberOfDeliveryThreads REG_DWORD 0x16
MaxConnections REG_DWORD 0x4e20
MaxPendingDeliveries REG_DWORD 0x7d0
DB_Protocol REG_SZ DBLIB
TxnMonitor REG_SZ COM
DbServer REG_SZ Q0:tpcc-fame
DbName REG_SZ tpcc
DbUser REG_SZ sa
DbPassword REG_SZ
COM_SinglePool REG_SZ YES
SPprefix REG_SZ Q0

SQLServer Client

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client]
SharedMemoryOn REG_DWORD 0x0

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\TDS

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\ConnectTo
tpcc-fame REG_SZ DBNETLIB,via:tpcc-fame,1443,0

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\DB-Lib
AutoAnsiToOem REG_SZ ON
UseIntlSettings REG_SZ ON

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib
ProtocolOrder REG_SZ via
Encrypt REG_DWORD 0x0

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib\LastConnect
tpcc-fame REG_SZ 1929576456:via:TPCC-FAME:TPCC-FAME,1443,0

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib\Np
DefaultPipe REG_SZ sql\query

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib\Tcp
DefaultPort REG_DWORD 0x599

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib\VIA

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib\VIA
Vendor REG_SZ QLogic
DefaultServerPort REG_SZ 0:1443
DefaultClientNIC REG_SZ 0
RecognizedVendors REG_SZ Giganet, ServerNet II,QLogic

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\TDS
tpcc-fame REG_SZ 7.0
via:tpcc-fame,1443,0 REG_SZ 7.0
tpcc-via REG_SZ 7.0
via:tpcc-fame,1453,1 REG_SZ 7.0
via:tpcc-fame REG_SZ 7.0
tcp:tpcc-fame REG_SZ 7.0
via:tpcc-fame,1463,2 REG_SZ 7.0

InetInfo registry

Extraire et insérer ici les valeurs des clés de registre:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters
ListenBackLog REG_DWORD 0xfa
DispatchEntries REG_MULTI_SZ LDAPSVC
PoolThreadLimit REG_DWORD 0x7d0
ThreadTimeout REG_DWORD 0x15180

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance
Library REG_SZ infoctrs.dll
Open REG_SZ OpenINFOPerformanceData
Close REG_SZ CloseINFOPerformanceData
Collect REG_SZ CollectINFOPerformanceData
Last Counter REG_DWORD 0x842
Last Help REG_DWORD 0x843
First Counter REG_DWORD 0x802
First Help REG_DWORD 0x803
Library Validation Code REG_BINARY 1EFCBD5F29D9C3011025000000000000
WbemAdapFileTime REG_BINARY 009862045B36C301
WbemAdapFileSize REG_DWORD 0x2510
WbemAdapStatus REG_DWORD 0x0

WWW Service registry

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC
Type REG_DWORD 0x20
Start REG_DWORD 0x2
ErrorControl REG_DWORD 0x1
ImagePath REG_EXPAND_SZ C:\WINNT\system32\inetsrv\inetinfo.exe
DisplayName REG_SZ World Wide Web Publishing Service
DependOnService REG_MULTI_SZ IISADMIN
DependOnGroup REG_MULTI_SZ
ObjectName REG_SZ LocalSystem
Description REG_SZ Provides Web connectivity and administration through the Internet Information Services snap-in.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP
NOTE REG_SZ This is for backward compatibility only.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP\Parameters

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters
MajorVersion REG_DWORD 0x5
MinorVersion REG_DWORD 0x0
InstallPath REG_SZ C:\WINNT\system32\inetsrv
CertMapList REG_SZ C:\WINNT\system32\inetsrv\iiscrmap.dll
AccessDeniedMessage REG_SZ Error: Access is Denied.
Filter DLLs REG_SZ
LogFileDirectory REG_SZ C:\WINNT\system32\LogFiles
AcceptExOutstanding REG_DWORD 0x28

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots
/ REG_SZ c:\inetpub\wwwroot,,205
/Scripts REG_SZ c:\inetpub\scripts,,204
/IISHelp REG_SZ c:\winnt\help\iishelp,,201
/IISAdmin REG_SZ C:\WINNT\system32\inetsrv\iisadmin,,201
/IISSamples REG_SZ c:\inetpub\iissamples,,201
/MSADC REG_SZ c:\program files\common files\system\msadc,,205
/Printers REG_SZ C:\WINNT\web\printers,,201

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Performance
Library REG_SZ w3ctrs.dll
Open REG_SZ OpenW3PerformanceData
Close REG_SZ CloseW3PerformanceData
Collect REG_SZ CollectW3PerformanceData
Last Counter REG_DWORD 0x8e6
Last Help REG_DWORD 0x8e7
First Counter REG_DWORD 0x844
First Help REG_DWORD 0x845
Library Validation Code REG_BINARY 0C94376029D9C301101D000000000000
WbemAdapFileTime REG_BINARY 009862045B36C301
WbemAdapFileSize REG_DWORD 0x1d10
WbemAdapStatus REG_DWORD 0x0

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Security
Security REG_BINARY
01001480A0000000AC000000140000003000000002001C0001000000002801400FF010F0001010000000000
0100000000020070000400000000001800FD0102000101000000000051200000074006F0000001C00FF01
0F000102000000000005200000002002000072007300000018008D0102000101000000000050B000000200
2000000001C00FD0102000102000000000005200000002302000072007300010100000000000512000000001
0100000000000512000000

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Enum
0 REG_SZ Root\LEGACY_W3SVC\0000

Count REG_DWORD 0x1
NextInstance REG_DWORD 0x1

System Information

Common to 8 clients except for System Name, IP address and MAC address.

Mesures\NS5080\CLIENT05_omsinfo32.txt

System Information report written at: 06/09/2004 03:59:26 PM
[System Information]

[Following are sub-categories of this main category]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Server
Version	5.0.2195 Service Pack 4 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	CLIENT05
System Manufacturer	NEC
System Model	Express5800/120Rf-2 [N8100-878E]
System Type	X86-based PC
Processor	x86 Family 15 Model 2 Stepping 9 GenuineIntel ~2392 Mhz
Processor	x86 Family 15 Model 2 Stepping 9 GenuineIntel ~2392 Mhz
BIOS Version	SWV25 v1.0.01
Windows Directory	C:\WINNT
System Directory	C:\WINNT\system32
Boot Device	\Device\Harddisk0\Partition2
Locale	United States
User Name	CLIENT05\Administrator
Time Zone	Romance Daylight Time
Total Physical Memory	1,048,024 KB
Available Physical Memory	890,868 KB
Total Virtual Memory	4,091,656 KB
Available Virtual Memory	3,870,300 KB
Page File Space	3,043,632 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
0x2000-0x4FFF	Intel(R) E7000 Series Hub Interface C PCI-to-PCI Bridge - 2545	OK
0x2000-0x4FFF	Intel(R) P64H2 PCI to PCI Bridge - 1460	OK
0x2000-0x4FFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0x3000-0x4FFF	Intel(R) P64H2 PCI to PCI Bridge - 1460	OK
0x3000-0x4FFF	Adaptec AIC-7902 - Ultra320 SCSI	OK
0x3800-0x38FF	Adaptec AIC-7902 - Ultra320 SCSI	OK
0x4000-0x40FF	Adaptec AIC-7902 - Ultra320 SCSI	OK
0x3400-0x34FF	Adaptec AIC-7902 - Ultra320 SCSI	OK
0x2440-0x247F	Intel(R) 82546EB Based Dual Port Network Connection	OK
0x2400-0x243F	Intel(R) 82546EB Based Dual Port Network Connection #2	OK
0x2800-0x281F	Intel(R) PRO/1000 XT Server Adapter	OK
0x5020-0x503F	Intel(R) 82801CA/CAM USB Universal Host Controller - 2482	OK
0x5000-0x501F	Intel(R) 82801CA/CAM USB Universal Host Controller - 2484	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
0x0092-0x0092	Motherboard resources	OK
0x00B2-0x00B3	Motherboard resources	OK
0x03F0-0x03F1	Motherboard resources	OK
0x0400-0x04BF	Motherboard resources	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
0x0500-0x052F	Motherboard resources	OK
0x0532-0x053F	Motherboard resources	OK
0x0540-0x055F	Motherboard resources	OK
0x0560-0x057F	Motherboard resources	OK
0x05A0-0x05BF	Motherboard resources	OK
0x0CA2-0x0CA5	Motherboard resources	OK
0x0020-0x0021	Programmable interrupt controller	OK
0x00A0-0x00A1	Programmable interrupt controller	OK
0x0024-0x0025	Programmable interrupt controller	OK
0x0028-0x0029	Programmable interrupt controller	OK
0x002C-0x002D	Programmable interrupt controller	OK
0x0030-0x0031	Programmable interrupt controller	OK
0x0034-0x0035	Programmable interrupt controller	OK
0x0038-0x0039	Programmable interrupt controller	OK
0x003C-0x003D	Programmable interrupt controller	OK
0x0050-0x0052	Programmable interrupt controller	OK
0x00A4-0x00A5	Programmable interrupt controller	OK
0x00A8-0x00A9	Programmable interrupt controller	OK
0x00AC-0x00AD	Programmable interrupt controller	OK
0x00B0-0x00B1	Programmable interrupt controller	OK
0x00B4-0x00B5	Programmable interrupt controller	OK
0x00B8-0x00B9	Programmable interrupt controller	OK
0x00BC-0x00BD	Programmable interrupt controller	OK
0x0080-0x0080	Direct memory access controller	OK
0x0081-0x0083	Direct memory access controller	OK
0x0084-0x0086	Direct memory access controller	OK
0x0087-0x0087	Direct memory access controller	OK
0x0088-0x0088	Direct memory access controller	OK
0x0089-0x008B	Direct memory access controller	OK
0x008C-0x008E	Direct memory access controller	OK
0x008F-0x008F	Direct memory access controller	OK
0x0090-0x0091	Direct memory access controller	OK
0x0093-0x009F	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
0x0072-0x0073	System CMOS/real time clock	OK
0x0074-0x0075	System CMOS/real time clock	OK
0x0076-0x0077	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	Intel(R) 82801CA Ultra ATA Storage Controller - 248B	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
0x0580-0x059F	Intel(R) 82801CA/CAM SMBus Controller - 2483	OK

[IRQs]

IRQ Number	Device
9	Microsoft ACPI-Compliant System
50	Adaptec AIC-7902 - Ultra320 SCSI
49	Adaptec AIC-7902 - Ultra320 SCSI
30	Intel(R) 82546EB Based Dual Port Network Connection
31	Intel(R) 82546EB Based Dual Port Network Connection #2
24	Intel(R) PRO/1000 XT Server Adapter
27	QLogic QLA23xx PCI Fibre Channel Adapter
16	Intel(R) 82801CA/CAM USB Universal Host Controller - 2482
19	Intel(R) 82801CA/CAM USB Universal Host Controller - 2484
17	ATI Technologies Inc. RAGE XL PCI
11	Motherboard resources
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)
14	Primary IDE Channel
15	Secondary IDE Channel

[Memory]

Range	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	ATI Technologies Inc. RAGE XL PCI	OK
0x40000000-0xFEBFFFFF	PCI bus	OK
0xFFC00000-0xFFFFFFFF	PCI bus	OK
0xFFC00000-0xFFFFFFFF	Intel(r) 82802 Firmware Hub Device	OK
0xFE500000-0xFEAFFFFF	Intel(R) E7000 Series Hub Interface C PCI-to-PCI Bridge - 2545	OK
0xFE500000-0xFEAFFFFF	Intel(R) P64H2 PCI to PCI Bridge - 1460	OK
0xFC000000-0xFC2FFFFF	Intel(R) E7000 Series Hub Interface C PCI-to-PCI Bridge - 2545	OK
0xFC000000-0xFC2FFFFF	Intel(R) P64H2 PCI to PCI Bridge - 1460	OK
0xFEAE0000-0xFEAE0FFF	Intel(R) P64H2 I/O Advanced Programmable Interrupt Controller - 1461	OK
0xFE700000-0xFE9FFFFF	Intel(R) P64H2 PCI to PCI Bridge - 1460	OK
0xFC100000-0xFC1FFFFF	Intel(R) P64H2 PCI to PCI Bridge - 1460	OK
0xFE9E0000-0xFE9E1FFF	Adaptec AIC-7902 - Ultra320 SCSI	OK
0xFE9F0000-0xFE9F1FFF	Adaptec AIC-7902 - Ultra320 SCSI	OK
0xFEAF0000-0xFEAF0FFF	Intel(R) P64H2 I/O Advanced Programmable Interrupt Controller - 1461	OK
0xFE560000-0xFE57FFFF	Intel(R) 82546EB Based Dual Port Network Connection	OK
0xFE580000-0xFE59FFFF	Intel(R) 82546EB Based Dual Port Network Connection #2	OK
0xFE680000-0xFE69FFFF	Intel(R) PRO/1000 XT Server Adapter	OK
0xFE660000-0xFE67FFFF	Intel(R) PRO/1000 XT Server Adapter	OK
0xFE6E0000-0xFE6E0FFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0xFD000000-0xFDFFFFFF	ATI Technologies Inc. RAGE XL PCI	OK
0xFE3F0000-0xFE3F0FFF	ATI Technologies Inc. RAGE XL PCI	OK
0xFFF00000-0xFFFFFFFF	Intel(r) 82802 Firmware Hub Device	OK
0xFFE00000-0xFFEFFFFF	Intel(r) 82802 Firmware Hub Device	OK
0xFFD00000-0xFFDFFFFF	Intel(r) 82802 Firmware Hub Device	OK
0xFEBFFC00-0xFEBFFFFF	Intel(R) 82801CA Ultra ATA Storage Controller - 248B	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\iac25_32.ax	Intel Corporation	Indeo® audio software		OK	C:\WINNT\system32\IAC25_32.AX	2.05.53	195.00 KB (199,680 bytes)		1/1/1980 1:00:00 AM
c:\winnt\system32\msg723.acm	Microsoft Corporation			OK	C:\WINNT\system32\MSG723.ACM	4.4.3385	106.77 KB (109,328 bytes)		1/12/2004 5:32:11 PM
c:\winnt\system32\lhacm.acm	Microsoft Corporation			OK	C:\WINNT\system32\LHACM.ACM	4.4.3385	33.27 KB (34,064 bytes)		1/12/2004 5:32:12 PM
c:\winnt\system32\tssoft32.acm	DSP GROUP, INC.			OK	C:\WINNT\system32\TSSOFT32.ACM	1.01	9.27 KB (9,488 bytes)		1/1/1980 1:00:00 AM
c:\winnt\system32\msg711.acm	Microsoft Corporation			OK	C:\WINNT\system32\MSG711.ACM	5.00.2134.1	10.27 KB (10,512 bytes)		1/1/1980 1:00:00 AM
c:\winnt\system32\imaadp32.acm	Microsoft Corporation			OK	C:\WINNT\system32\IMAADP32.ACM	5.00.2195.6612	16.27 KB (16,656 bytes)		1/1/1980 1:00:00 AM
c:\winnt\system32\msgsm32.acm	Microsoft Corporation			OK	C:\WINNT\system32\MSGSM32.ACM	5.00.2134.1	22.27 KB (22,800 bytes)		1/1/1980 1:00:00 AM
c:\winnt\system32\msadp32.acm	Microsoft Corporation			OK	C:\WINNT\system32\MSADP32.ACM	5.00.2134.1	14.77 KB (15,120 bytes)		1/1/1980 1:00:00 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)		1/1/1980 1:00:00 AM
c:\winnt\system32\iccvid.dll	Radius Inc.			OK	C:\WINNT\system32\ICCVID.DLL	1.10.0.6	108.00 KB (110,592 bytes)		1/1/1980 1:00:00 AM
c:\winnt\system32\msh263.drv	Microsoft Corporation				OK	C:\WINNT\system32\MSH263.DRV	4.4.3385	252.27 KB (258,320 bytes)	1/12/2004 5:32:05 PM
c:\winnt\system32\msrle32.dll	Microsoft Corporation				OK	C:\WINNT\system32\MSRLE32.DLL	5.00.2195.6612	10.77 KB (11,024 bytes)	1/1/1980 1:00:00 AM
c:\winnt\system32\msvidc32.dll	Microsoft Corporation				OK	C:\WINNT\system32\MSVIDC32.DLL	5.00.2134.1	27.27 KB (27,920 bytes)	1/1/1980 1:00:00 AM
c:\winnt\system32\msh261.drv	Microsoft Corporation				OK	C:\WINNT\system32\MSH261.DRV	4.4.3385	163.77 KB (167,696 bytes)	1/12/2004 5:32:12 PM
c:\winnt\system32\ir32_32.dll	Intel(R) Corporation			OK	C:\WINNT\system32\IR32_32.DLL	Not Available	194.50 KB (199,168 bytes)		1/1/1980 1:00:00 AM

[CD-ROM]

Item	Value
Drive	D:
Description	CD-ROM Drive
Media Loaded	False
Media Type	CD-ROM
Name	MATSHITA DVD-ROM SR-8177
Manufacturer	(Standard CD-ROM drives)
Status	OK
Transfer Rate	Not Available
SCSI Target ID	0
PNP Device ID	IDE\CDROMMATSHITA_DVD-ROM_SR-8177_____NZ21____\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 75 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	0000040c

PNP Device ID ACPI\PNP0303\4&1E30281&0
NumberOfFunctionKeys 12

[Pointing Device]

Item Value
Hardware Type Logitech PS/2 Port Mouse
Number of Buttons 5
Status OK
PNP Device ID ACPI\PNP0F12\4&1E30281&0
Power Management Supported 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] Intel(R) PRO/1000 XT Server Adapter
Adapter Type Ethernet 802.3
Product Name Intel(R) PRO/1000 XT Server Adapter
Installed True
PNP Device ID PCI\VEN_8086&DEV_1008&SUBSYS_11078086&REV_02\5&279870C6&0&40F818
Last Reset 6/9/2004 2:42:44 PM
Index 0
Service Name E1000
IP Address
IP Subnet
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:D9:B9:53
Service Name E1000
IRQ Number 24
I/O Port 0x2800-0x281F
Driver c:\winnt\system32\drivers\e1000nt5.sys (130416, 7.3.13.0)

Name [00000001] 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&279870C6&0&38F818
Last Reset 6/9/2004 2:42:44 PM
Index 1
Service Name E1000
IP Address 199.182.101.15
IP Subnet 255.255.255.0
Default IP Gateway 199.182.101.200
DHCP Enabled False
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:07:E9:25:1A:AA
Service Name E1000
IRQ Number 30
I/O Port 0x2440-0x247F
Driver c:\winnt\system32\drivers\e1000nt5.sys (130416, 7.3.13.0)

Name [00000002] 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&279870C6&0&39F818
Last Reset 6/9/2004 2:42:44 PM
Index 2
Service Name E1000
IP Address
IP Subnet
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:E9:25:1A:AB
Service Name E1000
IRQ Number 31
I/O Port 0x2400-0x243F
Driver c:\winnt\system32\drivers\e1000nt5.sys (130416, 7.3.13.0)

Name [00000003] RAS Async Adapter
Adapter Type Not Available
Product Name RAS Async Adapter
Installed True
PNP Device ID Not Available
Last Reset 6/9/2004 2:42:44 PM
Index 3
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 [00000004] WAN Miniport (L2TP)
Adapter Type Not Available
Product Name WAN Miniport (L2TP)
Installed True
PNP Device ID ROOT\MS_L2TPMINIPORT\0000
Last Reset 6/9/2004 2:42:44 PM
Index 4
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 (52112, 5.00.2195.6655)

Name [00000005] 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 6/9/2004 2:42:44 PM
Index 5
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 (48464, 5.00.2195.6711)

Name [00000006] Direct Parallel
Adapter Type Not Available
Product Name Direct Parallel
Installed True
PNP Device ID ROOT\MS_PTIMINIPORT\0000

Last Reset 6/9/2004 2:42:44 PM
Index 6
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 [00000007] WAN Miniport (IP)
Adapter Type Not Available
Product Name WAN Miniport (IP)
Installed True
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 6/9/2004 2:42:44 PM
Index 7
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 (93360, 5.00.2195.6699)

[Protocol]

Item Value
Name MSAFD Tcpip [TCP/IP]
ConnectionlessService False
GuaranteesDeliveryTrue
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
GuaranteesDeliveryFalse
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
GuaranteesDeliveryFalse
GuaranteesSequencing False
MaximumAddressSize 16 bytes
MaximumMessageSize 65467 bytes
MessageOriented True
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption True
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting True

Name RSVP TCP Service Provider
ConnectionlessService False
GuaranteesDeliveryTrue
GuaranteesSequencing True
MaximumAddressSize 16 bytes
MaximumMessageSize 0 bytes
MessageOriented False
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption True
SupportsExpeditedData True
SupportsGracefulClosing True
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{8878F9F1-628F-4EEB-8184-CB7A14131206}] SEQPACKET 0
ConnectionlessService False
GuaranteesDeliveryTrue
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{8878F9F1-628F-4EEB-8184-CB7A14131206}] DATAGRAM 0
ConnectionlessService True
GuaranteesDeliveryFalse
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{877806A8-F22A-4F60-85EC-0D6CC71D0B54}] SEQPACKET 1
ConnectionlessService False
GuaranteesDeliveryTrue
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True

MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{877806A8-F22A-4F60-85EC-0D6CC71D0B54}] DATAGRAM 1
ConnectionlessService True
GuaranteesDeliveryFalse
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{856681A2-EDEC-4BA5-BFB3-326D652217F6}] SEQPACKET 2
ConnectionlessService False
GuaranteesDeliveryTrue
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{856681A2-EDEC-4BA5-BFB3-326D652217F6}] DATAGRAM 2
ConnectionlessService True
GuaranteesDeliveryFalse
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{FFC97F40-DC1C-45C7-965A-79642CB7A1A6}] SEQPACKET 3
ConnectionlessService False
GuaranteesDeliveryTrue
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{FFC97F40-DC1C-45C7-965A-79642CB7A1A6}] DATAGRAM 3
ConnectionlessService True
GuaranteesDeliveryFalse
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{AB37E84F-E4C5-4225-B058-DB7150DC670E}] SEQPACKET 4
ConnectionlessService False
GuaranteesDeliveryTrue
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{AB37E84F-E4C5-4225-B058-DB7150DC670E}] DATAGRAM 4
ConnectionlessService True
GuaranteesDeliveryFalse
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

[WinSock]

Item Value
File c:\winnt\system32\winsock.dll
Version 3.10
Size 2.80 KB (2,864 bytes)

File c:\winnt\system32\wsock32.dll
Version 5.00.2195.6603
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 (62736, 5.00.2195.6655)

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 (62736, 5.00.2195.6655)

[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	16.00 GB (17,182,609,408 bytes)
Free Space	3.16 GB (3,387,724,288 bytes)
Volume Name	
Volume Serial Number	D47C7F59
Partition	Disk #0, Partition #1
Partition Size	16.00 GB (17,182,609,920 bytes)
Starting Offset	57576960 bytes
Drive Description	Disk drive
Drive Manufacturer	(Standard disk drives)
Drive Model	SEAGATE ST336753LC SCSI Disk Device
Drive BytesPerSector	512
Drive MediaLoaded	True
Drive MediaType	Fixed hard disk media
Drive Partitions	2
Drive SCSIBus	0
Drive SCSILogicalUnit	0
Drive SCSIPort	3
Drive SCSITargetId	0
Drive SectorsPerTrack	63
Drive Size	36413314560 bytes
Drive TotalCylinders	4427
Drive TotalSectors	71119755
Drive TotalTracks	1128885
Drive TracksPerCylinder	255

Drive	K:
Description	Network Connection
Provider Name	\\tpcc-master\kit

[SCSI]

Item	Value
Name	Adaptec AIC-7902 - Ultra320 SCSI
Caption	Adaptec AIC-7902 - Ultra320 SCSI
Driver	adpu320
Status	OK
PNP Device ID	PCI\VEN_9005&DEV_801F&SUBSYS_81B41033&REV_03\5&36D4E837&0&38E818
Device ID	PCI\VEN_9005&DEV_801F&SUBSYS_81B41033&REV_03\5&36D4E837&0&38E818

Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 50
I/O Port 0x3800-0x38FF
I/O Port 0x4000-0x40FF
Driver c:\winnt\system32\drivers\adpu320.sys (120995, 1.1.000.000)

Name Adaptec AIC-7902 - Ultra320 SCSI
Caption Adaptec AIC-7902 - Ultra320 SCSI
Driver adpu320
Status OK
PNP Device ID PCI\VEN_9005&DEV_801F&SUBSYS_81B41033&REV_03\5&36D4E837&0&39E818
Device ID PCI\VEN_9005&DEV_801F&SUBSYS_81B41033&REV_03\5&36D4E837&0&39E818
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 49
I/O Port 0x3000-0x4FFF
I/O Port 0x3400-0x34FF
Driver c:\winnt\system32\drivers\adpu320.sys (120995, 1.1.000.000)

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&279870C6&0&48F818
Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&279870C6&0&48F818
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 27
I/O Port 0x2000-0x4FFF
Driver c:\winnt\system32\drivers\ql2300.sys (445858, 8.2.2.10 (W2K VI))

[Printing]

Name Port Name Server Name
No printing information

[Problem Devices]

Device PNP Device ID Error Code
No Problem Devices

[USB]

Device PNP Device ID
Intel(R) 82801CA/CAM USB Universal Host Controller - 2482 PCI\VEN_8086&DEV_2482&SUBSYS_81B41033&REV_02\3&267A616A&0&E8
USB Root Hub USB\ROOT_HUB\4&42B666F&0
Intel(R) 82801CA/CAM USB Universal Host Controller - 2484 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	Not Available	Kernel Driver	False	Disabled	Stopped	OK	Normal	False	False		
adpu320	adpu320	c:\winnt\system32\drivers\adpu320.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		
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	c:\winnt\system32\drivers\intelide.sys	Kernel Driver	True	Boot	Running	OK	Normal	False	True		
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		
irenum	IR Enumerator Service	c:\winnt\system32\drivers\irenum.sys	Kernel Driver	False	Manual	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		
ndisuio	NDIS Usermode I/O Protocol	c:\winnt\system32\drivers\ndisuio.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	False	False		
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		
nechwid	nechwid	c:\winnt\system32\drivers\nechwid.sys	Kernel Driver	False	Manual	Stopped	OK	Normal	False	False		
necras	NEC Baseboard Management Controller	c:\winnt\system32\drivers\necras.sys	Kernel Driver	True	System	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		
pciidump	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\raspptp.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		
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	False	Disabled	Stopped	OK	Normal	False	False		
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%;%SystemRoot%\System32\Wbem;C:\Program Files\Microsoft SQL Server\80\Tools\BINN	<SYSTEM>
windir	%SystemRoot%	<SYSTEM>
OS	Windows_NT	<SYSTEM>
PROCESSOR_ARCHITECTURE	x86	<SYSTEM>
PROCESSOR_LEVEL	15	<SYSTEM>
PROCESSOR_IDENTIFIER	x86 Family 15 Model 2 Stepping 9, GenuineIntel	<SYSTEM>
PROCESSOR_REVISION	0209	<SYSTEM>
NUMBER_OF_PROCESSORS	2	<SYSTEM>
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH	<SYSTEM>
TEMP	%SystemRoot%\TEMP	<SYSTEM>
TMP	%SystemRoot%\TEMP	<SYSTEM>
TEMP	%USERPROFILE%\Local Settings\Temp	CLIENT05\Administrator
TMP	%USERPROFILE%\Local Settings\Temp	CLIENT05\Administrator

[Jobs]

[Following are sub-categories of this main category]

[Print]

Document	Size	Owner	Notify	Status	Time Submitted	Start Time	Until Time	Elapsed Time	Pages Printed	Job ID	Priority	Parameters
	Driver Name	Print Processor	Host Print Queue	Data Type	Name							
Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown	Unknown
	Unknown	Unknown	Unknown	Unknown	Unknown							

[Network Connections]

Local Name	Remote Name	Type	Status	User Name
K:	\\tpcc-master\kit	Disk	OK	CLIENT05\Administrator

[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	164	11	204800	1413120	6/9/2004 12:43:07 PM	5.00.2195.6601	44.77 KB (45,840 bytes)	1/1/1980 1:00:00 AM		
csrss.exe	Not Available	192	13	Not Available	Not Available	6/9/2004 12:43:10 PM	Unknown	Unknown	Unknown		
winlogon.exe	c:\winnt\system32\winlogon.exe	188	13	204800	1413120	6/9/2004 12:43:11 PM	5.00.2195.6714	176.77 KB (181,008 bytes)	1/1/1980 1:00:00 AM		
services.exe	c:\winnt\system32\services.exe	240	9	204800	1413120	6/9/2004 12:43:11 PM	5.00.2195.6700	87.27 KB (89,360 bytes)	1/1/1980 1:00:00 AM		
lsass.exe	c:\winnt\system32\lsass.exe	252	9	204800	1413120	6/9/2004 12:43:11 PM	5.00.2195.6695	32.77 KB (33,552 bytes)	1/1/1980 1:00:00 AM		
svchost.exe	c:\winnt\system32\svchost.exe	412	8	204800	1413120	6/9/2004 12:43:13 PM	5.00.2134.1	7.77 KB (7,952 bytes)	1/1/1980 1:00:00 AM		
msdtc.exe	c:\winnt\system32\msdtc.exe	436	8	204800	1413120	6/9/2004 12:43:13 PM	1999.9.3421.3	6.77 KB (6,928 bytes)	1/12/2004 5:30:27 PM		
svchost.exe	c:\winnt\system32\svchost.exe	564	8	204800	1413120	6/9/2004 12:43:13 PM	5.00.2134.1	7.77 KB (7,952 bytes)	1/1/1980 1:00:00 AM		
llssrv.exe	c:\winnt\system32\llssrv.exe	592	9	204800	1413120	6/9/2004 12:43:14 PM	5.00.2195.6697	81.77 KB (83,728 bytes)	1/1/1980 1:00:00 AM		
regsvc.exe	c:\winnt\system32\regsvc.exe	664	8	204800	1413120	6/9/2004 12:43:14 PM	5.00.2195.6701	66.77 KB (68,368 bytes)	1/1/1980 1:00:00 AM		
mstask.exe	c:\winnt\system32\mstask.exe	684	8	204800	1413120	6/9/2004 12:43:14 PM	4.71.2195.6704	116.77 KB (119,568 bytes)	1/12/2004 5:32:07 PM		
tcpsvcs.exe	c:\winnt\system32\tcpsvcs.exe	728	8	204800	1413120	6/9/2004 12:43:15 PM	5.00.2134.1	24.77 KB (25,360 bytes)	1/1/1980 1:00:00 AM		
winmgmt.exe	c:\winnt\system32\wbem\winmgmt.exe	764	8	204800	1413120	6/9/2004 12:43:15 PM	1.50.1085.0100	192.10 KB (196,706 bytes)	1/1/1980 1:00:00 AM		
dfssvc.exe	c:\winnt\system32\dfssvc.exe	792	8	204800	1413120	6/9/2004 12:43:15 PM	5.00.2195.6664	88.77 KB (90,896 bytes)	1/1/1980 1:00:00 AM		
inetinfo.exe	c:\winnt\system32\inetsrv\inetinfo.exe	808	8	204800	1413120	6/9/2004 12:43:15 PM	5.00.0984	14.27 KB (14,608 bytes)	1/12/2004 5:30:22 PM		
svchost.exe	c:\winnt\system32\svchost.exe	1032	8	204800	1413120	6/9/2004 12:43:27 PM	5.00.2134.1	7.77 KB (7,952 bytes)	1/1/1980 1:00:00 AM		
explorer.exe	c:\winnt\explorer.exe	848	8	204800	1413120	6/9/2004 12:49:35 PM	5.00.3700.6690	237.77 KB (243,472 bytes)	1/1/1980 1:00:00 AM		
internat.exe	c:\winnt\system32\internat.exe	1008	8	204800	1413120	6/9/2004 12:49:36 PM	5.00.2920.0000	20.27 KB (20,752 bytes)	1/1/1980 1:00:00 AM		
cmd.exe	c:\winnt\system32\cmd.exe	1144	8	204800	1413120	6/9/2004 3:56:18 PM	5.00.2195.6656	230.77 KB (236,304 bytes)	1/1/1980 1:00:00 AM		
msinfo-au3.exe	k:\outils\msinfo-au3.exe	468	8	204800	1413120	6/9/2004 3:58:49 PM	3, 0, 100, 0	81.99 KB (83,955 bytes)	5/13/2004 5:16:16 PM		
mmc.exe	c:\winnt\system32\mmc.exe	512	8	204800	1413120	6/9/2004 3:58:50 PM	5.00.2195.6601	589.27 KB (603,408 bytes)	1/1/1980 1:00:00 AM		

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
wbemprox.dll	1.50.1085.0100	40.10 KB (41,061 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wbem\wbemprox.dll

mlang.dll	5.00.3700.6655	510.77 KB (523,024 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\mlang.dll
cabinet.dll	5.00.2147.1	54.77 KB (56,080 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\cabinet.dll
msinfo32.dll	5.00.2195.6601	312.27 KB (319,760 bytes)	1/12/2004 5:32:08 PM	Microsoft Corporation	c:\program files\common files\microsoft shared\msinfo\msinfo32.dll
mmcndmgr.dll	5.00.2195.6601	816.27 KB (835,856 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\mmcndmgr.dll
mmc.exe	5.00.2195.6601	589.27 KB (603,408 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\mmc.exe
msinfo-au3.exe	3, 0, 100, 0	81.99 KB (83,955 bytes)	5/13/2004 5:16:16 PM	Not Available	k:\outils\msinfo-au3.exe
cmd.exe	5.00.2195.6656	230.77 KB (236,304 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\cmd.exe
internat.exe	5.00.2920.0000	20.27 KB (20,752 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\internat.exe
shdoclc.dll	5.00.3700.6668	324.50 KB (332,288 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\shdoclc.dll
wininet.dll	5.00.3700.6713	455.77 KB (466,704 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wininet.dll
faxshell.dll	5.00.2134.1	8.27 KB (8,464 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\faxshell.dll
msacm32.dll	5.00.2134.1	65.27 KB (66,832 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\msacm32.dll
avifil32.dll	5.00.2195.6612	76.77 KB (78,608 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\avifil32.dll
msvfw32.dll	5.00.2195.6612	113.77 KB (116,496 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\msvfw32.dll
docprop2.dll	5.00.2178.1	297.77 KB (304,912 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\docprop2.dll
urlmon.dll	5.00.3700.6705	442.77 KB (453,392 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\urlmon.dll
browselc.dll	5.00.3700.6661	34.50 KB (35,328 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\browselc.dll
linkinfo.dll	5.00.2134.1	15.77 KB (16,144 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\linkinfo.dll
imm32.dll	5.00.2195.6655	94.27 KB (96,528 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\imm32.dll
indicdll.dll	5.00.2920.0000	11.27 KB (11,536 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\indicdll.dll
powrprof.dll	5.00.3502.6601	13.27 KB (13,584 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\powrprof.dll
batmeter.dll	5.00.3502.6601	20.27 KB (20,752 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\batmeter.dll
stobject.dll	5.00.2195.6601	79.27 KB (81,168 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\stobject.dll
msi.dll	2.0.2600.1183	1.92 MB (2,017,792 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\msi.dll
webcheck.dll	5.00.3502.6601	251.77 KB (257,808 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\webcheck.dll
ntshrui.dll	5.00.2134.1	46.77 KB (47,888 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntshrui.dll
mydocs.dll	5.00.3502.6601	55.77 KB (57,104 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\mydocs.dll
browseui.dll	5.00.3700.6661	789.27 KB (808,208 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\browseui.dll
shdocvw.dll	5.00.3700.6668	1.06 MB (1,107,728 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\shdocvw.dll
explorer.exe	5.00.3700.6690	237.77 KB (243,472 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\explorer.exe
h323.tsp	5.00.2195.6699	248.77 KB (254,736 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\h323.tsp
ipconf.tsp	5.00.2143.1	10.77 KB (11,024 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ipconf.tsp
ndptsp.tsp	5.00.2143.1	38.27 KB (39,184 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ndptsp.tsp
kmddsp.tsp	5.00.2150.1	17.77 KB (18,192 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\kmddsp.tsp
uniplat.dll	5.00.2195.6601	14.27 KB (14,608 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\uniplat.dll
unimdm.tsp	5.00.2195.6601	199.27 KB (204,048 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\unimdm.tsp
tapisrv.dll	5.00.2195.6666	169.27 KB (173,328 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\tapisrv.dll
iislog.dll	5.00.0984	75.27 KB (77,072 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\iislog.dll
httpext.dll	5.00.0984	240.77 KB (246,544 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\httpext.dll
md5filt.dll	5.00.0984	32.77 KB (33,552 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\md5filt.dll
gzip.dll	5.00.0984	30.27 KB (30,992 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\gzip.dll
compfilt.dll	5.00.0984	22.77 KB (23,312 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\compfilt.dll
sspifilt.dll	5.00.0984	42.77 KB (43,792 bytes)	1/12/2004 5:30:23 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\sspifilt.dll
iscomlog.dll	5.00.0984	24.27 KB (24,848 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\iscomlog.dll
lonsint.dll	5.00.0984	11.77 KB (12,048 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\lonsint.dll
inetsloc.dll	5.00.0984	20.27 KB (20,752 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsloc.dll
iisfecnv.dll	5.00.0984	7.27 KB (7,440 bytes)	1/12/2004 5:30:26 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\iisfecnv.dll
isatq.dll	5.00.0984	61.27 KB (62,736 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\isatq.dll
infocomm.dll	5.00.0984	242.27 KB (248,080 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\infocomm.dll
w3svc.dll	5.00.0984	338.27 KB (346,384 bytes)	1/12/2004 5:30:23 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\w3svc.dll
security.dll	5.00.2154.1	5.77 KB (5,904 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\security.dll
svcext.dll	5.00.0984	39.77 KB (40,720 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\svcext.dll
admexs.dll	5.00.0984	27.77 KB (28,432 bytes)	1/12/2004 5:30:21 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\admexs.dll
wamreg.dll	5.00.0984	45.77 KB (46,864 bytes)	1/12/2004 5:30:23 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\wamreg.dll
metadata.dll	5.00.0984	68.77 KB (70,416 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\metadata.dll
iismap.dll	5.00.0984	56.27 KB (57,616 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\iismap.dll
nsepm.dll	5.00.0984	43.27 KB (44,304 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\nsepm.dll
admwprox.dll	5.00.0984	31.77 KB (32,528 bytes)	1/12/2004 5:30:26 PM	Microsoft Corporation	c:\winnt\system32\admwprox.dll
coadmin.dll	5.00.0984	39.77 KB (40,720 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\coadmin.dll
iisadmin.dll	5.00.0984	15.77 KB (16,144 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\iisadmin.dll
rpcref.dll	5.00.0984	4.27 KB (4,368 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\rpcref.dll
iisrtl.dll	5.00.0984	121.27 KB (124,176 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\iisrtl.dll
inetinfo.exe	5.00.0984	14.27 KB (14,608 bytes)	1/12/2004 5:30:22 PM	Microsoft Corporation	c:\winnt\system32\inetsrv\inetinfo.exe
dfssvc.exe	5.00.2195.6664	88.77 KB (90,896 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\dfssvc.exe
wshnetbs.dll	5.00.2134.1	7.77 KB (7,952 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wshnetbs.dll
ntmarta.dll	5.00.2195.6666	100.27 KB (102,672 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntmarta.dll
provthrd.dll	1.50.1085.0000	68.07 KB (69,708 bytes)	1/12/2004 5:32:09 PM	Microsoft Corporation	c:\winnt\system32\wbem\provthrd.dll
ntevt.dll	1.50.1085.0072	192.06 KB (196,671 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wbem\ntevt.dll
psapi.dll	5.00.2134.1	28.27 KB (28,944 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\psapi.dll
framedyn.dll	1.50.1085.0076	164.07 KB (168,009 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wbem\framedyn.dll
cimwin32.dll	1.50.1085.0103	1.04 MB (1,089,637 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wbem\cimwin32.dll
wbemsvc.dll	1.50.1085.0007	40.07 KB (41,036 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wbem\wbemsvc.dll
wbemess.dll	1.50.1085.0100	364.09 KB (372,825 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wbem\wbemess.dll
fastprox.dll	1.50.1085.0100	152.10 KB (155,749 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wbem\fastprox.dll
wbemcore.dll	1.50.1085.0100	632.09 KB (647,257 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wbem\wbemcore.dll
wbemcomn.dll	1.50.1085.0100	692.09 KB (708,696 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wbem\wbemcomn.dll
winmgmt.exe	1.50.1085.0100	192.10 KB (196,706 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wbem\winmgmt.exe
simptcp.dll	5.00.2134.1	19.27 KB (19,728 bytes)	1/12/2004 5:30:26 PM	Microsoft Corporation	c:\winnt\system32\simptcp.dll
tcpsvcs.exe	5.00.2134.1	24.77 KB (25,360 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\tcpsvcs.exe
msidle.dll	5.00.2920.0000	6.27 KB (6,416 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\msidle.dll
mstask.exe	4.71.2195.6704	116.77 KB (119,568 bytes)	1/12/2004 5:32:07 PM	Microsoft Corporation	c:\winnt\system32\mstask.exe
perfos.dll	5.00.2155.1	21.27 KB (21,776 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\perfos.dll
regsvc.exe	5.00.2195.6701	66.77 KB (68,368 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\regsvc.exe
llsrpc.dll	5.00.2195.6601	47.77 KB (48,912 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\llsrpc.dll
llssrv.exe	5.00.2195.6697	81.77 KB (83,728 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\llssrv.exe
wmi.dll	5.00.2191.1	6.27 KB (6,416 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wmi.dll
netshell.dll	5.00.2195.6604	466.27 KB (477,456 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\netshell.dll
netman.dll	5.00.2195.6660	93.27 KB (95,504 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\netman.dll
ntmsdba.dll	5.00.2195.6655	169.27 KB (173,328 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntmsdba.dll
ipbootp.dll	5.00.2168.1	33.77 KB (34,576 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ipbootp.dll
cryptui.dll	5.131.2195.6628	433.27 KB (443,664 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\cryptui.dll
rastls.dll	5.00.2195.6680	98.27 KB (100,624 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\rastls.dll
raschap.dll	5.00.2195.6663	59.27 KB (60,688 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\raschap.dll
rasppp.dll	5.00.2195.6626	194.27 KB (198,928 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\rasppp.dll
rastapi.dll	5.00.2195.6604	52.77 KB (54,032 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\rastapi.dll
rasdlg.dll	5.00.2195.6625	516.77 KB (529,168 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\rasdlg.dll
netcfgx.dll	5.00.2195.6604	534.77 KB (547,600 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\netcfgx.dll
rasmans.dll	5.00.2195.6696	149.77 KB (153,360 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\rasmans.dll
sens.dll	5.00.2195.6627	37.27 KB (38,160 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\sens.dll
ntmssvc.dll	5.00.2195.6655	391.77 KB (401,168 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntmssvc.dll
es.dll	2000.2.3504.0	227.77 KB (233,232 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\es.dll
mtxoci.dll	2000.2.3504.0	103.27 KB (105,744 bytes)	1/12/2004 5:30:25 PM	Microsoft Corporation	c:\winnt\system32\mtxoci.dll
resutils.dll	5.00.2195.6702	39.77 KB (40,720 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\resutils.dll
clusapi.dll	5.00.2195.6683	54.27 KB (55,568 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\clusapi.dll
msvcp50.dll	5.00.7051	552.50 KB (565,760 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\msvcp50.dll
xolehlp.dll	1999.9.3421.3	17.27 KB (17,680 bytes)	1/12/2004 5:30:27 PM	Microsoft Corporation	c:\winnt\system32\xolehlp.dll
msdtclog.dll	2000.2.3504.0	86.77 KB (88,848 bytes)	1/12/2004 5:30:23 PM	Microsoft Corporation	c:\winnt\system32\msdtclog.dll
mtxclu.dll	2000.2.3504.0	51.27 KB (52,496 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\mtxclu.dll
msdtcprx.dll	2000.2.3504.0	690.77 KB (707,344 bytes)	1/12/2004 5:30:25 PM	Microsoft Corporation	c:\winnt\system32\msdtcprx.dll
txfaux.dll	2000.2.3504.0	388.27 KB (397,584 bytes)	1/12/2004 5:30:25 PM	Microsoft Corporation	c:\winnt\system32\txfaux.dll
msdtctm.dll	2000.2.3504.0	1.08 MB (1,131,280 bytes)	1/12/2004 5:30:25 PM	Microsoft Corporation	c:\winnt\system32\msdtctm.dll
msdtc.exe	1999.9.3421.3	6.77 KB (6,928 bytes)	1/12/2004 5:30:27 PM	Microsoft Corporation	c:\winnt\system32\msdtc.exe
clbcatq.dll	2000.2.3504.0	498.27 KB (510,224 bytes)	1/12/2004 5:30:23 PM	Microsoft Corporation	c:\winnt\system32\clbcatq.dll
rpcss.dll	5.00.2195.6702	233.77 KB (239,376 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\rpcss.dll
svchost.exe	5.00.2134.1	7.77 KB (7,952 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\svchost.exe
dssenh.dll	5.00.2195.6612	143.77 KB (147,216 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\dssenh.dll
oakley.dll	5.00.2195.6662	435.77 KB (446,224 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\oakley.dll
mfc42u.dll	6.00.9586.0	988.05 KB (1,011,764 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\mfc42u.dll
polagent.dll	5.00.2195.6655	109.27 KB (111,888 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\polagent.dll
scecli.dll	5.00.2195.6704	111.77 KB (114,448 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\scecli.dll
esent.dll	6.1.3940.31	1.08 MB (1,135,376 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\esent.dll
ntdsatq.dll	5.00.2195.6620	31.27 KB (32,016 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntdsatq.dll
ntdsa.dll	5.00.2195.6697	1016.27 KB (1,040,656 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntdsa.dll
kdcsvc.dll	5.00.2195.6627	144.77 KB (148,240 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\kdcsvc.dll
sfmapi.dll	5.00.2134.1	38.77 KB (39,696 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\sfmapi.dll
rassfm.dll	5.00.2195.6604	21.27 KB (21,776 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\rassfm.dll
rsabase.dll	5.00.2195.6619	129.27 KB (132,368 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\rsabase.dll
schannel.dll	5.00.2195.6705	144.27 KB (147,728 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\schannel.dll
netlogon.dll	5.00.2195.6695	363.27 KB (371,984 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\netlogon.dll
msv1_0.dll	5.00.2195.6680	114.77 KB (117,520 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\msv1_0.dll
kerberos.dll	5.00.2195.6666	207.77 KB (212,752 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\kerberos.dll
msprivs.dll	5.00.2195.6695	46.00 KB (47,104 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\msprivs.dll
samsrv.dll	5.00.2195.6697	380.77 KB (389,904 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\samsrv.dll
lsasrv.dll	5.00.2195.6695	506.77 KB (518,928 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\lsasrv.dll
lsass.exe	5.00.2195.6695	32.77 KB (33,552 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\lsass.exe
ntlsapi.dll	5.00.2195.6601	6.77 KB (6,928 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntlsapi.dll
wmicore.dll	5.00.2195.6611	72.77 KB (74,512 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wmicore.dll
rasadhlp.dll	5.00.2168.1	7.27 KB (7,440 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\rasadhlp.dll
winrnr.dll	5.00.2160.1	18.77 KB (19,216 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\winrnr.dll
rnr20.dll	5.00.2195.6603	35.77 KB (36,624 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\rnr20.dll
wshtcpip.dll	5.00.2195.6601	17.27 KB (17,680 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wshtcpip.dll
msafd.dll	5.00.2195.6602	106.27 KB (108,816 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\msafd.dll
mswsock.dll	5.00.2195.6603	62.77 KB (64,272 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\mswsock.dll
dhcpcsvc.dll	5.00.2195.6685	90.77 KB (92,944 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\dhcpcsvc.dll
tapi32.dll	5.00.2195.6664	123.77 KB (126,736 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\tapi32.dll
rasman.dll	5.00.2195.6604	54.77 KB (56,080 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\rasman.dll
rasapi32.dll	5.00.2195.6625	192.77 KB (197,392 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\rasapi32.dll
rtutils.dll	5.00.2168.1	43.77 KB (44,816 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\rtutils.dll
adsldpc.dll	5.00.2195.6701	130.77 KB (133,904 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\adsldpc.dll
activeds.dll	5.00.2195.6601	177.77 KB (182,032 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\activeds.dll
mprapi.dll	5.00.2181.1	79.27 KB (81,168 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\mprapi.dll
iphlpapi.dll	5.00.2195.6602	68.27 KB (69,904 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\iphlpapi.dll

msgsvc.dll	5.00.2195.6656	34.77 KB (35,600 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\msgsvc.dll
browser.dll	5.00.2195.6693	67.27 KB (68,880 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\browser.dll
alrsvc.dll	5.00.2134.1	17.77 KB (18,192 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\alrsvc.dll
trkwks.dll	5.00.2195.6623	88.27 KB (90,384 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\trkwks.dll
seclogon.dll	5.00.2195.6707	16.77 KB (17,168 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\seclogon.dll
psbase.dll	5.00.2195.6661	112.77 KB (115,472 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\psbase.dll
cryptsvc.dll	5.00.2195.6661	74.27 KB (76,048 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\cryptsvc.dll
cryptdll.dll	5.00.2195.6607	43.27 KB (44,304 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\cryptdll.dll
wkssvc.dll	5.00.2195.6692	95.77 KB (98,064 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wkssvc.dll
srvsvc.dll	5.00.2195.6697	81.77 KB (83,728 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\srvsvc.dll
cfgmgr32.dll	5.00.2134.1	16.77 KB (17,168 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\cfgmgr32.dll
dmserver.dll	2195.6605.297.3	11.77 KB (12,048 bytes)	1/1/1980 1:00:00 AM	VERITAS Software Corp.	c:\winnt\system32\dmserver.dll
icmp.dll	5.00.2134.1	7.27 KB (7,440 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\icmp.dll
lmhsvc.dll	5.00.2195.6601	9.77 KB (10,000 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\lmhsvc.dll
eventlog.dll	5.00.2195.6716	46.77 KB (47,888 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\eventlog.dll
ntdsapi.dll	5.00.2195.6666	56.27 KB (57,616 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntdsapi.dll
scesrv.dll	5.00.2195.6704	248.77 KB (254,736 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\scesrv.dll
umpnpmgr.dll	5.00.2182.1	86.27 KB (88,336 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\umpnpmgr.dll
services.exe	5.00.2195.6700	87.27 KB (89,360 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\services.exe
netmsg.dll	5.00.2137.1	152.50 KB (156,160 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\netmsg.dll
comdlg32.dll	5.00.3700.6693	235.77 KB (241,424 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\comdlg32.dll
netui2.dll	5.00.2134.1	280.27 KB (286,992 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\netui2.dll
mprui.dll	5.00.2195.6601	54.77 KB (56,080 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\mprui.dll
netui1.dll	5.00.2134.1	210.27 KB (215,312 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\netui1.dll
netui0.dll	5.00.2195.6601	70.27 KB (71,952 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\netui0.dll
ntlanman.dll	5.00.2195.6601	35.27 KB (36,112 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntlanman.dll
wzcsapi.dll	5.00.2195.6604	29.27 KB (29,968 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wzcsapi.dll
oleaut32.dll	2.40.4522	612.27 KB (626,960 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\oleaut32.dll
wzcdlg.dll	5.00.2195.6604	51.27 KB (52,496 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wzcdlg.dll
cscui.dll	5.00.2195.6705	237.27 KB (242,960 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\cscui.dll
lz32.dll	5.00.2195.6611	9.77 KB (10,000 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\lz32.dll
version.dll	5.00.2195.6623	15.77 KB (16,144 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\version.dll
rsaenh.dll	5.00.2195.6611	131.77 KB (134,928 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\rsaenh.dll
mscat32.dll	5.131 2134.1	7.77 KB (7,952 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\mscat32.dll
ole32.dll	5.00.2195.6692	972.77 KB (996,112 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ole32.dll
imagehlp.dll	5.00.2195.6613	125.77 KB (128,784 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\imagehlp.dll
wintrust.dll	5.131.2195.6624	162.27 KB (166,160 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wintrust.dll
mpr.dll	5.00.2195.6611	53.77 KB (55,056 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\mpr.dll
winspool.drv	5.00.2195.6659	111.27 KB (113,936 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\winspool.drv
winscard.dll	5.00.2195.6609	77.27 KB (79,120 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\winscard.dll
msasn1.dll	5.00.2195.6666	51.77 KB (53,008 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\msasn1.dll
crypt32.dll	5.131 2195.6661	468.27 KB (479,504 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\crypt32.dll
atl.dll	3.00.9435	73.06 KB (74,810 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\atl.dll
certcli.dll	5.00.2195.6619	132.27 KB (135,440 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\certcli.dll
wlnotify.dll	5.00.2195.6706	56.27 KB (57,616 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wlnotify.dll
cscdll.dll	5.00.2195.6713	98.77 KB (101,136 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\cscdll.dll
setupapi.dll	5.00.2195.6622	556.77 KB (570,128 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\setupapi.dll
winmm.dll	5.00.2161.1	184.77 KB (189,200 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\winmm.dll
winsta.dll	5.00.2195.6701	38.27 KB (39,184 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\winsta.dll
comctl32.dll	5.81	537.77 KB (550,672 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\comctl32.dll
shlwapi.dll	5.00.3502.6601	282.77 KB (289,552 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\shlwapi.dll
shell32.dll	5.00.3700.6705	2.27 MB (2,383,632 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\shell32.dll
msgina.dll	5.00.2195.6669	326.27 KB (334,096 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\msgina.dll
wsock32.dll	5.00.2195.6603	21.27 KB (21,776 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wsock32.dll
dnsapi.dll	5.00.2195.6680	131.77 KB (134,928 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\dnsapi.dll
wldap32.dll	5.00.2195.6666	158.27 KB (162,064 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\wldap32.dll
ws2help.dll	5.00.2134.1	17.77 KB (18,192 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ws2help.dll
ws2_32.dll	5.00.2195.6601	68.27 KB (69,904 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ws2_32.dll
samlib.dll	5.00.2195.6666	48.77 KB (49,936 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\samlib.dll
netrap.dll	5.00.2134.1	11.27 KB (11,536 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\netrap.dll
netapi32.dll	5.00.2195.6601	304.27 KB (311,568 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\netapi32.dll
profmap.dll	5.00.2195.6610	29.27 KB (29,968 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\profmap.dll
secur32.dll	5.00.2195.6695	47.77 KB (48,912 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\secur32.dll
sfc.dll	5.00.2195.6673	92.80 KB (95,024 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\sfc.dll
nddeapi.dll	5.00.2195.6661	15.77 KB (16,144 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\nddeapi.dll
userenv.dll	5.00.2195.6711	380.77 KB (389,904 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\userenv.dll
user32.dll	5.00.2195.6688	393.77 KB (403,216 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\user32.dll
gdi32.dll	5.00.2195.6660	228.27 KB (233,744 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\gdi32.dll
rpcrt4.dll	5.00.2195.6701	443.77 KB (454,416 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\rpcrt4.dll
advapi32.dll	5.00.2195.6710	378.27 KB (387,344 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\advapi32.dll
kernel32.dll	5.00.2195.6688	725.77 KB (743,184 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\kernel32.dll
msvcrt.dll	6.10.9844.0	280.05 KB (286,773 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\msvcrt.dll
winlogon.exe	5.00.2195.6714	176.77 KB (181,008 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\winlogon.exe
sfcfiles.dll	5.00.2195.6717	948.27 KB (971,024 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\sfcfiles.dll
ntdll.dll	5.00.2195.6685	480.27 KB (491,792 bytes)	1/1/1980 1:00:00 AM	Microsoft Corporation	c:\winnt\system32\ntdll.dll
smss.exe	5.00.2195.6601	44.77 KB (45,840 bytes)	1/1/1980 1:00:00 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	Stopped	Manual	Share Process	c:\winnt\system32\services.exe	Normal	LocalSystem	0		
Background Intelligent Transfer Service		BITS	Stopped	Disabled	Share Process	c:\winnt\system32\svchost.exe -k bitsgroup	Normal	LocalSystem	0		
Computer Browser	Browser	Running	Auto	Share Process	c:\winnt\system32\services.exe	Normal	LocalSystem	0			
Indexing Service	cisvc	Stopped	Disabled	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	Stopped	Disabled	Share Process	c:\winnt\system32\services.exe	Normal	LocalSystem	0			
Logical Disk Manager Administrative Service		dmadmin	Stopped	Manual	Share Process	c:\winnt\system32\dmadmin.exe /com	Normal	LocalSystem	0		
Logical Disk Manager		dmserver	Running	Auto	Share Process	c:\winnt\system32\services.exe	Normal	LocalSystem	0		
DNS Client	Dnscache	Stopped	Manual	Share Process	c:\winnt\system32\services.exe	Normal	LocalSystem	0			
Event Log	Eventlog	Running	Auto	Share Process	c:\winnt\system32\services.exe	Normal	LocalSystem	0			
COM+ Event System		EventSystem	Running	Manual	Share Process	c:\winnt\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0		
Fax Service	Fax	Stopped	Manual	Own 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			
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	Running	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 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	Stopped	Disabled	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 TCP/IP Services		SimpTcp	Running	Auto	Share Process	c:\winnt\system32\tcpsvcs.exe	Normal	LocalSystem	0		
Print Spooler	Spooler	Stopped	Disabled	Own Process	c:\winnt\system32\spoolsv.exe	Normal	LocalSystem	0			
Performance Logs and Alerts		SysmonLog	Stopped	Manual	Own Process	c:\winnt\system32\smlogsvc.exe	Normal	LocalSystem	0		
Telephony	TapiSrv	Running	Manual	Share Process	c:\winnt\system32\svchost.exe -k tapisrv		Normal	LocalSystem	0		
Terminal Services	TermService	Stopped	Disabled	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	
Automatic Updates	wuauserv	Stopped	Disabled	Share Process	c:\winnt\system32\svchost.exe -k wugroup		Normal	LocalSystem	0		
Wireless Configuration		WZCSVC	Stopped	Manual	Share Process	c:\winnt\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	

Accessories\System Tools	Default User:Accessories\System Tools	Default User	
Startup	Default User:Startup	Default User	
Accessories	All Users:Accessories	All Users	
Accessories\Accessibility	All Users:Accessories\Accessibility	All Users	
Accessories\Communications	All Users:Accessories\Communications	All Users	
Accessories\Entertainment	All Users:Accessories\Entertainment	All Users	
Accessories\Games	All Users:Accessories\Games	All Users	
Accessories\Microsoft Script Debugger	All Users:Accessories\Microsoft Script Debugger	All Users	
Accessories\System Tools	All Users:Accessories\System Tools	All Users	
Administrative Tools	All Users:Administrative Tools	All Users	
Microsoft SQL Server	All Users:Microsoft SQL Server	All Users	
Startup	All Users:Startup	All Users	
Accessories	CLIENT05\Administrator:Accessories	CLIENT05\Administrator	
Accessories\Accessibility	CLIENT05\Administrator:Accessories\Accessibility	CLIENT05\Administrator	
Accessories\Communications	CLIENT05\Administrator:Accessories\Communications	CLIENT05\Administrator	
Accessories\Communications\HyperTerminal	CLIENT05\Administrator:Accessories\Communications\HyperTerminal	CLIENT05\Administrator	
Accessories\Entertainment	CLIENT05\Administrator:Accessories\Entertainment	CLIENT05\Administrator	
Accessories\System Tools	CLIENT05\Administrator:Accessories\System Tools	CLIENT05\Administrator	
Administrative Tools	CLIENT05\Administrator:Administrative Tools	CLIENT05\Administrator	
QLogic Corporation	CLIENT05\Administrator:QLogic Corporation	CLIENT05\Administrator	
QLogic Corporation\SANblade Control VIX	CLIENT05\Administrator:QLogic Corporation\SANblade Control VIX	CLIENT05\Administrator	
Startup	CLIENT05\Administrator:Startup	CLIENT05\Administrator	

[Startup Programs]

Program	Command	User Name	Location
internat.exe	internat.exe	CLIENT05\Administrator	HKU\S-1-5-21-2052111302-1788223648-839522115-500\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

[OLE Registration]

Object	Local Server
Sound (OLE2)	sndrec32.exe
Media Clip	mplay32.exe
Video Clip	mplay32.exe /avi
MIDI Sequence	mplay32.exe /mid
Sound	Not Available
Media Clip	Not Available
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	5.00.3700.1000
Build	53700.1000
Product ID	51876-335-9285675-05635
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	5.0.2195.6710	378 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
advpack.dll	5.0.3502.6601	87 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
browselc.dll	5.0.3700.6661	35 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
browseui.dll	5.0.3700.6661	789 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
ckcnv.exe	5.0.2189.1	9 KB	12/8/1999	C:\WINNT\system32	Microsoft Corporation
comctl32.dll	5.81.3502.6601	538 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
crypt32.dll	5.131.2195.6661	468 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
enhsig.dll	<File Missing>	Not Available	Not Available	Not Available	
iemigrat.dll	<File Missing>	Not Available	Not Available	Not Available	
iesetup.dll	5.0.3502.6601	57 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
iexplore.exe	5.0.2920.0	59 KB	12/8/1999	C:\Program Files\Internet Explorer	Microsoft Corporation
imagehlp.dll	5.0.2195.6613	126 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
imghelp.dll	<File Missing>	Not Available	Not Available	Not Available	
inseng.dll	5.0.3502.6601	72 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
jobexec.dll	5.0.0.1	47 KB	12/8/1999	C:\WINNT\system32	Microsoft Corporation
jscript.dll	5.1.0.8513	476 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
jsproxy.dll	5.0.2920.0	13 KB	12/8/1999	C:\WINNT\system32	Microsoft Corporation
msaahtml.dll	<File Missing>	Not Available	Not Available	Not Available	
mshtml.dll	5.0.3700.6699	2299 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
msoss.dll	<File Missing>	Not Available	Not Available	Not Available	
msxml.dll	8.0.6730.0	502 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
occache.dll	5.0.3502.6601	86 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
ole32.dll	5.0.2195.6692	973 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
oleaut32.dll	2.40.4522.0	612 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
olepro32.dll	5.0.4522.0	160 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
rsabase.dll	5.0.2195.6619	129 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
rsaenh.dll	5.0.2195.6611	132 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
rsapi32.dll	<File Missing>	Not Available	Not Available	Not Available	
rsasig.dll	<File Missing>	Not Available	Not Available	Not Available	
schannel.dll	5.1.2195.6705	144 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
shdoc401.dll	<File Missing>	Not Available	Not Available	Not Available	
shdocvw.dll	5.0.3700.6668	1082 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
shell32.dll	5.0.3700.6705	2328 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
shlwapi.dll	5.0.3502.6601	283 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
url.dll	5.0.3502.6601	82 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
urlmon.dll	5.0.3700.6705	443 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
vbscript.dll	5.1.0.7426	428 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
webcheck.dll	5.0.3502.6601	252 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
win.com	5.0.2134.1	24 KB	12/8/1999	C:\WINNT\system32	Microsoft Corporation
wininet.dll	5.0.3700.6713	456 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
winsock.dll	3.10.0.103	3 KB	12/8/1999	C:\WINNT\system32	Microsoft Corporation
wintrust.dll	5.131.2195.6624	162 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
wsock.vxd	<File Missing>	Not Available	Not Available	Not Available	
wsock32.dll	5.0.2195.6603	21 KB	6/19/2003 2:05:04 PM	C:\WINNT\system32	Microsoft Corporation
wsock32n.dll	<File Missing>	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	16386 MB
Available Disk Space	3230 MB
Maximum Cache Size	512 MB
Available Cache Size	512 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	1/12/2004 to 12/19/2103	sha1RSA

[Other People Certificates]

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

[Publishers]

Name
No publisher information available

[Security]

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

Novascale 5080 Configuration

Services

Only the following services were activated during the measurement.

Name	Status	Startup Type	Log On As
COM+ Event System	Started	Manual	Local System
Event Log	Started	Automatic	Local System
Help and Support	Started	Automatic	Local System
Logical Disk Manager	Started	Automatic	Local System
Microsoft Search	Started	Automatic	Local System
Network Connections	Started	Manual	Local System
Network Location Awareness (NLA)	Started	Manual	Local System
NT LM Security Support Provider	Started	Manual	Local System
ONC/RPC Portmapper	Started	Automatic	Local System
Plug and Play	Started	Automatic	Local System
Remote Procedure Call (RPC)	Started	Automatic	Local System
Security Accounts Manager	Started	Automatic	Local System
Server	Started	Automatic	Local System
Shell Hardware Detection	Started	Automatic	Local System
Special Administration Console Helper	Started	Manual	Local System
System Event Notification	Started	Automatic	Local System
Windows Management Instrumentation	Started	Automatic	Local System
Workstation	Started	Automatic	Local System

Network Adapter tuning

We bind each client/server connection (i.e., 4 QLA2350 VIA NICs) onto the distinct CPU sets exclusively. To do this, apply the following settings by using the SQL Server's "Server Network Utility".

Enable protocols: "VIA" only
Vendor: Select "QLOGIC"
Listen info: Enter the following string
"0:1443[0x3],1:1453[0xc],2:1463[0x30],3:1473[0xc0]]"

Registry tuning

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qldirect\Parameters
  SRBLISTSIZE  REG_DWORD  0x400
  FLAGS  REG_DWORD  0x0
  MAXPATHSPERDEVICE  REG_DWORD  0x10
  INSPECTIONINTERVAL  REG_DWORD  0x258
  OPTIMIZATION  REG_DWORD  0x0
  MAXRETRIESPERPATH  REG_DWORD  0x3
  MAXRETRIESPERIO  REG_DWORD  0x31


HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\SuperSocketNetLib
  ProtocolList  REG_MULTI_SZ  tcp\0via
  Encrypt  REG_DWORD  0x0

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\SuperSocketNetLib\Np
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\SuperSocketNetLib\Tcp
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\SuperSocketNetLib\VIA

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\SuperSocketNetLib\VIA
  Vendor  REG_SZ  QLogic
```

```
RecognizedVendors    REG_SZ    QLogic
ListenInfo    REG_SZ    0:1443[0x3],1:1453[0xc],2:1463[0x30],3:1473[0xc0]
```

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\SuperSocketNetLib\tcp
    TcpHideFlag    REG_DWORD    0x0
    TcpDynamicPorts    REG_SZ
    TcpPort    REG_SZ    1433
```

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\I/O system
    CountOperations    REG_DWORD    0x0
```

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 used large amounts of physical memory.

System Information

Mesures\NS5080\TPCC-FAME_omsinfo32.txt

```
System Information report written at: 06/09/04 15:09:48
System Name: TPCC-FAME
[System Summary]

Item            Value
OS Name         Microsoft(R) Windows(R) Server 2003, Datacenter Edition
Version         5.2.3790 Service Pack 1, v.1159 Build 3790
OS Manufacturer Microsoft Corporation
System Name     TPCC-FAME
System Manufacturer             BULL.
System Model    NovaScale
System Type     Itanium (TM) -based System
Processor       ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1501 Mhz
Processor       ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1501 Mhz
Processor       ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1501 Mhz
Processor       ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1501 Mhz
Processor       ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1501 Mhz
Processor       ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1501 Mhz
Processor       ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1501 Mhz
Processor       ia64 Family 31 Model 1 Stepping 5 GenuineIntel ~1501 Mhz
BIOS Version/Date BULL. M504.001.06/23/2003.15:19:38., 23/06/2003
SMBIOS Version  2.3
Windows Directory C:\WINDOWS
System Directory C:\WINDOWS\system32
Boot Device     \Device\HarddiskVolume21
Locale          United States
Hardware Abstraction Layer      Version = "5.2.3790.1159 (dnsrv.040209-1620)"
User Name       TPCC-FAME\Administrator
Time Zone       Romance Daylight Time
Total Physical Memory           131 072,00 MB
Available Physical Memory       125,10 GB
Total Virtual Memory            310,95 GB
Available Virtual Memory        310,73 GB
Page File Space 182,96 GB
Page File       C:\pagefile.sys

[Hardware Resources]



[Conflicts/Sharing]

Resource        Device
I/O Port 0x0000A000-0x0000BFFF  PCI bus
I/O Port 0x0000A000-0x0000BFFF  Intel(r) 82870 Hub Interface to PCI Bridges

I/O Port 0x00000000-0x00009FFF  PCI bus
I/O Port 0x00000000-0x00009FFF  Direct memory access controller

Memory Address 0xF4000000-0xF9FFFFFF            PCI bus
Memory Address 0xF4000000-0xF9FFFFFF            Intel(r) 82870 Hub Interface to PCI Bridges

IRQ 11          Intel(r) 82870 Hot Plug controller
IRQ 11          Intel(r) 82870 Hot Plug controller
IRQ 11          Intel(r) 82870 Hot Plug controller
IRQ 11          Intel(r) 82870 Hot Plug controller
IRQ 11          Intel(r) 82870 Hot Plug controller
IRQ 11          Intel(r) 82870 Hot Plug controller
IRQ 11          Intel(r) 82870 Hot Plug controller
IRQ 11          Intel(r) 82870 Hot Plug controller

I/O Port 0x0000E000-0x0000FFFF  PCI bus
I/O Port 0x0000E000-0x0000FFFF  Intel(r) 82870 Hub Interface to PCI Bridges
I/O Port 0x0000E000-0x0000FFFF  QLogic QLA234x PCI Fibre Channel Adapter

Memory Address 0xFEC00000-0xFEC000FF            PCI bus
Memory Address 0xFEC00000-0xFEC000FF            Motherboard resources

Memory Address 0xED000000-0xEDFFFFFF            PCI bus
Memory Address 0xED000000-0xEDFFFFFF            Intel(r) 82870 Hub Interface to PCI Bridges

Memory Address 0xA0000-0xFFFFF  PCI bus
Memory Address 0xA0000-0xFFFFF  Standard VGA Graphics Adapter

Memory Address 0xFA000000-0xFAFFFFFF            PCI bus
Memory Address 0xFA000000-0xFAFFFFFF            Intel(r) 82870 Hub Interface to PCI Bridges

I/O Port 0x0000C000-0x0000CFFF  PCI bus
I/O Port 0x0000C000-0x0000CFFF  Intel(r) 82870 Hub Interface to PCI Bridges

I/O Port 0x0000D000-0x0000DFFF  PCI bus
I/O Port 0x0000D000-0x0000DFFF  Intel(r) 82870 Hub Interface to PCI Bridges

Memory Address 0xEE000000-0xF3FFFFFF            PCI bus
Memory Address 0xEE000000-0xF3FFFFFF            Intel(r) 82870 Hub Interface to PCI Bridges


[DMA]

Resource        Device          Status
Channel 4       Direct memory access controller         OK

[Forced Hardware]

Device          PNP Device ID

[I/O]
```

Resource	Device	Status
0x00000000-0x00009FFF	PCI bus	OK
0x00000000-0x00009FFF	Direct memory access controller	OK
0x00009F40-0x00009F5F	Intel(r) 82801DB/DBM USB Universal Host Controller - 24C2	OK
0x00009F80-0x00009F9F	Intel(r) 82801DB/DBM USB Universal Host Controller - 24C4	OK
0x00008800-0x000088FF	Standard VGA Graphics Adapter	OK
0x000003B0-0x000003BB	Standard VGA Graphics Adapter	OK
0x000003C0-0x000003DF	Standard VGA Graphics Adapter	OK
0x00008400-0x000084FF	Other PCI Bridge Device	OK
0x00008F00-0x00008F3F	Other PCI Bridge Device	OK
0x00008E80-0x00008EBF	Intel 8255x-based PCI Ethernet Adapter (10/100)	OK
0x00001000-0x0000100F	Intel(r) 82801DB Ultra ATA Storage Controller-24CB	OK
0x000001F0-0x000001F7	Primary IDE Channel	OK
0x000003F6-0x000003F6	Primary IDE Channel	OK
0x00000170-0x00000177	Secondary IDE Channel	OK
0x00000376-0x00000376	Secondary IDE Channel	OK
0x00000CC0-0x00000CDF	Intel(r) 82801DB SMBus Controller - 24C3	OK
0x00000010-0x0000001F	Motherboard resources	OK
0x00000022-0x0000002D	Motherboard resources	OK
0x00000030-0x0000003F	Motherboard resources	OK
0x00000044-0x0000005F	Motherboard resources	OK
0x00000062-0x00000063	Motherboard resources	OK
0x00000065-0x0000006F	Motherboard resources	OK
0x00000072-0x0000007F	Motherboard resources	OK
0x00000080-0x00000080	Motherboard resources	OK
0x00000084-0x00000086	Motherboard resources	OK
0x00000088-0x00000088	Motherboard resources	OK
0x0000008C-0x0000008E	Motherboard resources	OK
0x00000090-0x0000009F	Motherboard resources	OK
0x000000A2-0x000000BD	Motherboard resources	OK
0x000000E0-0x000000EF	Motherboard resources	OK
0x000004D0-0x000004D1	Motherboard resources	OK
0x00000C00-0x00000C07	Motherboard resources	OK
0x00000020-0x00000021	Programmable interrupt controller	OK
0x000000A0-0x000000A1	Programmable interrupt controller	OK
0x00000081-0x00000083	Direct memory access controller	OK
0x00000087-0x00000087	Direct memory access controller	OK
0x00000089-0x0000008B	Direct memory access controller	OK
0x0000008F-0x0000008F	Direct memory access controller	OK
0x000000C0-0x000000DF	Direct memory access controller	OK
0x00000040-0x00000043	System timer	OK
0x00000070-0x00000071	System CMOS/real time clock	OK
0x00000061-0x00000061	System speaker	OK
0x0000002E-0x0000002F	Motherboard resources	OK
0x000003F8-0x000003FF	Communications Port (COM1)	OK
0x0000A000-0x0000BFFF	PCI bus	OK
0x0000A000-0x0000BFFF	Intel(r) 82870 Hub Interface to PCI Bridges	OK
0x0000A800-0x0000A8FF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0x0000B000-0x0000BFFF	Intel(r) 82870 Hub Interface to PCI Bridges	OK
0x0000B800-0x0000B8FF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0x0000C000-0x0000CFFF	PCI bus	OK
0x0000C000-0x0000CFFF	Intel(r) 82870 Hub Interface to PCI Bridges	OK
0x0000C800-0x0000C8FF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0x0000D000-0x0000DFFF	PCI bus	OK
0x0000D000-0x0000DFFF	Intel(r) 82870 Hub Interface to PCI Bridges	OK
0x0000D800-0x0000D8FF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0x0000E000-0x0000FFFF	PCI bus	OK
0x0000E000-0x0000FFFF	Intel(r) 82870 Hub Interface to PCI Bridges	OK
0x0000E000-0x0000FFFF	QLogic QLA234x PCI Fibre Channel Adapter	OK
0x0000E400-0x0000E4FF	LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)	OK
0x0000E800-0x0000E8FF	LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)	OK
0x0000F000-0x0000FFFF	Intel(r) 82870 Hub Interface to PCI Bridges	OK
0x0000F400-0x0000F4FF	LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)	OK
0x0000F800-0x0000F8FF	LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)	OK

[IRQs]

Resource	Device	Status
IRQ 9	Microsoft ACPI-Compliant System	OK
IRQ 16	Intel(r) 82801DB/DBM USB Universal Host Controller - 24C2	OK
IRQ 19	Intel(r) 82801DB/DBM USB Universal Host Controller - 24C4	OK
IRQ 5	Other PCI Bridge Device	OK
IRQ 18	Intel 8255x-based PCI Ethernet Adapter (10/100)	OK
IRQ 14	Primary IDE Channel	OK
IRQ 15	Secondary IDE Channel	OK
IRQ 0	System timer	OK
IRQ 8	System CMOS/real time clock	OK
IRQ 4	Communications Port (COM1)	OK
IRQ 48	QLogic QLA23xx PCI Fibre Channel Adapter	OK
IRQ 11	Intel(r) 82870 Hot Plug controller	OK
IRQ 11	Intel(r) 82870 Hot Plug controller	OK
IRQ 11	Intel(r) 82870 Hot Plug controller	OK
IRQ 11	Intel(r) 82870 Hot Plug controller	OK
IRQ 11	Intel(r) 82870 Hot Plug controller	OK
IRQ 11	Intel(r) 82870 Hot Plug controller	OK
IRQ 11	Intel(r) 82870 Hot Plug controller	OK
IRQ 11	Intel(r) 82870 Hot Plug controller	OK
IRQ 24	QLogic QLA23xx PCI Fibre Channel Adapter	OK
IRQ 72	QLogic QLA23xx PCI Fibre Channel Adapter	OK
IRQ 120	QLogic QLA23xx PCI Fibre Channel Adapter	OK
IRQ 192	LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)	OK
IRQ 193	LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)	OK
IRQ 196	QLogic QLA234x PCI Fibre Channel Adapter	OK
IRQ 168	LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)	OK
IRQ 169	LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)	OK

[Memory]

Resource	Device	Status
0xA0000-0xFFFFF	PCI bus	OK
0xA0000-0xFFFFF	Standard VGA Graphics Adapter	OK
0xFB000000-0xFDFFFFFF	PCI bus	OK
0xFEC00000-0xFEC000FF	PCI bus	OK
0xFEC00000-0xFEC000FF	Motherboard resources	OK
0xFC000000-0xFCFFFFFF	Standard VGA Graphics Adapter	OK
0xFD1FF000-0xFD1FFFFF	Standard VGA Graphics Adapter	OK
0xFD1FE000-0xFD1FEFFF	Other PCI Bridge Device	OK
0xFD100000-0xFD17FFFF	Other PCI Bridge Device	OK
0xFD1FD000-0xFD1FDFFF	Intel 8255x-based PCI Ethernet Adapter (10/100)	OK
0xFD1A0000-0xFD1BFFFF	Intel 8255x-based PCI Ethernet Adapter (10/100)	OK
0xFDFFFC00-0xFDFFFFFF	Intel(r) 82801DB Ultra ATA Storage Controller-24CB	OK
0xFEE00000-0xFEE00FFF	Motherboard resources	OK
0xFA000000-0xFAFFFFFF	PCI bus	OK
0xFA000000-0xFAFFFFFF	Intel(r) 82870 Hub Interface to PCI Bridges	OK
0xFAFFF000-0xFAFFFFFF	Intel(r) 82870 I/OxAPIC Interrupt Controller	OK
0xFA0FF000-0xFA0FFFFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0xFA0FEC00-0xFA0FECFF	Intel(r) 82870 Hot Plug controller	OK
0xFAFFE000-0xFAFFEFFF	Intel(r) 82870 I/OxAPIC Interrupt Controller	OK
0xFA100000-0xFA1FFFFF	Intel(r) 82870 Hub Interface to PCI Bridges	OK
0xFA1FF000-0xFA1FFFFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0xFA1FEC00-0xFA1FECFF	Intel(r) 82870 Hot Plug controller	OK
0xF4000000-0xF9FFFFFF	PCI bus	OK
0xF4000000-0xF9FFFFFF	Intel(r) 82870 Hub Interface to PCI Bridges	OK
0xF9FFF000-0xF9FFFFFF	Intel(r) 82870 I/OxAPIC Interrupt Controller	OK
0xF90FFC00-0xF90FFCFF	Intel(r) 82870 Hot Plug controller	OK
0xF9FFE000-0xF9FFEFFF	Intel(r) 82870 I/OxAPIC Interrupt Controller	OK
0xF9100000-0xF91FFFFF	Intel(r) 82870 Hub Interface to PCI Bridges	OK
0xF91FF000-0xF91FFFFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0xF91FEC00-0xF91FECFF	Intel(r) 82870 Hot Plug controller	OK
0xEE000000-0xF3FFFFFF	PCI bus	OK
0xEE000000-0xF3FFFFFF	Intel(r) 82870 Hub Interface to PCI Bridges	OK
0xF3FFF000-0xF3FFFFFF	Intel(r) 82870 I/OxAPIC Interrupt Controller	OK
0xF30FFC00-0xF30FFCFF	Intel(r) 82870 Hot Plug controller	OK
0xF3FFE000-0xF3FFEFFF	Intel(r) 82870 I/OxAPIC Interrupt Controller	OK
0xF3100000-0xF31FFFFF	Intel(r) 82870 Hub Interface to PCI Bridges	OK
0xF31FF000-0xF31FFFFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0xF31FEC00-0xF31FECFF	Intel(r) 82870 Hot Plug controller	OK
0xED000000-0xEDFFFFFF	PCI bus	OK
0xED000000-0xEDFFFFFF	Intel(r) 82870 Hub Interface to PCI Bridges	OK
0xEDFFF000-0xEDFFFFFF	Intel(r) 82870 I/OxAPIC Interrupt Controller	OK
0xED4D0000-0xED4DFFFF	LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)	OK
0xED4C0000-0xED4CFFFF	LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)	OK
0xED4F0000-0xED4FFFFF	LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)	OK
0xED4E0000-0xED4EFFFF	LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)	OK
0xED4BF000-0xED4BFFFF	QLogic QLA234x PCI Fibre Channel Adapter	OK
0xED4BEC00-0xED4BECFF	Intel(r) 82870 Hot Plug controller	OK
0xEDFFE000-0xEDFFEFFF	Intel(r) 82870 I/OxAPIC Interrupt Controller	OK
0xED500000-0xED9FFFFF	Intel(r) 82870 Hub Interface to PCI Bridges	OK
0xED9D0000-0xED9DFFFF	LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)	OK
0xED9C0000-0xED9CFFFF	LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)	OK
0xED9F0000-0xED9FFFFF	LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)	OK

0xED9E0000-0xED9EFFFF LSI Adapter, FC 7002 series, Dual or Quad port (with 929X) OK
0xED9BFC00-0xED9BFCFF Intel(r) 82870 Hot Plug controller OK

[Components]

[Multimedia]

[Audio Codecs]

CODEC	Manufacturer	Description	Status	File	Version	Size	Creation Date
c:\windows\system32\msgsm32.acm	Microsoft Corporation			OK	C:\WINDOWS\system32\MSGSM32.ACM	5.2.3790.0 (srv03_rtm.030324-2048)	66,50 KB (68 096 bytes) 11/02/2004 13:00
c:\windows\system32\msg711.acm	Microsoft Corporation			OK	C:\WINDOWS\system32\MSG711.ACM	5.2.3790.0 (srv03_rtm.030324-2048)	33,00 KB (33 792 bytes) 11/02/2004 13:00
c:\windows\system32\imaadp32.acm	Microsoft Corporation			OK	C:\WINDOWS\system32\IMAADP32.ACM	5.2.3790.0 (srv03_rtm.030324-2048)	55,00 KB (56 320 bytes) 11/02/2004 13:00
c:\windows\system32\tssoft32.acm	DSP GROUP, INC.			OK	C:\WINDOWS\system32\TSSOFT32.ACM	1.01	29,00 KB (29 696 bytes) 11/02/2004 13:00
c:\windows\system32\msadp32.acm	Microsoft Corporation			OK	C:\WINDOWS\system32\MSADP32.ACM	5.2.3790.0 (srv03_rtm.030324-2048)	49,00 KB (50 176 bytes) 11/02/2004 13:00

[Video Codecs]

CODEC	Manufacturer	Description	Status	File	Version	Size	Creation Date
c:\windows\system32\msrle32.dll	Microsoft Corporation			OK	C:\WINDOWS\system32\MSRLE32.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	24,50 KB (25 088 bytes) 11/02/2004 13:00
c:\windows\system32\msvidc32.dll	Microsoft Corporation			OK	C:\WINDOWS\system32\MSVIDC32.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	67,00 KB (68 608 bytes) 11/02/2004 13:00

[CD-ROM]

Item	Value
Drive	H:
Description	CD-ROM Drive
Media Loaded	No
Media Type	CD-ROM
Name	MATSHITA DVD-ROM SR-8177
Manufacturer	(Standard CD-ROM drives)
Status	OK
Transfer Rate	Not Available
SCSI Target ID	0
PNP Device ID	IDE\CDROMMATSHITA_DVD-ROM_SR-8177_____NZ20____\5&11538027&0&0.0.0
Driver	c:\windows\system32\drivers\cdrom.sys (5.2.3790.1159 (dnsrv.040209-1620), 143,50 KB (146 944 bytes), 11/02/2004 13:00)

[Sound Device]

Item	Value

[Display]

Item	Value
Name	Standard VGA Graphics Adapter
PNP Device ID	PCI\VEN_1002&DEV_4752&SUBSYS_80081002&REV_65\4&2AF9ED5&0&00F0
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	5.2.3790.0
INF File	display.inf (vga section)
Color Planes	1
Color Table Entries	16777216
Resolution	1024 x 768 x 1 hertz
Bits/Pixel	24
Memory Address	0xFC000000-0xFCFFFFFF
I/O Port	0x00008800-0x000088FF
Memory Address	0xFD1FF000-0xFD1FFFFF
I/O Port	0x000003B0-0x000003BB
I/O Port	0x000003C0-0x000003DF
Memory Address	0xA0000-0xFFFFF
Driver	c:\windows\system32\drivers\vgapnp.sys (5.2.3790.0 (srv03_rtm.030324-2048), 68,50 KB (70 144 bytes), 09/04/2004 10:34)

[Infrared]

Item	Value

[Input]

[Keyboard]

Item	Value
Description	USB Human Interface Device
Name	Enhanced (101- or 102-key)
Layout	0000040C
PNP Device ID	USB\VID_0D3D&PID_0001&MI_00\6&1E2EBC7E&0&0000
Number of Function Keys	12
Driver	c:\windows\system32\drivers\hidusb.sys (5.2.3790.0 (srv03_rtm.030324-2048), 32,00 KB (32 768 bytes), 11/02/2004 13:00)

[Pointing Device]

Item	Value
Hardware Type	HID-compliant mouse
Number of Buttons	3
Status	OK
PNP Device ID	HID\VID_0D3D&PID_0001&MI_01&COL01\7&2CAA6DB2&0&0000
Power Management Supported	No
Double Click Threshold	6
Handedness	Right Handed Operation
Driver	c:\windows\system32\drivers\mouhid.sys (5.2.3790.0 (srv03_rtm.030324-2048), 35,50 KB (36 352 bytes), 25/03/2003 02:06)

[Modem]

Item	Value

[Network]

[Adapter]

Item	Value
Name	[00000001] Intel 8255x-based PCI Ethernet Adapter (10/100)
Adapter Type	Ethernet 802.3
Product Type	Intel 8255x-based PCI Ethernet Adapter (10/100)
Installed	Yes
PNP Device ID	PCI\VEN_8086&DEV_1229&SUBSYS_0000119F&REV_0D\4&2AF9ED5&0&10F0
Last Reset	09/06/2004 09:25
Index	1
Service Name	E100B
IP Address	199.182.101.2
IP Subnet	255.255.255.0
Default IP Gateway	199.182.101.200
DHCP Enabled	No
DHCP Server	Not Available
DHCP Lease Expires	Not Available
DHCP Lease Obtained	Not Available
MAC Address	08:00:38:35:30:42
Memory Address	0xFD1FD000-0xFD1FDFFF
I/O Port	0x00008E80-0x00008EBF
Memory Address	0xFD1A0000-0xFD1BFFFF
IRQ Channel	IRQ 18
Driver	c:\windows\system32\drivers\e100b645.sys (6.6.8.1 built by: WinDDK, 580,00 KB (593 920 bytes), 09/04/2004 10:33)

Name	[00000002] RAS Async Adapter
Adapter Type	Not Available
Product Type	RAS Async Adapter
Installed	Yes
PNP Device ID	Not Available
Last Reset	09/06/2004 09:25
Index	2
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 [00000003] WAN Miniport (L2TP)
Adapter Type Not Available
Product Type WAN Miniport (L2TP)
Installed Yes
PNP Device ID ROOT\MS_L2TPMINIPORT\0000
Last Reset 09/06/2004 09:25
Index 3
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\rasl2tp.sys (5.2.3790.1159 (dnsrv.040209-1620), 180,00 KB (184 320 bytes), 11/02/2004 13:00)

Name [00000004] 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 09/06/2004 09:25
Index 4
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30
Driver c:\windows\system32\drivers\raspptp.sys (5.2.3790.1159 (dnsrv.040209-1620), 177,00 KB (181 248 bytes), 11/02/2004 13:00)

Name [00000005] 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 09/06/2004 09:25
Index 5
Service Name RasPppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30
Driver c:\windows\system32\drivers\raspppoe.sys (5.2.3790.0 (srv03_rtm.030324-2048), 115,50 KB (118 272 bytes), 11/02/2004 13:00)

Name [00000006] Direct Parallel
Adapter Type Not Available
Product Type Direct Parallel
Installed Yes
PNP Device ID ROOT\MS_PTIMINIPORT\0000
Last Reset 09/06/2004 09:25
Index 6
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\raspti.sys (5.2.3790.0 (srv03_rtm.030324-2048), 49,50 KB (50 688 bytes), 11/02/2004 13:00)

Name [00000007] WAN Miniport (IP)
Adapter Type Not Available
Product Type WAN Miniport (IP)
Installed Yes
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 09/06/2004 09:25
Index 7
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\ndiswan.sys (5.2.3790.1159 (dnsrv.040209-1620), 249,50 KB (255 488 bytes), 11/02/2004 13:00)

[Protocol]

Item Value
Name MSAFD Tcpip [TCP/IP]
Connectionless Service No
Guarantees Delivery Yes
Guarantees Sequencing Yes
Maximum Address Size 16 bytes
Maximum Message Size 0 bytes
Message Oriented No
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data Yes
Supports Graceful Closing Yes
Supports Guaranteed Bandwidth No
Supports Multicasting No

Name MSAFD Tcpip [UDP/IP]
Connectionless Service Yes
Guarantees Delivery No
Guarantees Sequencing No
Maximum Address Size 16 bytes
Maximum Message Size 63,93 KB (65 467 bytes)
Message Oriented Yes
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting Yes

Name RSVP UDP Service Provider
Connectionless Service Yes
Guarantees Delivery No
Guarantees Sequencing No
Maximum Address Size 16 bytes
Maximum Message Size 63,93 KB (65 467 bytes)
Message Oriented Yes
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption Yes
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting Yes

Name RSVP TCP Service Provider
Connectionless Service No
Guarantees Delivery Yes

Guarantees Sequencing Yes
Maximum Address Size 16 bytes
Maximum Message Size 0 bytes
Message Oriented No
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption Yes
Supports Expedited Data Yes
Supports Graceful Closing Yes
Supports Guaranteed Bandwidth No
Supports Multicasting No

[WinSock]

Item Value
File c:\windows\system32\wsock32.dll
Size 23,00 KB (23 552 bytes)
Version 5.2.3790.0 (srv03_rtm.030324-2048)

[Ports]

[Serial]

Item Value
Name Communications Port (COM1)
Status OK
PNP Device ID ACPI\PNP0501\1
Maximum Input Buffer Size 0
Maximum Output Buffer Size No
Settable Baud Rate Yes
Settable Data Bits Yes
Settable Flow Control Yes
Settable Parity Yes
Settable Parity Check Yes
Settable Stop Bits Yes
Settable RLSD Yes
Supports RLSD Yes
Supports 16 Bit Mode No
Supports Special Characters No
Baud Rate 9600
Bits/Byte 8
Stop Bits 1
Parity None
Busy No
Abort Read/Write on Error No
Binary Mode Enabled Yes
Continue XMit on XOff No
CTS Outflow Control No
Discard NULL Bytes No
DSR Outflow Control 0
DSR Sensitivity 0
DTR Flow Control Type Enable
EOF Character 0
Error Replace Character 0
Error Replacement Enabled No
Event Character 0
Parity Check Enabled No
RTS Flow Control Type Enable
XOff Character 19
XOffXMit Threshold 512
XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control 0
I/O Port 0x000003F8-0x000003FF
IRQ Channel IRQ 4
Driver c:\windows\system32\drivers\serial.sys (5.2.3790.0 (srv03_rtm.030324-2048), 174,50 KB (178 688 bytes), 11/02/2004 13:00)

[Parallel]

Item Value

[Storage]

[Drives]

Item Value
Drive A:
Description 3 1/2 Inch Floppy Drive

Drive C:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 29,58 GB (31 757 803 520 bytes)
Free Space 23,30 GB (25 017 462 784 bytes)
Volume Name
Volume Serial Number FC8C24E7

Drive D:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 92,00 GB (98 784 243 712 bytes)
Free Space 140,63 MB (147 464 192 bytes)
Volume Name Swap1
Volume Serial Number C889960C

Drive E:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 92,00 GB (98 784 243 712 bytes)
Free Space 140,63 MB (147 464 192 bytes)
Volume Name Swap2
Volume Serial Number 889B223E

Drive F:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 33,07 GB (35 513 167 872 bytes)
Free Space 33,01 GB (35 444 469 760 bytes)
Volume Name BOOT2
Volume Serial Number 541F07FB

Drive G:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 49,67 GB (53 337 911 296 bytes)
Free Space 49,61 GB (53 268 668 416 bytes)
Volume Name TempDBLog
Volume Serial Number AC82845E

Drive H:
Description CD-ROM Disc

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 Network Connection
Provider Name \\199.182.101.200\kit

[Disks]

Item Value
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 5
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 8
SCSI Target ID 0
Sectors/Track 63
Size 499,31 GB (536 131 975 680 bytes)
Total Cylinders 65 181
Total Sectors 1 047 132 765
Total Tracks 16 621 155
Tracks/Cylinder 255
Partition Disk #6, Partition #0
Partition Size 45,00 GB (48 323 487 744 bytes)
Partition Starting Offset 32 256 bytes
Partition Disk #6, Partition #1
Partition Size 36,88 GB (39 596 497 920 bytes)
Partition Starting Offset 48 323 520 000 bytes
Partition Disk #6, Partition #2
Partition Size 20,63 GB (22 150 679 040 bytes)
Partition Starting Offset 87 920 017 920 bytes
Partition Disk #6, Partition #3
Partition Size 324,49 GB (348 422 860 800 bytes)
Partition Starting Offset 187 709 114 880 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 5
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 8
SCSI Target ID 1
Sectors/Track 63
Size 499,31 GB (536 131 975 680 bytes)
Total Cylinders 65 181
Total Sectors 1 047 132 765
Total Tracks 16 621 155
Tracks/Cylinder 255
Partition Disk #7, Partition #0
Partition Size 45,00 GB (48 323 487 744 bytes)
Partition Starting Offset 32 256 bytes
Partition Disk #7, Partition #1
Partition Size 36,88 GB (39 596 497 920 bytes)
Partition Starting Offset 48 323 520 000 bytes
Partition Disk #7, Partition #2
Partition Size 20,63 GB (22 150 679 040 bytes)
Partition Starting Offset 87 920 017 920 bytes
Partition Disk #7, Partition #3
Partition Size 324,49 GB (348 422 860 800 bytes)
Partition Starting Offset 187 709 114 880 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 5
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 8
SCSI Target ID 2
Sectors/Track 63
Size 499,31 GB (536 131 975 680 bytes)
Total Cylinders 65 181
Total Sectors 1 047 132 765
Total Tracks 16 621 155
Tracks/Cylinder 255
Partition Disk #8, Partition #0
Partition Size 45,00 GB (48 323 487 744 bytes)
Partition Starting Offset 32 256 bytes
Partition Disk #8, Partition #1
Partition Size 36,88 GB (39 596 497 920 bytes)
Partition Starting Offset 48 323 520 000 bytes
Partition Disk #8, Partition #2
Partition Size 20,63 GB (22 150 679 040 bytes)
Partition Starting Offset 87 920 017 920 bytes
Partition Disk #8, Partition #3
Partition Size 324,49 GB (348 422 860 800 bytes)
Partition Starting Offset 187 709 114 880 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 5
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 8
SCSI Target ID 3
Sectors/Track 63
Size 499,31 GB (536 131 975 680 bytes)
Total Cylinders 65 181
Total Sectors 1 047 132 765
Total Tracks 16 621 155
Tracks/Cylinder 255
Partition Disk #9, Partition #0
Partition Size 45,00 GB (48 323 487 744 bytes)
Partition Starting Offset 32 256 bytes
Partition Disk #9, Partition #1
Partition Size 36,88 GB (39 596 497 920 bytes)
Partition Starting Offset 48 323 520 000 bytes
Partition Disk #9, Partition #2
Partition Size 20,63 GB (22 150 679 040 bytes)
Partition Starting Offset 87 920 017 920 bytes
Partition Disk #9, Partition #3
Partition Size 324,49 GB (348 422 860 800 bytes)
Partition Starting Offset 187 709 114 880 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 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 6
SCSI Target ID 0
Sectors/Track 63
Size 30,00 GB (32 210 196 480 bytes)
Total Cylinders 3 916
Total Sectors 62 910 540
Total Tracks 998 580
Tracks/Cylinder 255
Partition Disk #0, Partition #0
Partition Size 305,89 MB (320 753 664 bytes)
Partition Starting Offset 32 256 bytes
Partition Disk #0, Partition #1
Partition Size 29,58 GB (31 757 806 080 bytes)
Partition Starting Offset 452 390 400 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 1

SCSI Port 6
SCSI Target ID 0
Sectors/Track 63
Size 233,00 GB (250 180 116 480 bytes)
Total Cylinders 30 416
Total Sectors 488 633 040
Total Tracks 7 756 080
Tracks/Cylinder 255
Partition Disk #1, Partition #0
Partition Size 232,88 GB (250 047 627 264 bytes)
Partition Starting Offset 134 235 136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 6
SCSI Target ID 0
Sectors/Track 63
Size 166,40 GB (178 669 532 160 bytes)
Total Cylinders 21 722
Total Sectors 348 963 930
Total Tracks 5 539 110
Tracks/Cylinder 255
Partition Disk #2, Partition #0
Partition Size 92,00 GB (98 784 247 808 bytes)
Partition Starting Offset 134 235 136 bytes
Partition Disk #2, Partition #1
Partition Size 74,28 GB (79 755 739 136 bytes)
Partition Starting Offset 98 918 482 944 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 3
SCSI Port 6
SCSI Target ID 0
Sectors/Track 63
Size 166,40 GB (178 669 532 160 bytes)
Total Cylinders 21 722
Total Sectors 348 963 930
Total Tracks 5 539 110
Tracks/Cylinder 255
Partition Disk #3, Partition #0
Partition Size 92,00 GB (98 784 247 808 bytes)
Partition Starting Offset 134 235 136 bytes
Partition Disk #3, Partition #1
Partition Size 74,28 GB (79 755 739 136 bytes)
Partition Starting Offset 98 918 482 944 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 4
SCSI Port 6
SCSI Target ID 0
Sectors/Track 63
Size 99,80 GB (107 158 947 840 bytes)
Total Cylinders 13 028
Total Sectors 209 294 820
Total Tracks 3 322 140
Tracks/Cylinder 255
Partition Disk #4, Partition #0
Partition Size 50,00 GB (53 687 091 200 bytes)
Partition Starting Offset 134 235 136 bytes
Partition Disk #4, Partition #1
Partition Size 49,67 GB (53 337 915 392 bytes)
Partition Starting Offset 53 821 326 336 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model NEC iStorage 2000 SCSI Disk Device
Bytes/Sector 512
Media Loaded No
Media Type Fixed hard disk
Partitions Not Available
SCSI Bus 0
SCSI Logical Unit 5
SCSI Port 6
SCSI Target ID 0
Sectors/Track 63
Size 33,20 GB (35 648 363 520 bytes)
Total Cylinders 4 334
Total Sectors 69 625 710
Total Tracks 1 105 170
Tracks/Cylinder 255
Partition Disk #5, Partition #0
Partition Size 33,07 GB (35 513 171 968 bytes)
Partition Starting Offset 134 235 136 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 5
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 9
SCSI Target ID 0
Sectors/Track 63
Size 499,31 GB (536 131 975 680 bytes)
Total Cylinders 65 181
Total Sectors 1 047 132 765
Total Tracks 16 621 155
Tracks/Cylinder 255
Partition Disk #10, Partition #0
Partition Size 45,00 GB (48 323 487 744 bytes)
Partition Starting Offset 32 256 bytes
Partition Disk #10, Partition #1
Partition Size 36,88 GB (39 596 497 920 bytes)
Partition Starting Offset 48 323 520 000 bytes
Partition Disk #10, Partition #2
Partition Size 20,63 GB (22 150 679 040 bytes)
Partition Starting Offset 87 920 017 920 bytes
Partition Disk #10, Partition #3
Partition Size 324,49 GB (348 422 860 800 bytes)
Partition Starting Offset 187 709 114 880 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 5
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 9
SCSI Target ID 1
Sectors/Track 63
Size 499,31 GB (536 131 975 680 bytes)
Total Cylinders 65 181
Total Sectors 1 047 132 765
Total Tracks 16 621 155
Tracks/Cylinder 255

Partition Disk #11, Partition #0
Partition Size 45,00 GB (48 323 487 744 bytes)
Partition Starting Offset 32 256 bytes
Partition Disk #11, Partition #1
Partition Size 36,88 GB (39 596 497 920 bytes)
Partition Starting Offset 48 323 520 000 bytes
Partition Disk #11, Partition #2
Partition Size 20,63 GB (22 150 679 040 bytes)
Partition Starting Offset 87 920 017 920 bytes
Partition Disk #11, Partition #3
Partition Size 324,49 GB (348 422 860 800 bytes)
Partition Starting Offset 187 709 114 880 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 5
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 9
SCSI Target ID 2
Sectors/Track 63
Size 499,31 GB (536 131 975 680 bytes)
Total Cylinders 65 181
Total Sectors 1 047 132 765
Total Tracks 16 621 155
Tracks/Cylinder 255
Partition Disk #12, Partition #0
Partition Size 45,00 GB (48 323 487 744 bytes)
Partition Starting Offset 32 256 bytes
Partition Disk #12, Partition #1
Partition Size 36,88 GB (39 596 497 920 bytes)
Partition Starting Offset 48 323 520 000 bytes
Partition Disk #12, Partition #2
Partition Size 20,63 GB (22 150 679 040 bytes)
Partition Starting Offset 87 920 017 920 bytes
Partition Disk #12, Partition #3
Partition Size 324,49 GB (348 422 860 800 bytes)
Partition Starting Offset 187 709 114 880 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 5
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 9
SCSI Target ID 3
Sectors/Track 63
Size 499,31 GB (536 131 975 680 bytes)
Total Cylinders 65 181
Total Sectors 1 047 132 765
Total Tracks 16 621 155
Tracks/Cylinder 255
Partition Disk #13, Partition #0
Partition Size 45,00 GB (48 323 487 744 bytes)
Partition Starting Offset 32 256 bytes
Partition Disk #13, Partition #1
Partition Size 36,88 GB (39 596 497 920 bytes)
Partition Starting Offset 48 323 520 000 bytes
Partition Disk #13, Partition #2
Partition Size 20,63 GB (22 150 679 040 bytes)
Partition Starting Offset 87 920 017 920 bytes
Partition Disk #13, Partition #3
Partition Size 324,49 GB (348 422 860 800 bytes)
Partition Starting Offset 187 709 114 880 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 5
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 10
SCSI Target ID 0
Sectors/Track 63
Size 499,31 GB (536 131 975 680 bytes)
Total Cylinders 65 181
Total Sectors 1 047 132 765
Total Tracks 16 621 155
Tracks/Cylinder 255
Partition Disk #14, Partition #0
Partition Size 45,00 GB (48 323 487 744 bytes)
Partition Starting Offset 32 256 bytes
Partition Disk #14, Partition #1
Partition Size 36,88 GB (39 596 497 920 bytes)
Partition Starting Offset 48 323 520 000 bytes
Partition Disk #14, Partition #2
Partition Size 20,63 GB (22 150 679 040 bytes)
Partition Starting Offset 87 920 017 920 bytes
Partition Disk #14, Partition #3
Partition Size 324,49 GB (348 422 860 800 bytes)
Partition Starting Offset 187 709 114 880 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 5
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 10
SCSI Target ID 1
Sectors/Track 63
Size 499,31 GB (536 131 975 680 bytes)
Total Cylinders 65 181
Total Sectors 1 047 132 765
Total Tracks 16 621 155
Tracks/Cylinder 255
Partition Disk #15, Partition #0
Partition Size 45,00 GB (48 323 487 744 bytes)
Partition Starting Offset 32 256 bytes
Partition Disk #15, Partition #1
Partition Size 36,88 GB (39 596 497 920 bytes)
Partition Starting Offset 48 323 520 000 bytes
Partition Disk #15, Partition #2
Partition Size 20,63 GB (22 150 679 040 bytes)
Partition Starting Offset 87 920 017 920 bytes
Partition Disk #15, Partition #3
Partition Size 324,49 GB (348 422 860 800 bytes)
Partition Starting Offset 187 709 114 880 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 5
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 10
SCSI Target ID 2
Sectors/Track 63
Size 499,31 GB (536 131 975 680 bytes)
Total Cylinders 65 181
Total Sectors 1 047 132 765
Total Tracks 16 621 155
Tracks/Cylinder 255
Partition Disk #16, Partition #0
Partition Size 45,00 GB (48 323 487 744 bytes)

Partition Starting Offset 32 256 bytes
Partition Disk #16, Partition #1
Partition Size 36,88 GB (39 596 497 920 bytes)
Partition Starting Offset 48 323 520 000 bytes
Partition Disk #16, Partition #2
Partition Size 20,63 GB (22 150 679 040 bytes)
Partition Starting Offset 87 920 017 920 bytes
Partition Disk #16, Partition #3
Partition Size 324,49 GB (348 422 860 800 bytes)
Partition Starting Offset 187 709 114 880 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 5
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 10
SCSI Target ID 3
Sectors/Track 63
Size 499,31 GB (536 131 975 680 bytes)
Total Cylinders 65 181
Total Sectors 1 047 132 765
Total Tracks 16 621 155
Tracks/Cylinder 255
Partition Disk #17, Partition #0
Partition Size 45,00 GB (48 323 487 744 bytes)
Partition Starting Offset 32 256 bytes
Partition Disk #17, Partition #1
Partition Size 36,88 GB (39 596 497 920 bytes)
Partition Starting Offset 48 323 520 000 bytes
Partition Disk #17, Partition #2
Partition Size 20,63 GB (22 150 679 040 bytes)
Partition Starting Offset 87 920 017 920 bytes
Partition Disk #17, Partition #3
Partition Size 324,49 GB (348 422 860 800 bytes)
Partition Starting Offset 187 709 114 880 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\4&207C21BC&0&08E8
I/O Port 0x0000A800-0x0000A8FF
Memory Address 0xFA0FF000-0xFA0FFFFF
IRQ Channel IRQ 48
Driver c:\windows\system32\drivers\ql2350.sys (8.2.2.10 (W64 VI), 690,75 KB (707 328 bytes), 18/02/2004 09:39)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&2AAD566B&0&08F8
I/O Port 0x0000B800-0x0000B8FF
Memory Address 0xFA1FF000-0xFA1FFFFF
IRQ Channel IRQ 24
Driver c:\windows\system32\drivers\ql2350.sys (8.2.2.10 (W64 VI), 690,75 KB (707 328 bytes), 18/02/2004 09:39)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&28BE6AA6&0&08F8
I/O Port 0x0000C800-0x0000C8FF
Memory Address 0xF91FF000-0xF91FFFFF
IRQ Channel IRQ 72
Driver c:\windows\system32\drivers\ql2350.sys (8.2.2.10 (W64 VI), 690,75 KB (707 328 bytes), 18/02/2004 09:39)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&16D68E&0&08F8
I/O Port 0x0000D800-0x0000D8FF
Memory Address 0xF31FF000-0xF31FFFFF
IRQ Channel IRQ 120
Driver c:\windows\system32\drivers\ql2350.sys (8.2.2.10 (W64 VI), 690,75 KB (707 328 bytes), 18/02/2004 09:39)

Name LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)
Manufacturer LSI Logic
Status OK
PNP Device ID PCI\VEN_1000&DEV_0626&SUBSYS_10101000&REV_00\4&1B67A8B9&0&08E8
I/O Port 0x0000E400-0x0000E4FF
Memory Address 0xED4D0000-0xED4DFFFF
Memory Address 0xED4C0000-0xED4CFFFF
IRQ Channel IRQ 192
Driver c:\windows\system32\drivers\symmpi.sys (1.09.15.00 built by: WinDDK, 109,50 KB (112 128 bytes), 29/04/2004 10:28)

Name LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)
Manufacturer LSI Logic
Status OK
PNP Device ID PCI\VEN_1000&DEV_0626&SUBSYS_10101000&REV_00\4&1B67A8B9&0&09E8
I/O Port 0x0000E800-0x0000E8FF
Memory Address 0xED4F0000-0xED4FFFFF
Memory Address 0xED4E0000-0xED4EFFFF
IRQ Channel IRQ 193
Driver c:\windows\system32\drivers\symmpi.sys (1.09.15.00 built by: WinDDK, 109,50 KB (112 128 bytes), 29/04/2004 10:28)

Name QLogic QLA234x PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\4&1B67A8B9&0&10E8
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xED4BF000-0xED4BFFFF
IRQ Channel IRQ 196
Driver c:\windows\system32\drivers\ql2300.sys (8.2.3.11 (wia64 VI), 655,00 KB (670 720 bytes), 18/02/2004 09:38)

Name LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)
Manufacturer LSI Logic
Status OK
PNP Device ID PCI\VEN_1000&DEV_0626&SUBSYS_10101000&REV_00\4&22394B12&0&08F8
I/O Port 0x0000F400-0x0000F4FF
Memory Address 0xED9D0000-0xED9DFFFF
Memory Address 0xED9C0000-0xED9CFFFF
IRQ Channel IRQ 168
Driver c:\windows\system32\drivers\symmpi.sys (1.09.15.00 built by: WinDDK, 109,50 KB (112 128 bytes), 29/04/2004 10:28)

Name LSI Adapter, FC 7002 series, Dual or Quad port (with 929X)
Manufacturer LSI Logic
Status OK
PNP Device ID PCI\VEN_1000&DEV_0626&SUBSYS_10101000&REV_00\4&22394B12&0&09F8
I/O Port 0x0000F800-0x0000F8FF
Memory Address 0xED9F0000-0xED9FFFFF
Memory Address 0xED9E0000-0xED9EFFFF
IRQ Channel IRQ 169
Driver c:\windows\system32\drivers\symmpi.sys (1.09.15.00 built by: WinDDK, 109,50 KB (112 128 bytes), 29/04/2004 10:28)

[IDE]

Item Value
Name Intel(r) 82801DB Ultra ATA Storage Controller-24CB
Manufacturer Intel
Status OK
PNP Device ID PCI\VEN_8086&DEV_24CB&SUBSYS_0000119F&REV_01\3&267A616A&0&F9
I/O Port 0x00001000-0x0000100F
Memory Address 0xFDFFFC00-0xFDFFFFFF
Driver c:\windows\system32\drivers\intelide.sys (5.2.3790.0 (srv03_rtm.030324-2048), 11,50 KB (11 776 bytes), 09/04/2004 10:32)

Name Primary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCIIDE\IDECHANNEL\4&2ECBF3E8&0&0
I/O Port 0x000001F0-0x000001F7
I/O Port 0x000003F6-0x000003F6
IRQ Channel IRQ 14
Driver c:\windows\system32\drivers\atapi.sys (5.2.3790.1159 (dnsrv.040209-1620), 281,50 KB (288 256 bytes), 11/02/2004 13:00)

Name Secondary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK

PNP Device ID PCIIDE\IDECHANNEL\4&2ECBF3E8&0&1
I/O Port 0x00000170-0x00000177
I/O Port 0x00000376-0x00000376
IRQ Channel IRQ 15
Driver c:\windows\system32\drivers\atapi.sys (5.2.3790.1159 (dnsrv.040209-1620), 281,50 KB (288 256 bytes), 11/02/2004 13:00)

[Printing]

Name Driver Port Name Server Name

[Problem Devices]

Device PNP Device ID Error Code
Other PCI Bridge Device PCI\VEN_8086&DEV_B555&SUBSYS_B555119F&REV_02\4&2AF9ED5&0&08F0 This device is disabled.

[USB]

Device PNP Device ID
Intel(r) 82801DB/DBM USB Universal Host Controller - 24C2 PCI\VEN_8086&DEV_24C2&SUBSYS_0000119F&REV_01\3&267A616A&0&E8
USB Root Hub USB\ROOT_HUB\4&A453B83&0
USB Composite Device USB\VID_0D3D&PID_0001\5&26EC6CB&0&1
USB Human Interface Device USB\VID_0D3D&PID_0001&MI_00\6&1E2EBC7E&0&0000
HID Keyboard Device HID\VID_0D3D&PID_0001&MI_00\7&8D32FF0&0&0000
USB Human Interface Device USB\VID_0D3D&PID_0001&MI_01\6&1E2EBC7E&0&0001
HID-compliant mouse HID\VID_0D3D&PID_0001&MI_01&COL01\7&2CAA6DB2&0&0000
HID-compliant consumer control device HID\VID_0D3D&PID_0001&MI_01&COL02\7&2CAA6DB2&0&0001
HID-compliant device HID\VID_0D3D&PID_0001&MI_01&COL03\7&2CAA6DB2&0&0002
Intel(r) 82801DB/DBM USB Universal Host Controller - 24C4 PCI\VEN_8086&DEV_24C4&SUBSYS_0000119F&REV_01\3&267A616A&0&E9
USB Root Hub USB\ROOT_HUB\4&19DC9BE3&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	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No		
adpu320	adpu320	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No		
afcnt	afcnt	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No		
afd	AFD Networking Support Environment	c:\windows\system32\drivers\afd.sys	Kernel Driver	Yes	Auto	Running	OK	Normal	No	Yes		
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		
e100b	Intel(R) PRO Adapter Driver	c:\windows\system32\drivers\e100b645.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No	Yes		
fastfat	Fastfat	c:\windows\system32\drivers\fastfat.sys	File System Driver	Yes	Disabled	Running	OK	Normal	No	Yes		
fdc	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		
fltmgr	FltMgr	c:\windows\system32\drivers\fltmgr.sys	File System Driver	Yes	Boot	Running	OK	Normal	No	Yes		
ftdisk	Volume Manager Driver	c:\windows\system32\drivers\ftdisk.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No	Yes		
gpc	Generic Packet Classifier	c:\windows\system32\drivers\msgpc.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No	Yes		
hidusb	Microsoft HID Class Driver	c:\windows\system32\drivers\hidusb.sys	Kernel Driver	Yes	Manual	Running	OK	Ignore	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		
imapi	CD-Burning Filter Driver	c:\windows\system32\drivers\imapi.sys	Kernel Driver	No	System	Stopped	OK	Normal	No	No		
intelide	Intellde	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		
kbdhid	Keyboard HID Driver	c:\windows\system32\drivers\kbdhid.sys	Kernel Driver	Yes	System	Running	OK	Ignore	No	Yes		
ksecdd	KSecDD	c:\windows\system32\drivers\ksecdd.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No	Yes		
ksthunk	Kernel Streaming WOW64 Thunk Service	c:\windows\system32\drivers\ksthunk.sys	Kernel Driver	Yes	Manual	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		
mouhid	Mouse HID Driver	c:\windows\system32\drivers\mouhid.sys	Kernel Driver	Yes	Manual	Running	OK	Ignore	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		
mssmbios	Microsoft System Management BIOS Driver	c:\windows\system32\drivers\mssmbios.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No	Yes		
mup	Mup	c:\windows\system32\drivers\mup.sys	File System Driver	Yes	Boot	Running	OK	Normal	No	Yes		
ndis	NDIS System Driver	c:\windows\system32\drivers\ndis.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No	Yes		
ndistapi	Remote Access NDIS TAPI Driver	c:\windows\system32\drivers\ndistapi.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No	Yes		
ndisuio	NDIS Usermode I/O Protocol	c:\windows\system32\drivers\ndisuio.sys	Kernel Driver	No	Manual	Stopped	OK	Normal	No	No		
ndiswan	Remote Access NDIS WAN Driver	c:\windows\system32\drivers\ndiswan.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No	Yes		
ndproxy	NDIS Proxy	c:\windows\system32\drivers\ndproxy.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No	Yes		
netbios	NetBIOS Interface	c:\windows\system32\drivers\netbios.sys	File System Driver	Yes	System	Running	OK	Normal	No	Yes		
netbt	NetBios over Tcpip	c:\windows\system32\drivers\netbt.sys	Kernel Driver	Yes	System	Running	OK	Normal	No	Yes		
nfrd960	nfrd960	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No		
npfs	Npfs	c:\windows\system32\drivers\npfs.sys	File System Driver	Yes	System	Running	OK	Normal	No	Yes		

ntfs	Ntfs	c:\windows\system32\drivers\ntfs.sys	File System Driver	Yes	Disabled	Running	OK	Normal	No	Yes	
null	Null	c:\windows\system32\drivers\null.sys	Kernel Driver	Yes	System	Running	OK	Normal	No	Yes	
partmgr	Partition Manager	c:\windows\system32\drivers\partmgr.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No	Yes	
pci	PCI Bus Driver	c:\windows\system32\drivers\pci.sys	Kernel Driver	Yes	Boot	Running	OK	Critical	No	Yes	
pciide	PCIIde	c:\windows\system32\drivers\pciide.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No	Yes	
pcmcia	Pcmcia	c:\windows\system32\drivers\pcmcia.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No	
pdcomp	PDCOMP	Not Available	Kernel Driver	No	Manual	Stopped	OK	Ignore	No	No	
pdframe	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	
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	
ql2200	ql2200	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No	
ql2300	ql2300	c:\windows\system32\drivers\ql2350.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No	Yes	
ql2340	ql2340	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	Auto	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	High-Capacity Floppy Disk Drive	c:\windows\system32\drivers\sfloppy.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No	Yes	
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	c:\windows\system32\drivers\symmpi.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No	Yes	
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	Yes	System	Running	OK	Ignore	No	Yes	
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	
usbccgp	Microsoft USB Generic Parent Driver	c:\windows\system32\drivers\usbccgp.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No	Yes	
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	Vialde	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	HTREE\ROOT\0			
ACPI IA64-based PC	No	COMPUTER	5.2.3790.1159	10/01/2002	(Standard computers)		hal.inf	Not Available	ROOT\ACPI_HAL\0000	
Microsoft ACPI-Compliant System	No	SYSTEM	5.2.3790.0	10/01/2002	Microsoft	acpi.inf	Not Available	ACPI_HAL\PNP0C08\0		
Processor	No	PROCESSOR	5.2.3790.1159	10/01/2002	(Standard processor types)		cpu.inf	Not Available	ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_1_0	
Processor	No	PROCESSOR	5.2.3790.1159	10/01/2002	(Standard processor types)		cpu.inf	Not Available	ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_1_1	
Processor	No	PROCESSOR	5.2.3790.1159	10/01/2002	(Standard processor types)		cpu.inf	Not Available	ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_1_4	
Processor	No	PROCESSOR	5.2.3790.1159	10/01/2002	(Standard processor types)		cpu.inf	Not Available	ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_1_5	
Processor	No	PROCESSOR	5.2.3790.1159	10/01/2002	(Standard processor types)		cpu.inf	Not Available	ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_1_8	
Processor	No	PROCESSOR	5.2.3790.1159	10/01/2002	(Standard processor types)		cpu.inf	Not Available	ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_1_9	
Processor	No	PROCESSOR	5.2.3790.1159	10/01/2002	(Standard processor types)		cpu.inf	Not Available	ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_1\12	
Processor	No	PROCESSOR	5.2.3790.1159	10/01/2002	(Standard processor types)		cpu.inf	Not Available	ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_1\13	
ACPI Thermal Zone	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)		machine.inf	Not Available	ACPI\THERMALZONE\TZ0_	
PCI bus	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)		machine.inf	Not Available	ACPI\PNP0A03\0	
Intel(r) 82801DB/DBM USB Universal Host Controller - 24C2				No	USB	5.2.3790.0	10/01/2002	Intel	usbport.inf	Not Available
PCI\VEN_8086&DEV_24C2&SUBSYS_0000119F&REV_01\3&267A616A&0&E8										
USB Root Hub	No	USB	5.2.3790.0	10/01/2002	(Standard USB Host Controller)		usbport.inf	Not Available	USB\ROOT_HUB\4&A453B83&0	
USB Composite Device		No	USB	5.2.3790.0	10/01/2002	(Standard USB Host Controller)		usb.inf	Not Available	USB\VID_0D3D&PID_0001\5&26EC6CB&0&1
USB Human Interface Device		No	HIDCLASS	5.2.3790.0	10/01/2002	(Standard system devices)		input.inf	Not Available	
USB\VID_0D3D&PID_0001&MI_00\6&1E2EBC7E&0&0000										
HID Keyboard Device		No	KEYBOARD	5.2.3790.0	10/01/2002	(Standard keyboards)		keyboard.inf	Not Available	
HID\VID_0D3D&PID_0001&MI_00\7&8D32FF0&0&0000										
USB Human Interface Device		No	HIDCLASS	5.2.3790.0	10/01/2002	(Standard system devices)		input.inf	Not Available	
USB\VID_0D3D&PID_0001&MI_01\6&1E2EBC7E&0&0001										
HID-compliant mouse		No	MOUSE	5.2.3790.0	10/01/2002	Microsoft	msmouse.inf	Not Available	HID\VID_0D3D&PID_0001&MI_01&COL01\7&2CAA6DB2&0&0000	
HID-compliant consumer control device		No	HIDCLASS	5.2.3790.0	10/01/2002	Microsoft	hidserv.inf	Not Available	HID\VID_0D3D&PID_0001&MI_01&COL02\7&2CAA6DB2&0&0001	
HID-compliant device		No	HIDCLASS	5.2.3790.0	10/01/2002	(Standard system devices)		input.inf	Not Available	
HID\VID_0D3D&PID_0001&MI_01&COL03\7&2CAA6DB2&0&0002										
Intel(r) 82801DB/DBM USB Universal Host Controller - 24C4				No	USB	5.2.3790.0	10/01/2002	Intel	usbport.inf	Not Available
PCI\VEN_8086&DEV_24C4&SUBSYS_0000119F&REV_01\3&267A616A&0&E9										
USB Root Hub	No	USB	5.2.3790.0	10/01/2002	(Standard USB Host Controller)		usbport.inf	Not Available	USB\ROOT_HUB\4&19DC9BE3&0	
Intel(R) 82801DB PCI Bridge - 244E	No	SYSTEM	5.2.3790.1159	10/01/2002	Intel	machine.inf	Not Available			
PCI\VEN_8086&DEV_244E&SUBSYS_00000000&REV_81\3&267A616A&0&F0										
Standard VGA Graphics Adapter	No	DISPLAY	5.2.3790.0	10/01/2002	(Standard display types)		display.inf	Not Available		
PCI\VEN_1002&DEV_4752&SUBSYS_80081002&REV_65\4&2AF9ED5&0&00F0										
Plug and Play Monitor		No	MONITOR	5.1.2001.0	06/06/2001	(Standard monitor types)		monitor.inf	Not Available	DISPLAY\ZDS1750\5&24A77C58&0&12345678&01&00
Other PCI Bridge Device	Not Available	UNKNOWN	Not Available	Not Available	Not Available	Not Available	Not Available			
PCI\VEN_8086&DEV_B555&SUBSYS_B555119F&REV_02\4&2AF9ED5&0&08F0										
Intel 8255x-based PCI Ethernet Adapter (10/100)		No	NET	6.6.8.1	10/01/2002	Intel	net557.inf	Not Available		
PCI\VEN_8086&DEV_1229&SUBSYS_0000119F&REV_0D\4&2AF9ED5&0&10F0										
Intel(r) 82801DB LPC Interface Controller - 24C0		No	SYSTEM	5.2.3790.1159	10/01/2002	Intel	machine.inf	Not Available		
PCI\VEN_8086&DEV_24C0&SUBSYS_00000000&REV_01\3&267A616A&0&F8										
Intel(r) 82801DB Ultra ATA Storage Controller-24CB		No	HDC	5.2.3790.1159	10/01/2002	Intel	mshdc.inf	Not Available		
PCI\VEN_8086&DEV_24CB&SUBSYS_0000119F&REV_01\3&267A616A&0&F9										
Primary IDE Channel		No	HDC	5.2.3790.1159	10/01/2002	(Standard IDE ATA/ATAPI controllers)		mshdc.inf	Not Available	PCIIDE\IDECHANNEL\4&2ECBF3E8&0&0
High-Capacity Floppy Disk Drive		No	FLOPPYDISK	5.2.3790.0	10/01/2002	(Standard floppy disk drives)		flpydisk.inf	Not Available	IDE\DISKMATSHITA_LS-120/240____00_____H050___\59323830484D3041313036322020202020202020
Secondary IDE Channel		No	HDC	5.2.3790.1159	10/01/2002	(Standard IDE ATA/ATAPI controllers)		mshdc.inf	Not Available	PCIIDE\IDECHANNEL\4&2ECBF3E8&0&1
CD-ROM Drive	No	CDROM	5.2.3790.0	10/01/2002	(Standard CD-ROM drives)		cdrom.inf	Not Available	IDE\CDROMMATSHITA_DVD-ROM_SR-8177_____NZ20____\5&11538027&0&0.0.0	
Intel(r) 82801DB SMBus Controller - 24C3				No	SYSTEM	5.2.3790.1159	10/01/2002	Intel	machine.inf	Not Available
PCI\VEN_8086&DEV_24C3&SUBSYS_0000119F&REV_01\3&267A616A&0&FB										
Generic Bus	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)		machine.inf	Not Available	ACPI\PNP0A05\3&267A616A&0	
Motherboard resources		No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)		machine.inf	Not Available	ACPI\PNP0C02\1
Programmable interrupt controller		No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)		machine.inf	Not Available	ACPI\PNP0000\4&1B47357E&0
Direct memory access controller		No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)		machine.inf	Not Available	ACPI\PNP0200\4&1B47357E&0
System timer	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)		machine.inf	Not Available	ACPI\PNP0100\4&1B47357E&0	

System CMOS/real time clock No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0B00\4&1B47357E&0
System speaker No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0800\4&1B47357E&0
Motherboard resources No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0C02\4
Communications Port No PORTS 5.2.3790.0 10/01/2002 (Standard port types) msports.inf Not Available ACPI\PNP0501\1
PCI bus No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0A03\1
Intel(r) 82870 I/OxAPIC Interrupt Controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1461&SUBSYS_00000000&REV_04\3&13C0B0C5&0&E0
Intel(r) 82870 Hub Interface to PCI Bridges No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1460&SUBSYS_00000000&REV_04\3&13C0B0C5&0&E8
QLogic QLA23xx PCI Fibre Channel Adapter No SCSIADAPTER 8.2.2.10 16/04/2003 QLogic oem2.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&207C21BC&0&08E8
Qlogic processor device No SYSTEM 5.2.3790.0 10/01/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\5&21326B4F&0&07F0
Intel(r) 82870 Hot Plug controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1462&SUBSYS_03F08086&REV_04\4&207C21BC&0&F8E8
Intel(r) 82870 I/OxAPIC Interrupt Controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1461&SUBSYS_00000000&REV_04\3&13C0B0C5&0&F0
Intel(r) 82870 Hub Interface to PCI Bridges No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1460&SUBSYS_00000000&REV_04\3&13C0B0C5&0&F8
QLogic QLA23xx PCI Fibre Channel Adapter No SCSIADAPTER 8.2.2.10 16/04/2003 QLogic oem2.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&2AAD566B&0&08F8
Qlogic processor device No SYSTEM 5.2.3790.0 10/01/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\5&BF12E7B&0&07F0
Intel(r) 82870 Hot Plug controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1462&SUBSYS_03F08086&REV_04\4&2AAD566B&0&F8F8
PCI bus No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0A03\2
Intel(r) 82870 I/OxAPIC Interrupt Controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1461&SUBSYS_00000000&REV_04\3&1070020&0&E0
Intel(r) 82870 Hub Interface to PCI Bridges No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1460&SUBSYS_00000000&REV_04\3&1070020&0&E8
Intel(r) 82870 Hot Plug controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1462&SUBSYS_03B08086&REV_04\4&226B0D81&0&F8E8
Intel(r) 82870 I/OxAPIC Interrupt Controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1461&SUBSYS_00000000&REV_04\3&1070020&0&F0
Intel(r) 82870 Hub Interface to PCI Bridges No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1460&SUBSYS_00000000&REV_04\3&1070020&0&F8
QLogic QLA23xx PCI Fibre Channel Adapter No SCSIADAPTER 8.2.2.10 16/04/2003 QLogic oem2.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&28BE6AA6&0&08F8
Qlogic processor device No SYSTEM 5.2.3790.0 10/01/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\5&308FF895&0&07F0
Intel(r) 82870 Hot Plug controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1462&SUBSYS_03F08086&REV_04\4&28BE6AA6&0&F8F8
PCI bus No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0A03\3
Intel(r) 82870 I/OxAPIC Interrupt Controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1461&SUBSYS_00000000&REV_04\3&29E81982&0&E0
Intel(r) 82870 Hub Interface to PCI Bridges No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1460&SUBSYS_00000000&REV_04\3&29E81982&0&E8
Intel(r) 82870 Hot Plug controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1462&SUBSYS_03B08086&REV_04\4&1EF53336&0&F8E8
Intel(r) 82870 I/OxAPIC Interrupt Controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1461&SUBSYS_00000000&REV_04\3&29E81982&0&F0
Intel(r) 82870 Hub Interface to PCI Bridges No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1460&SUBSYS_00000000&REV_04\3&29E81982&0&F8
QLogic QLA23xx PCI Fibre Channel Adapter No SCSIADAPTER 8.2.2.10 16/04/2003 QLogic oem2.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\4&16D68E&0&08F8
Qlogic processor device No SYSTEM 5.2.3790.0 10/01/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\5&1F78F1BA&0&07F0
Intel(r) 82870 Hot Plug controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1462&SUBSYS_03F08086&REV_04\4&16D68E&0&F8F8
PCI bus No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0A03\4
Intel(r) 82870 I/OxAPIC Interrupt Controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1461&SUBSYS_00000000&REV_04\3&172E68DD&0&E0
Intel(r) 82870 Hub Interface to PCI Bridges No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1460&SUBSYS_00000000&REV_04\3&172E68DD&0&E8
LSI Adapter, FC 7002 series, Dual or Quad port (with 929X) No SCSIADAPTER 1.9.15.0 19/02/2004 LSI Logic oem4.inf Not Available
PCI\VEN_1000&DEV_0626&SUBSYS_10101000&REV_00\4&1B67A8B9&0&08E8
LSI Pseudo Device No SYSTEM 6.2.0.0 14/11/2002 LSI Logic oem5.inf Not Available SCSI\BRIDGE&VEN_LSI&PROD_PSEUDO_DEVICE&REV_2.34\5&18C2B144&0&0F0

LSI Adapter, FC 7002 series, Dual or Quad port (with 929X) No SCSIADAPTER 1.9.15.0 19/02/2004 LSI Logic oem4.inf Not Available
PCI\VEN_1000&DEV_0626&SUBSYS_10101000&REV_00\4&1B67A8B9&0&09E8
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&315BD8EE&0&001
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&315BD8EE&0&010
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&315BD8EE&0&021
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&315BD8EE&0&030
LSI Pseudo Device No SYSTEM 6.2.0.0 14/11/2002 LSI Logic oem5.inf Not Available SCSI\BRIDGE&VEN_LSI&PROD_PSEUDO_DEVICE&REV_2.34\5&315BD8EE&0&0F0

QLogic QLA234x PCI Fibre Channel Adapter No SCSIADAPTER 8.2.3.11 25/07/2003 QLogic oem1.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_01001077&REV_02\4&1B67A8B9&0&10E8
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&2C6F8152&0&000
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&2C6F8152&0&001
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&2C6F8152&0&002
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&2C6F8152&0&003
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&2C6F8152&0&004
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&2C6F8152&0&005
Qlogic processor device No SYSTEM 5.2.3790.0 10/01/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\5&2C6F8152&0&07F0
Intel(r) 82870 Hot Plug controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1462&SUBSYS_03B08086&REV_04\4&1B67A8B9&0&F8E8
Intel(r) 82870 I/OxAPIC Interrupt Controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1461&SUBSYS_00000000&REV_04\3&172E68DD&0&F0
Intel(r) 82870 Hub Interface to PCI Bridges No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1460&SUBSYS_00000000&REV_04\3&172E68DD&0&F8
LSI Adapter, FC 7002 series, Dual or Quad port (with 929X) No SCSIADAPTER 1.9.15.0 19/02/2004 LSI Logic oem4.inf Not Available
PCI\VEN_1000&DEV_0626&SUBSYS_10101000&REV_00\4&22394B12&0&08F8
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&5AC7C05&0&001
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&5AC7C05&0&010
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&5AC7C05&0&021
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&5AC7C05&0&030
LSI Pseudo Device No SYSTEM 6.2.0.0 14/11/2002 LSI Logic oem5.inf Not Available SCSI\BRIDGE&VEN_LSI&PROD_PSEUDO_DEVICE&REV_2.34\5&5AC7C05&0&0F0

LSI Adapter, FC 7002 series, Dual or Quad port (with 929X) No SCSIADAPTER 1.9.15.0 19/02/2004 LSI Logic oem4.inf Not Available
PCI\VEN_1000&DEV_0626&SUBSYS_10101000&REV_00\4&22394B12&0&09F8
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&24818389&0&001
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&24818389&0&010
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&24818389&0&021
Disk drive No DISKDRIVE 5.2.3790.0 10/01/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_NEC&PROD_ISTORAGE_2000&REV_2300\5&24818389&0&030
LSI Pseudo Device No SYSTEM 6.2.0.0 14/11/2002 LSI Logic oem5.inf Not Available SCSI\BRIDGE&VEN_LSI&PROD_PSEUDO_DEVICE&REV_2.34\5&24818389&0&0F0

Intel(r) 82870 Hot Plug controller No SYSTEM 5.2.3790.1159 10/01/2002 Intel machine.inf Not Available
PCI\VEN_8086&DEV_1462&SUBSYS_03F08086&REV_04\4&22394B12&0&F8F8
PCI bus No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0A03\FF
PCI standard host CPU bridge No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0500&SUBSYS_00000000&REV_20\3&4F11E618&0&C0
PCI standard host CPU bridge No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0501&SUBSYS_00000000&REV_20\3&4F11E618&0&C1
PCI standard host CPU bridge No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0502&SUBSYS_00000000&REV_20\3&4F11E618&0&C2
PCI standard host CPU bridge No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0503&SUBSYS_00000000&REV_20\3&4F11E618&0&C3
PCI standard host CPU bridge No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0500&SUBSYS_00000000&REV_20\3&4F11E618&0&C8
PCI standard host CPU bridge No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0501&SUBSYS_00000000&REV_20\3&4F11E618&0&C9
PCI standard host CPU bridge No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0502&SUBSYS_00000000&REV_20\3&4F11E618&0&CA
PCI standard host CPU bridge No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0503&SUBSYS_00000000&REV_20\3&4F11E618&0&CB
PCI standard host CPU bridge No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0500&SUBSYS_00000000&REV_21\3&4F11E618&0&D0
PCI standard host CPU bridge No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0501&SUBSYS_00000000&REV_21\3&4F11E618&0&D1
PCI standard host CPU bridge No SYSTEM 5.2.3790.1159 10/01/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0502&SUBSYS_00000000&REV_21\3&4F11E618&0&D2

PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0503&SUBSYS_00000000&REV_21\3&4F11E61&0&D3									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0500&SUBSYS_00000000&REV_20\3&4F11E61&0&D8									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0501&SUBSYS_00000000&REV_20\3&4F11E61&0&D9									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0502&SUBSYS_00000000&REV_20\3&4F11E61&0&DA									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0503&SUBSYS_00000000&REV_20\3&4F11E61&0&DB									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0510&SUBSYS_00000000&REV_21\3&4F11E61&0&E0									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0511&SUBSYS_00000000&REV_21\3&4F11E61&0&E1									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0512&SUBSYS_00000000&REV_21\3&4F11E61&0&E2									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0513&SUBSYS_00000000&REV_21\3&4F11E61&0&E3									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0514&SUBSYS_00000000&REV_21\3&4F11E61&0&E4									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0515&SUBSYS_00000000&REV_21\3&4F11E61&0&E5									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0516&SUBSYS_00000000&REV_21\3&4F11E61&0&E6									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0530&SUBSYS_00000000&REV_10\3&4F11E61&0&E8									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0531&SUBSYS_00000000&REV_10\3&4F11E61&0&E9									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0532&SUBSYS_00000000&REV_10\3&4F11E61&0&EA									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0533&SUBSYS_00000000&REV_10\3&4F11E61&0&EB									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0534&SUBSYS_00000000&REV_10\3&4F11E61&0&EC									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0535&SUBSYS_00000000&REV_10\3&4F11E61&0&ED									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0536&SUBSYS_00000000&REV_10\3&4F11E61&0&EE									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0537&SUBSYS_00000000&REV_10\3&4F11E61&0&EF									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0530&SUBSYS_00000000&REV_10\3&4F11E61&0&F8									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0531&SUBSYS_00000000&REV_10\3&4F11E61&0&F9									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0532&SUBSYS_00000000&REV_10\3&4F11E61&0&FA									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0533&SUBSYS_00000000&REV_10\3&4F11E61&0&FB									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0534&SUBSYS_00000000&REV_10\3&4F11E61&0&FC									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0535&SUBSYS_00000000&REV_10\3&4F11E61&0&FD									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0536&SUBSYS_00000000&REV_10\3&4F11E61&0&FE									
PCI standard host CPU bridge	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available		
PCI\VEN_8086&DEV_0537&SUBSYS_00000000&REV_10\3&4F11E61&0&FF									
ACPI Fixed Feature Button	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available	ACPI\FIXEDBUTTON\2&DABA3FF&0	
Logical Disk Manager	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available	ROOT\DMIO\0000	
Volume Manager	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)	machine.inf	Not Available	ROOT\FTDISK\0000	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC78467DBOFFSET7E00LENGTHB404DE800									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC78467DBOFFSETB404E6600LENGTH938229C00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC78467DBOFFSET1478710200LENGTH528488A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC78467DBOFFSET2BB4564800LENGTH3E80487A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATUREC78467DBOFFSET6A349F4000LENGTH129F557A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATUREDF38FA58OFFSET7E00LENGTHB404DE800									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATUREDF38FA58OFFSETB404E6600LENGTH938229C00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATUREDF38FA58OFFSET1478710200LENGTH528488A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATUREDF38FA58OFFSET2BB4564800LENGTH3E80487A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATUREDF38FA58OFFSET6A349F4000LENGTH129F557A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE6E0F2C57OFFSET7E00LENGTHB404DE800									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE6E0F2C57OFFSETB404E6600LENGTH938229C00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE6E0F2C57OFFSET1478710200LENGTH528488A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE6E0F2C57OFFSET2BB4564800LENGTH3E80487A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE6E0F2C57OFFSET6A349F4000LENGTH129F557A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE6E0F2C56OFFSET7E00LENGTHB404DE800									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE6E0F2C56OFFSETB404E6600LENGTH938229C00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE6E0F2C56OFFSET1478710200LENGTH528488A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE6E0F2C56OFFSET2BB4564800LENGTH3E80487A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE6E0F2C56OFFSET6A349F4000LENGTH129F557A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{713FE541-7CB4-11D8-8FB1-505054503030}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{713FE542-7CB4-11D8-8FB1-505054503030}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2E41C2E0-1CB0-01C4-A1F4-04622FD5EC6D}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{03946B8E-8CE9-4C7B-A8D9-27095F91934C}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{307787D8-21EA-4D44-96CD-A2F93E9B7AEA}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0F2F5EB8-AA55-494E-AD10-DA733F850E05}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{27D916C3-88E6-43AB-93FC-994ECDB95229}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2C2528E2-20A0-41E2-A21E-E4E516FB8A3D}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2DB01F24-B020-4A49-A1AE-A5D37E2F1A8E}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7323A2EF-F6A0-4DB1-9AC3-0C0C43C0C392}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4C69723B-9CCF-4E6C-A3E4-AFD9A3AD0DDA}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F16A4FC1-E761-4109-869C-6725CFAB9DEB}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3CAB3710-F245-4BAD-AFAA-6079F7A94B85}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EBDE7CBF-07A5-4941-A3DD-91421A956210}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CB9FAC23-631D-4FAD-9CE1-97E952FE8CD0}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FA2DDC73-7D9D-4235-AC3A-7E57FA6CDDE4}	
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE7FECADOFFSET7E00LENGTHB404DE800									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE7FECADOFFSETB404E6600LENGTH938229C00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE7FECADOFFSET1478710200LENGTH528488A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE7FECADOFFSET2BB4564800LENGTH3E80487A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE7FECADOFFSET6A349F4000LENGTH129F557A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE4651CFC1OFFSET7E00LENGTHB404DE800									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE4651CFC1OFFSETB404E6600LENGTH938229C00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE4651CFC1OFFSET1478710200LENGTH528488A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE4651CFC1OFFSET2BB4564800LENGTH3E80487A00									
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available		
STORAGE\VOLUME\1&30A96598&0&SIGNATURE4651CFC1OFFSET6A349F4000LENGTH129F557A00									

Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURE480E5D6FOFFSET7E00LENGTHB404DE800
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURE480E5D6FOFFSETB404E6600LENGTH938229C00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURE480E5D6FOFFSET1478710200LENGTH528488A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURE480E5D6FOFFSET2BB4564800LENGTH3E80487A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURE480E5D6FOFFSET6A349F4000LENGTH129F557A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURE7FECAEOFFSET7E00LENGTHB404DE800
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURE7FECAEOFFSETB404E6600LENGTH938229C00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURE7FECAEOFFSET1478710200LENGTH528488A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURE7FECAEOFFSET2BB4564800LENGTH3E80487A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURE7FECAEOFFSET6A349F4000LENGTH129F557A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURED77E2788OFFSET7E00LENGTHB404DE800
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURED77E2788OFFSETB404E6600LENGTH938229C00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURED77E2788OFFSET1478710200LENGTH528488A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURED77E2788OFFSET2BB4564800LENGTH3E80487A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURED77E2788OFFSET6A349F4000LENGTH129F557A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURED77E2789OFFSET7E00LENGTHB404DE800
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURED77E2789OFFSETB404E6600LENGTH938229C00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURED77E2789OFFSET1478710200LENGTH528488A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURED77E2789OFFSET2BB4564800LENGTH3E80487A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATURED77E2789OFFSET6A349F4000LENGTH129F557A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATUREDC58E4E5OFFSET7E00LENGTHB404DE800
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATUREDC58E4E5OFFSETB404E6600LENGTH938229C00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATUREDC58E4E5OFFSET1478710200LENGTH528488A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATUREDC58E4E5OFFSET2BB4564800LENGTH3E80487A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATUREDC58E4E5OFFSET6A349F4000LENGTH129F557A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATUREDC58E4E7OFFSET7E00LENGTHB404DE800
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATUREDC58E4E7OFFSETB404E6600LENGTH938229C00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATUREDC58E4E7OFFSET1478710200LENGTH528488A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATUREDC58E4E7OFFSET2BB4564800LENGTH3E80487A00
Generic volume	No	VOLUME	5.2.3790.0	10/01/2002	Microsoft	volume.inf	Not Available	STORAGE\VOLUME\1&30A96598&0&SIGNATUREDC58E4E7OFFSET6A349F4000LENGTH129F557A00

AFD Networking Support Environment	Not Available	LEGACYDRIVER	Not Available	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	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	Not Available	ROOT\LEGACY_DMBOOT\0000
dmload	Not Available	LEGACYDRIVER	Not Available	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	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	Not Available	ROOT\LEGACY_IPSEC\0000
ksecdd	Not Available	LEGACYDRIVER	Not Available	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	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	Not Available	ROOT\LEGACY_NDPROXY\0000
NetBios over Tcpip	Not Available	LEGACYDRIVER	Not Available	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	Not Available	ROOT\LEGACY_NULL\0000
Partition Manager	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	ROOT\LEGACY_PARTMGR\0000
PCIIde	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	ROOT\LEGACY_PCIIDE\0000
qldirect	Not Available	LEGACYDRIVER	Not Available	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	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	Not Available	ROOT\LEGACY_RDPCDD\0000
RDPWD	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	ROOT\LEGACY_RDPWD\0000
sacdrv	Not Available	LEGACYDRIVER	Not Available	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	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
TDTCP	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	ROOT\LEGACY_TDTCP\0000
volsnap	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available	ROOT\LEGACY_VOLSNAP\0000
Remote Access IP ARP Driver		Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available	Not Available	Not Available	ROOT\LEGACY_WANARP\0000
Audio Codecs	No	MEDIA	5.2.3790.0	10/01/2002	(Standard system devices)		wave.inf	Not Available	ROOT\MEDIA\MS_MMACM
Legacy Audio Drivers		No	MEDIA	5.2.3790.0	10/01/2002	(Standard system devices)		wave.inf	Not Available	ROOT\MEDIA\MS_MMDRV
Media Control Devices		No	MEDIA	5.2.3790.0	10/01/2002	(Standard system devices)		wave.inf	Not Available	ROOT\MEDIA\MS_MMMCI
Legacy Video Capture Devices		No	MEDIA	5.2.3790.0	10/01/2002	(Standard system devices)		wave.inf	Not Available	ROOT\MEDIA\MS_MMVCD
Video Codecs	No	MEDIA	5.2.3790.0	10/01/2002	(Standard system devices)		wave.inf	Not Available	ROOT\MEDIA\MS_MMVID
WAN Miniport (L2TP)		No	NET	5.2.3790.1159	10/01/2002	Microsoft	netrasa.inf	Not Available	ROOT\MS_L2TPMINIPORT\0000
WAN Miniport (IP)	No	NET	5.2.3790.1159	10/01/2002	Microsoft	netrasa.inf	Not Available	ROOT\MS_NDISWANIP\0000	
WAN Miniport (PPPOE)		No	NET	5.2.3790.1159	10/01/2002	Microsoft	netrasa.inf	Not Available	ROOT\MS_PPPOEMINIPORT\0000
WAN Miniport (PPTP)		No	NET	5.2.3790.1159	10/01/2002	Microsoft	netrasa.inf	Not Available	ROOT\MS_PPTPMINIPORT\0000
Direct Parallel	No	NET	5.2.3790.1159	10/01/2002	Microsoft	netrasa.inf	Not Available	ROOT\MS_PTIMINIPORT\0000	
Terminal Server Device Redirector	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)		machine.inf	Not Available	ROOT\RDPDR\0000
Terminal Server Keyboard Driver	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)		machine.inf	Not Available	ROOT\RDP_KBD\0000
Terminal Server Mouse Driver	No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)		machine.inf	Not Available	ROOT\RDP_MOU\0000
Plug and Play Software Device Enumerator		No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)		machine.inf	Not Available	ROOT\SYSTEM\0000
Microsoft System Management BIOS Driver		No	SYSTEM	5.2.3790.1159	10/01/2002	(Standard system devices)		machine.inf	Not Available	ROOT\SYSTEM\0002

[Environment Variables]

Variable	Value	User Name	
ClusterLog	C:\WINDOWS\Cluster\cluster.log	<SYSTEM>	
ComSpec	%SystemRoot%\system32\cmd.exe	<SYSTEM>	
iSMrd	C:\Program Files (x86)\NEC\iSMsvr	<SYSTEM>	
NUMBER_OF_PROCESSORS	8	<SYSTEM>	
OS	Windows_NT	<SYSTEM>	
Path	C:\Program Files (x86)\Intel\VTune\CGGlbCache;C:\Program Files (x86)\Intel\VTune\Analyzer\Bin;C:\Program Files (x86)\Intel\VTune\Shared\Bin;C:\Program Files (x86)\Intel\VTune\Analyzer\Bin64;C:\Program Files (x86)\Intel\VTune\Shared\Bin64;%iSMrd%\sbin;%iSMrd%\lib;%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\Program Files\Microsoft SQL Server\80\Tools\Binn\;Z:\Outils\		<SYSTEM>
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH	<SYSTEM>	
PROCESSOR_ARCHITECTURE	IA64	<SYSTEM>	
PROCESSOR_IDENTIFIER	ia64 Family 31 Model 1 Stepping 5, GenuineIntel	<SYSTEM>	
PROCESSOR_LEVEL	31	<SYSTEM>	
PROCESSOR_REVISION	0105	<SYSTEM>	
TEMP	%SystemRoot%\TEMP	<SYSTEM>	
TMP	%SystemRoot%\TEMP	<SYSTEM>	
windir	%SystemRoot%	<SYSTEM>	
TEMP	%USERPROFILE%\Local Settings\Temp	NT AUTHORITY\SYSTEM	
TMP	%USERPROFILE%\Local Settings\Temp	NT AUTHORITY\SYSTEM	
TEMP	%USERPROFILE%\Local Settings\Temp	TPCC-FAME\Administrator	
TMP	%USERPROFILE%\Local Settings\Temp	TPCC-FAME\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	
Z:	\\199.182.101.200\kit		Disk	Persistent Connection	TPCC-FAME\Administrator

[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	Not Available	416	11	409600	2826240	09/06/2004 09:26	Not Available	Not Available	Not Available		
csrss.exe	Not Available	464	13	Not Available	Not Available	09/06/2004 09:26	Not Available	Not Available	Not Available		
winlogon.exe	c:\windows\system32\winlogon.exe	488	13	409600	2826240	09/06/2004 09:26	5.2.3790.1159 (dnsrv.040209-1620)	626,00 KB (641 024 bytes)		11/02/2004 13:00	
services.exe	c:\windows\system32\services.exe	532	9	409600	2826240	09/06/2004 09:26	5.2.3790.1159 (dnsrv.040209-1620)	294,00 KB (301 056 bytes)		11/02/2004 13:00	

lsass.exe	c:\windows\system32\lsass.exe	544	9	409600	2826240	09/06/2004 09:26	5.2.3790.0 (srv03_rtm.030324-2048)	15,00 KB (15 360 bytes)			11/02/2004 13:00
svchost.exe	c:\windows\system32\svchost.exe	752	8	409600	2826240	09/06/2004 09:26	5.2.3790.1159 (dnsrv.040209-1620)	35,00 KB (35 840 bytes)			11/02/2004 13:00
svchost.exe	c:\windows\system32\svchost.exe	792	8	409600	2826240	09/06/2004 09:26	5.2.3790.1159 (dnsrv.040209-1620)	35,00 KB (35 840 bytes)			11/02/2004 13:00
portmap.exe	c:\program files (x86)\qlogic corporation\sansurfer\portmap.exe			872	8	409600	2826240	09/06/2004 09:26	1.14.3	40,00 KB (40 960 bytes)	27/04/2004 09:47
mssearch.exe	c:\program files\common files\system\mssearch\bin\mssearch.exe			1060	8	409600	2826240	09/06/2004 09:26	9.107.8320.0	460,00 KB (471 040 bytes)	06/12/2002 19:07
explorer.exe	c:\windows\explorer.exe	1260	8	409600	2826240	09/06/2004 09:26	6.00.3790.1159 (dnsrv.040209-1620)	1,63 MB (1 709 568 bytes)			11/02/2004 13:00
ctfmon.exe	c:\windows\system32\ctfmon.exe	1328	8	409600	2826240	09/06/2004 09:26	5.2.3790.0 (srv03_rtm.030324-2048)	31,50 KB (32 256 bytes)			11/02/2004 13:00
ctfmon.exe	c:\windows\syswow64\ctfmon.exe	1348	8	409600	2826240	09/06/2004 09:26	5.2.3790.0 (srv03_rtm.030324-2048)	13,50 KB (13 824 bytes)			11/02/2004 13:00
sqlmangr.exe	c:\program files\microsoft sql server\80\tools\binn\sqlmangr.exe			1356	8	409600	2826240	09/06/2004 09:26	2000.080.0760.00	225,00 KB (230 400 bytes)	07/02/2003 05:48
wmiprvse.exe	Not Available	1444	8	Not Available	Not Available	09/06/2004 09:26	Not Available	Not Available	Not Available		
cmd.exe	c:\windows\system32\cmd.exe	1780	8	409600	2826240	09/06/2004 12:37	5.2.3790.1159 (dnsrv.040209-1620)	493,50 KB (505 344 bytes)			11/02/2004 13:00
calc.exe	c:\windows\system32\calc.exe	1160	8	409600	2826240	09/06/2004 15:00	5.2.3790.0 (srv03_rtm.030324-2048)	269,00 KB (275 456 bytes)			09/04/2004 17:40
notepad.exe	c:\windows\system32\notepad.exe	1120	8	409600	2826240	09/06/2004 15:03	5.2.3790.0 (srv03_rtm.030324-2048)	129,50 KB (132 608 bytes)			11/02/2004 13:00
helpctr.exe	c:\windows\pchealth\helpctr\binaries\helpctr.exe			1576	8	409600	2826240	09/06/2004 15:07	5.2.3790.1159 (dnsrv.040209-1620)	1,97 MB (2 068 992 bytes)	09/04/2004 17:42
helpsvc.exe	c:\windows\pchealth\helpctr\binaries\helpsvc.exe			1872	8	409600	2826240	09/06/2004 15:07	5.2.3790.1159 (dnsrv.040209-1620)	2,19 MB (2 295 296 bytes)	09/04/2004 17:42
wmiprvse.exe	Not Available	1096	8	Not Available	Not Available	09/06/2004 15:07	Not Available	Not Available	Not Available		

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
winlogon	5.2.3790.1159 (dnsrv.040209-1620)	626,00 KB (641 024 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\winlogon.exe
ntdll	5.2.3790.1159 (dnsrv.040209-1620)	1,48 MB (1 555 456 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\ntdll.dll
kernel32	5.2.3790.1159 (dnsrv.040209-1620)	1,78 MB (1 863 168 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\kernel32.dll
msvcrt	7.0.3790.1159 (dnsrv.040209-1620)	882,00 KB (903 168 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\msvcrt.dll
advapi32	5.2.3790.1159 (dnsrv.040209-1620)	1,33 MB (1 391 104 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\advapi32.dll
rpcrt4	5.2.3790.1159 (dnsrv.040209-1620)	2,06 MB (2 163 712 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\rpcrt4.dll
user32	5.2.3790.1159 (dnsrv.040209-1620)	1,31 MB (1 376 256 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\user32.dll
gdi32	5.2.3790.1159 (dnsrv.040209-1620)	783,50 KB (802 304 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\gdi32.dll
userenv	5.2.3790.1159 (dnsrv.040209-1620)	1,48 MB (1 551 360 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\userenv.dll
nddeapi	5.2.3790.0 (srv03_rtm.030324-2048)	39,50 KB (40 448 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\nddeapi.dll
crypt32	5.131.3790.1159 (dnsrv.040209-1620)	1,51 MB (1 579 008 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\crypt32.dll
msasn1	5.2.3790.1159 (dnsrv.040209-1620)	156,50 KB (160 256 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\msasn1.dll
secur32	5.2.3790.1159 (dnsrv.040209-1620)	169,00 KB (173 056 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\secur32.dll
winsta	5.2.3790.1159 (dnsrv.040209-1620)	141,50 KB (144 896 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\winsta.dll
netapi32	5.2.3790.1159 (dnsrv.040209-1620)	832,50 KB (852 480 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\netapi32.dll
profmap	5.2.3790.0 (srv03_rtm.030324-2048)	55,50 KB (56 832 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\profmap.dll
regapi	5.2.3790.1159 (dnsrv.040209-1620)	128,00 KB (131 072 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\regapi.dll
ws2_32	5.2.3790.1159 (dnsrv.040209-1620)	236,50 KB (242 176 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\ws2_32.dll
ws2help	5.2.3790.1159 (dnsrv.040209-1620)	50,00 KB (51 200 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\ws2help.dll
msgina	5.2.3790.1159 (dnsrv.040209-1620)	1,35 MB (1 420 288 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\msgina.dll
shsvcs	6.00.3790.1159 (dnsrv.040209-1620)	329,00 KB (336 896 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\shsvcs.dll
shlwapi	6.00.3790.1159 (dnsrv.040209-1620)	723,50 KB (740 864 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\shlwapi.dll
sfc	5.2.3790.0 (srv03_rtm.030324-2048)	7,50 KB (7 680 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\sfc.dll
sfc_os	5.2.3790.1159 (dnsrv.040209-1620)	257,00 KB (263 168 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\sfc_os.dll
wintrust	5.131.3790.1159 (dnsrv.040209-1620)	452,50 KB (463 360 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wintrust.dll
ole32	5.2.3790.1159 (dnsrv.040209-1620)	3,41 MB (3 571 200 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\ole32.dll
imagehlp	5.2.3790.0 (srv03_rtm.030324-2048)	128,50 KB (131 584 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\imagehlp.dll
comctl32	6.0 (dnsrv.040209-1620)	2,18 MB (2 287 104 bytes)	09/04/2004 10:30	Microsoft Corporation	c:\windows\winsxs\ia64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.101.0_x-ww_b3f03b2c\comctl32.dll
version	5.2.3790.0 (srv03_rtm.030324-2048)	44,00 KB (45 056 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\version.dll
winscard	5.2.3790.0 (srv03_rtm.030324-2048)	291,50 KB (298 496 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\winscard.dll
wtsapi32	5.2.3790.1159 (dnsrv.040209-1620)	48,00 KB (49 152 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wtsapi32.dll
sxs	5.2.3790.0 (srv03_rtm.030324-2048)	1,77 MB (1 860 608 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\sxs.dll
wldap32	5.2.3790.1159 (dnsrv.040209-1620)	410,00 KB (419 840 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wldap32.dll
cscdll	5.2.3790.1159 (dnsrv.040209-1620)	211,00 KB (216 064 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\cscdll.dll
wlnotify	5.2.3790.1159 (dnsrv.040209-1620)	227,50 KB (232 960 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wlnotify.dll
winmm	5.2.3790.1159 (dnsrv.040209-1620)	404,50 KB (414 208 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\winmm.dll
winspool	5.2.3790.1159 (dnsrv.040209-1620)	400,00 KB (409 600 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\winspool.drv
setupapi	5.2.3790.1159 (dnsrv.040209-1620)	1,90 MB (1 996 288 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\setupapi.dll
mpr	5.2.3790.0 (srv03_rtm.030324-2048)	163,00 KB (166 912 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\mpr.dll
shell32	6.00.3790.1159 (dnsrv.040209-1620)	12,36 MB (12 962 304 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\shell32.dll
oleaut32	5.2.3790.1159	3,57 MB (3 742 208 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\oleaut32.dll
comctl32	5.82 (dnsrv.040209-1620)	1,55 MB (1 624 064 bytes)	09/04/2004 10:30	Microsoft Corporation	c:\windows\winsxs\ia64_microsoft.windows.common-controls_6595b64144ccf1df_5.82.101.0_x-ww_dd1f50a7\comctl32.dll
uxtheme	6.00.3790.1159 (dnsrv.040209-1620)	544,00 KB (557 056 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\uxtheme.dll
samlib	5.2.3790.0 (srv03_rtm.030324-2048)	106,00 KB (108 544 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\samlib.dll
cscui	5.2.3790.1159 (dnsrv.040209-1620)	584,00 KB (598 016 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\cscui.dll
drprov	5.2.3790.1159 (dnsrv.040209-1620)	27,50 KB (28 160 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\drprov.dll
ntlanman	5.2.3790.0 (srv03_rtm.030324-2048)	108,00 KB (110 592 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\ntlanman.dll
netui0	5.2.3790.0 (srv03_rtm.030324-2048)	181,50 KB (185 856 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\netui0.dll
netui1	5.2.3790.0 (srv03_rtm.030324-2048)	482,00 KB (493 568 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\netui1.dll
davclnt	5.2.3790.0 (srv03_rtm.030324-2048)	59,00 KB (60 416 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\davclnt.dll
mprui	5.2.3790.0 (srv03_rtm.030324-2048)	97,00 KB (99 328 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\mprui.dll
netui2	5.2.3790.0 (srv03_rtm.030324-2048)	761,50 KB (779 776 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\netui2.dll
comdlg32	6.00.3790.1159 (dnsrv.040209-1620)	706,00 KB (722 944 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\comdlg32.dll
netmsg	5.2.3790.0 (srv03_rtm.030324-2048)	177,50 KB (181 760 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\netmsg.dll
clbcatq	2001.12.4720.1159 (dnsrv.040209-1620)	1,24 MB (1 296 384 bytes)	09/04/2004 17:40	Microsoft Corporation	c:\windows\system32\clbcatq.dll
comres	2001.12.4720.0 (srv03_rtm.030324-2048)	779,50 KB (798 208 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\comres.dll
ntmarta	5.2.3790.1159 (dnsrv.040209-1620)	344,00 KB (352 256 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\ntmarta.dll
wbemprox	5.2.3790.1159 (dnsrv.040209-1620)	50,00 KB (51 200 bytes)	09/04/2004 17:40	Microsoft Corporation	c:\windows\system32\wbem\wbemprox.dll
wbemcomn	5.2.3790.1159 (dnsrv.040209-1620)	603,00 KB (617 472 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wbem\wbemcomn.dll
wbemsvc	5.2.3790.0 (srv03_rtm.030324-2048)	62,50 KB (64 000 bytes)	09/04/2004 17:40	Microsoft Corporation	c:\windows\system32\wbem\wbemsvc.dll
fastprox	5.2.3790.1159 (dnsrv.040209-1620)	1,52 MB (1 593 344 bytes)	09/04/2004 17:40	Microsoft Corporation	c:\windows\system32\wbem\fastprox.dll
msvcp60	6.10.2240.8	941,50 KB (964 096 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\msvcp60.dll
ntdsapi	5.2.3790.0 (srv03_rtm.030324-2048)	181,50 KB (185 856 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\ntdsapi.dll
dnsapi	5.2.3790.1159 (dnsrv.040209-1620)	406,50 KB (416 256 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\dnsapi.dll
services	5.2.3790.1159 (dnsrv.040209-1620)	294,00 KB (301 056 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\services.exe
scesrv	5.2.3790.1159 (dnsrv.040209-1620)	766,50 KB (784 896 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\scesrv.dll
authz	5.2.3790.1159 (dnsrv.040209-1620)	204,00 KB (208 896 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\authz.dll
umpnpmgr	5.2.3790.0 (srv03_rtm.030324-2048)	314,50 KB (322 048 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\umpnpmgr.dll
ncobjapi	5.2.3790.1159 (dnsrv.040209-1620)	120,00 KB (122 880 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\ncobjapi.dll
eventlog	5.2.3790.1159 (dnsrv.040209-1620)	163,50 KB (167 424 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\eventlog.dll
psapi	5.2.3790.0 (srv03_rtm.030324-2048)	48,00 KB (49 152 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\psapi.dll
lsass	5.2.3790.0 (srv03_rtm.030324-2048)	15,00 KB (15 360 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\lsass.exe
lsasrv	5.2.3790.1159 (dnsrv.040209-1620)	1,99 MB (2 081 792 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\lsasrv.dll
samsrv	5.2.3790.1159 (dnsrv.040209-1620)	1,00 MB (1 049 600 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\samsrv.dll
cryptdll	5.2.3790.1159 (dnsrv.040209-1620)	61,00 KB (62 464 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\cryptdll.dll
msprivs	5.2.3790.0 (srv03_rtm.030324-2048)	46,00 KB (47 104 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\msprivs.dll
kerberos	5.2.3790.1159 (dnsrv.040209-1620)	893,00 KB (914 432 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\kerberos.dll
msv1_0	5.2.3790.1159 (dnsrv.040209-1620)	345,50 KB (353 792 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\msv1_0.dll
iphlpapi	5.2.3790.1159 (dnsrv.040209-1620)	223,50 KB (228 864 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\iphlpapi.dll
netlogon	5.2.3790.1159 (dnsrv.040209-1620)	936,50 KB (958 976 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\netlogon.dll
w32time	5.2.3790.1159 (dnsrv.040209-1620)	542,00 KB (555 008 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\w32time.dll
schannel	5.2.3790.1159 (dnsrv.040209-1620)	428,00 KB (438 272 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\schannel.dll
wdigest	5.2.3790.1159 (dnsrv.040209-1620)	186,00 KB (190 464 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wdigest.dll
rsaenh	5.2.3790.1159 (dnsrv.040209-1620)	379,98 KB (389 096 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\rsaenh.dll
rassfm	5.2.3790.1159 (dnsrv.040209-1620)	56,50 KB (57 856 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\rassfm.dll
kdcsvc	5.2.3790.1159 (dnsrv.040209-1620)	572,00 KB (585 728 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\kdcsvc.dll
ntdsa	5.2.3790.1159 (dnsrv.040209-1620)	3,91 MB (4 101 632 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\ntdsa.dll
ntdsatq	5.2.3790.1159 (dnsrv.040209-1620)	80,50 KB (82 432 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\ntdsatq.dll
mswsock	5.2.3790.1159 (dnsrv.040209-1620)	677,00 KB (693 248 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\mswsock.dll
esent	5.2.3790.1159 (dnsrv.040209-1620)	2,49 MB (2 615 808 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\esent.dll
scecli	5.2.3790.1159 (dnsrv.040209-1620)	468,00 KB (479 232 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\scecli.dll
wshtcpip	5.2.3790.0 (srv03_rtm.030324-2048)	38,00 KB (38 912 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wshtcpip.dll
dssenh	5.2.3790.1159 (dnsrv.040209-1620)	327,98 KB (335 848 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\dssenh.dll
svchost	5.2.3790.1159 (dnsrv.040209-1620)	35,00 KB (35 840 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\svchost.exe
rpcss	5.2.3790.1159 (dnsrv.040209-1620)	669,50 KB (685 568 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\rpcss.dll
wkssvc	5.2.3790.1159 (dnsrv.040209-1620)	306,50 KB (313 856 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wkssvc.dll
dmserver	5.2.3790.0 (srv03_rtm.030324-2048)	45,50 KB (46 592 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\dmserver.dll
es	2001.12.4720.1159 (dnsrv.040209-1620)	635,00 KB (650 240 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\es.dll
pchsvc	5.2.3790.0 (srv03_rtm.030324-2048)	94,50 KB (96 768 bytes)	09/04/2004 17:42	Microsoft Corporation	c:\windows\pchealth\helpctr\binaries\pchsvc.dll
srvsvc	5.2.3790.1159 (dnsrv.040209-1620)	188,00 KB (192 512 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\srvsvc.dll
sacsvr	5.2.3790.0 (srv03_rtm.030324-2048)	27,50 KB (28 160 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\sacsvr.dll
sens	5.2.3790.0 (srv03_rtm.030324-2048)	90,50 KB (92 672 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\sens.dll
wmisvc	5.2.3790.1159 (dnsrv.040209-1620)	420,50 KB (430 592 bytes)	09/04/2004 17:40	Microsoft Corporation	c:\windows\system32\wbem\wmisvc.dll
vssapi	5.2.3790.1159 (dnsrv.040209-1620)	1,29 MB (1 348 096 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\vssapi.dll
atl	3.00.2282	348,00 KB (356 352 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\atl.dll
comsvcs	2001.12.4720.1159 (dnsrv.040209-1620)	3,01 MB (3 151 872 bytes)	09/04/2004 17:40	Microsoft Corporation	c:\windows\system32\comsvcs.dll
wiarpc	5.2.3790.1159 (dnsrv.040209-1620)	69,50 KB (71 168 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wiarpc.dll
netman	5.2.3790.1159 (dnsrv.040209-1620)	592,50 KB (606 720 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\netman.dll
mprapi	5.2.3790.1159 (dnsrv.040209-1620)	244,50 KB (250 368 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\mprapi.dll
activeds	5.2.3790.1159 (dnsrv.040209-1620)	545,50 KB (558 592 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\activeds.dll
adsldpc	5.2.3790.1159 (dnsrv.040209-1620)	315,50 KB (323 072 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\adsldpc.dll
credui	5.2.3790.0 (srv03_rtm.030324-2048)	288,00 KB (294 912 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\credui.dll
rtutils	5.2.3790.1159 (dnsrv.040209-1620)	82,00 KB (83 968 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\rtutils.dll
rasapi32	5.2.3790.1159 (dnsrv.040209-1620)	596,50 KB (610 816 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\rasapi32.dll
rasman	5.2.3790.1159 (dnsrv.040209-1620)	155,00 KB (158 720 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\rasman.dll
tapi32	5.2.3790.0 (srv03_rtm.030324-2048)	493,00 KB (504 832 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\tapi32.dll

wzcsvc	5.2.3790.1159 (dnsrv.040209-1620)	637,50 KB (652 800 bytes)	10/02/2004 00:48	Microsoft Corporation	c:\windows\system32\wzcsvc.dll	
wmi	5.2.3790.0 (srv03_rtm.030324-2048)	5,00 KB (5 120 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wmi.dll	
dhcpcsvc	5.2.3790.1159 (dnsrv.040209-1620)	287,00 KB (293 888 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\dhcpcsvc.dll	
wzcsapi	5.2.3790.1159 (dnsrv.040209-1620)	61,50 KB (62 976 bytes)	10/02/2004 00:48	Microsoft Corporation	c:\windows\system32\wzcsapi.dll	
netshell	5.2.3790.1159 (dnsrv.040209-1620)	2,66 MB (2 789 888 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\netshell.dll	
clusapi	5.2.3790.1159 (dnsrv.040209-1620)	166,00 KB (169 984 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\clusapi.dll	
hnetcfg	5.2.3790.1159 (dnsrv.040209-1620)	765,50 KB (783 872 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\hnetcfg.dll	
wininet	6.00.3790.1159 (dnsrv.040209-1620)	1,44 MB (1 506 816 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wininet.dll	
wbemcore	5.2.3790.1159 (dnsrv.040209-1620)	1,70 MB (1 785 344 bytes)	09/04/2004 17:40	Microsoft Corporation	c:\windows\system32\wbem\wbemcore.dll	
esscli	5.2.3790.1159 (dnsrv.040209-1620)	935,50 KB (957 952 bytes)	09/04/2004 17:40	Microsoft Corporation	c:\windows\system32\wbem\esscli.dll	
wmiutils	5.2.3790.1159 (dnsrv.040209-1620)	290,00 KB (296 960 bytes)	09/04/2004 17:40	Microsoft Corporation	c:\windows\system32\wbem\wmiutils.dll	
repdrvfs	5.2.3790.1159 (dnsrv.040209-1620)	614,50 KB (629 248 bytes)	09/04/2004 17:40	Microsoft Corporation	c:\windows\system32\wbem\repdrvfs.dll	
wmiprvsd	5.2.3790.1159 (dnsrv.040209-1620)	1,38 MB (1 450 496 bytes)	09/04/2004 17:40	Microsoft Corporation	c:\windows\system32\wbem\wmiprvsd.dll	
wbemess	5.2.3790.1159 (dnsrv.040209-1620)	985,00 KB (1 008 640 bytes)	09/04/2004 17:40	Microsoft Corporation	c:\windows\system32\wbem\wbemess.dll	
ncprov	5.2.3790.1159 (dnsrv.040209-1620)	134,00 KB (137 216 bytes)	09/04/2004 17:40	Microsoft Corporation	c:\windows\system32\wbem\ncprov.dll	
rasdlg	5.2.3790.1159 (dnsrv.040209-1620)	1,36 MB (1 427 968 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\rasdlg.dll	
rasadhlp	5.2.3790.0 (srv03_rtm.030324-2048)	13,00 KB (13 312 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\rasadhlp.dll	
ntlsapi	5.2.3790.0 (srv03_rtm.030324-2048)	14,50 KB (14 848 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\ntlsapi.dll	
wbemcons	5.2.3790.1159 (dnsrv.040209-1620)	239,00 KB (244 736 bytes)	09/04/2004 17:40	Microsoft Corporation	c:\windows\system32\wbem\wbemcons.dll	
portmap	1.14.3	40,00 KB (40 960 bytes)	27/04/2004 09:47	Not Available	c:\program files (x86)\qlogic corporation\sansurfer\portmap.exe	
wow64	5.2.3790.1159 (dnsrv.040209-1620)	432,50 KB (442 880 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wow64.dll	
wow64win	5.2.3790.1159 (dnsrv.040209-1620)	518,00 KB (530 432 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wow64win.dll	
wow64cpu	5.2.3790.1159 (dnsrv.040209-1620)	40,50 KB (41 472 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wow64cpu.dll	
mssearch	9.107.8320.0	460,00 KB (471 040 bytes)	06/12/2002 19:07	Microsoft Corporation	c:\program files\common files\system\mssearch\bin\mssearch.exe	
mssws	9.107.8320.0	27,00 KB (27 648 bytes)	06/12/2002 19:07	Microsoft Corporation	c:\program files\common files\system\mssearch\bin\mssws.dll	
mssrch	9.107.8320.0	6,50 MB (6 819 328 bytes)	06/12/2002 19:07	Microsoft Corporation	c:\progra~1\common~1\system\mssearch\bin\mssrch.dll	
security	5.2.3790.0 (srv03_rtm.030324-2048)	7,00 KB (7 168 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\security.dll	
tquery	9.107.8320.0	5,85 MB (6 133 248 bytes)	06/12/2002 19:07	Microsoft Corporation	c:\program files\common files\system\mssearch\bin\tquery.dll	
propdefs	9.107.8320.0	888,50 KB (909 824 bytes)	06/12/2002 19:07	Microsoft Corporation	c:\progra~1\common~1\system\mssearch\bin\propdefs.dll	
srchidx	9.107.8320.0	2,26 MB (2 374 144 bytes)	06/12/2002 19:07	Microsoft Corporation	c:\progra~1\common~1\system\mssearch\bin\srchidx.dll	
iprop	5.2.3790.0 (srv03_rtm.030324-2048)	3,00 KB (3 072 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\iprop.dll	
explorer	6.00.3790.1159 (dnsrv.040209-1620)	1,63 MB (1 709 568 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\explorer.exe	
browseui	6.00.3790.1159 (dnsrv.040209-1620)	2,42 MB (2 539 520 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\browseui.dll	
shdocvw	6.00.3790.1159 (dnsrv.040209-1620)	3,21 MB (3 365 888 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\shdocvw.dll	
apphelp	5.2.3790.1159 (dnsrv.040209-1620)	266,50 KB (272 896 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\apphelp.dll	
themeui	6.00.3790.1159 (dnsrv.040209-1620)	825,50 KB (845 312 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\themeui.dll	
msimg32	5.2.3790.0 (srv03_rtm.030324-2048)	7,00 KB (7 168 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\msimg32.dll	
msutb	5.2.3790.1159 (dnsrv.040209-1620)	563,50 KB (577 024 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\msutb.dll	
msctf	5.2.3790.1159 (dnsrv.040209-1620)	931,00 KB (953 344 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\msctf.dll	
linkinfo	5.2.3790.1159 (dnsrv.040209-1620)	42,50 KB (43 520 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\linkinfo.dll	
ntshrui	6.00.3790.0 (srv03_rtm.030324-2048)	233,50 KB (239 104 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\ntshrui.dll	
urlmon	6.00.3790.1159 (dnsrv.040209-1620)	1,22 MB (1 277 952 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\urlmon.dll	
webcheck	6.00.3790.0 (srv03_rtm.030324-2048)	664,50 KB (680 448 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\webcheck.dll	
wsock32	5.2.3790.0 (srv03_rtm.030324-2048)	23,00 KB (23 552 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wsock32.dll	
stobject	5.2.3790.1159 (dnsrv.040209-1620)	178,00 KB (182 272 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\stobject.dll	
batmeter	6.00.3790.0 (srv03_rtm.030324-2048)	55,50 KB (56 832 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\batmeter.dll	
powrprof	6.00.3790.0 (srv03_rtm.030324-2048)	36,00 KB (36 864 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\powrprof.dll	
printui	5.2.3790.1159 (dnsrv.040209-1620)	1,09 MB (1 147 904 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\printui.dll	
cfgmgr32	5.2.3790.0 (srv03_rtm.030324-2048)	16,00 KB (16 384 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\cfgmgr32.dll	
browselc	6.00.3790.0 (srv03_rtm.030324-2048)	61,50 KB (62 976 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\browselc.dll	
shdoclc	6.00.3790.0 (srv03_rtm.030324-2048)	588,00 KB (602 112 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\shdoclc.dll	
wshext	5.6.0.8515	150,50 KB (154 112 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\wshext.dll	
mfc42	6.50.4012.0	3,37 MB (3 530 752 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\mfc42.dll	
mydocs	6.00.3790.0 (srv03_rtm.030324-2048)	129,00 KB (132 096 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\mydocs.dll	
actxprxy	6.00.3790.0 (srv03_rtm.030324-2048)	230,00 KB (235 520 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\actxprxy.dll	
ctfmon	5.2.3790.0 (srv03_rtm.030324-2048)	31,50 KB (32 256 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\ctfmon.exe	
ctfmon	5.2.3790.0 (srv03_rtm.030324-2048)	13,50 KB (13 824 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\syswow64\ctfmon.exe	
sqlmangr	2000.080.0760.00	225,00 KB (230 400 bytes)	07/02/2003 05:48	Microsoft Corporation	c:\program files\microsoft sql server\80\tools\binn\sqlmangr.exe	
w95scm	2000.080.0760.00	133,50 KB (136 704 bytes)	07/02/2003 05:49	Microsoft Corporation	c:\program files\microsoft sql server\80\tools\binn\w95scm.dll	
odbc32	3.525.1111.0 (dnsrv.040209-1620)	612,00 KB (626 688 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\odbc32.dll	
sqlsvc	2000.080.0760.00	315,50 KB (323 072 bytes)	07/02/2003 05:49	Microsoft Corporation	c:\program files\microsoft sql server\80\tools\binn\sqlsvc.dll	
odbcbcp	2000.085.1111.00 (dnsrv.040209-1620)	48,00 KB (49 152 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\odbcbcp.dll	
sqlresld	2000.080.0760.00	28,50 KB (29 184 bytes)	07/02/2003 05:49	Microsoft Corporation	c:\program files\microsoft sql server\80\tools\binn\sqlresld.dll	
odbcint	3.525.1111.0 (dnsrv.040209-1620)	88,00 KB (90 112 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\odbcint.dll	
resutils	5.2.3790.1159 (dnsrv.040209-1620)	148,00 KB (151 552 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\resutils.dll	
sqlsvc	2000.080.0760.00	2,00 KB (2 048 bytes)	06/02/2003 21:09	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)	06/02/2003 21:15	Microsoft Corporation	c:\program files\microsoft sql server\80\tools\binn\resources\1033\sqlmangr.rll	
cmd	5.2.3790.1159 (dnsrv.040209-1620)	493,50 KB (505 344 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\cmd.exe	
calc	5.2.3790.0 (srv03_rtm.030324-2048)	269,00 KB (275 456 bytes)	09/04/2004 17:40	Microsoft Corporation	c:\windows\system32\calc.exe	
notepad	5.2.3790.0 (srv03_rtm.030324-2048)	129,50 KB (132 608 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\notepad.exe	
helpctr	5.2.3790.1159 (dnsrv.040209-1620)	1,97 MB (2 068 992 bytes)	09/04/2004 17:42	Microsoft Corporation	c:\windows\pchealth\helpctr\binaries\helpctr.exe	
hcappres	5.2.3790.0 (srv03_rtm.030324-2048)	6,00 KB (6 144 bytes)	09/04/2004 17:42	Microsoft Corporation	c:\windows\pchealth\helpctr\binaries\hcappres.dll	
itss	5.2.3790.1159 (dnsrv.040209-1620)	349,00 KB (357 376 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\itss.dll	
msxml3	8.50.2132.0	3,48 MB (3 653 120 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\msxml3.dll	
winhttp	5.2.3790.1159 (dnsrv.040209-1620)	894,50 KB (915 968 bytes)	09/04/2004 10:30	Microsoft Corporation	c:\windows\winsxs\ia64_microsoft.windows.winhttp_6595b64144ccf1df_5.1.0.1_x-ww_0fbfffd7\winhttp.dll	
pchshell	5.2.3790.0 (srv03_rtm.030324-2048)	277,50 KB (284 160 bytes)	09/04/2004 17:42	Microsoft Corporation	c:\windows\pchealth\helpctr\binaries\pchshell.dll	
mlang	6.00.3790.1159 (dnsrv.040209-1620)	799,50 KB (818 688 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\mlang.dll	
mshtml	6.00.3790.1159 (dnsrv.040209-1620)	7,86 MB (8 243 200 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\mshtml.dll	
msimtf	5.2.3790.1159 (dnsrv.040209-1620)	529,00 KB (541 696 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\msimtf.dll	
jscript	5.6.0.8821	1,22 MB (1 278 976 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\jscript.dll	
msls31	3.10.349.0	448,00 KB (458 752 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\msls31.dll	
imm32	5.2.3790.1159 (dnsrv.040209-1620)	307,50 KB (314 880 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\imm32.dll	
mshtmled	6.00.3790.0 (srv03_rtm.030324-2048)	1,34 MB (1 408 512 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\mshtmled.dll	
vbscript	5.6.0.8821	1,07 MB (1 117 696 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\vbscript.dll	
msinfo	5.2.3790.0 (srv03_rtm.030324-2048)	1,20 MB (1 257 984 bytes)	09/04/2004 17:42	Microsoft Corporation	c:\windows\pchealth\helpctr\binaries\msinfo.dll	
mfc42u	6.50.4012.0	3,35 MB (3 510 272 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\mfc42u.dll	
riched32	5.2.3790.0 (srv03_rtm.030324-2048)	5,00 KB (5 120 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\riched32.dll	
riched20	5.31.23.1218	1,25 MB (1 313 280 bytes)	11/02/2004 13:00	Microsoft Corporation	c:\windows\system32\riched20.dll	
helpsvc	5.2.3790.1159 (dnsrv.040209-1620)	2,19 MB (2 295 296 bytes)	09/04/2004 17:42	Microsoft Corporation	c:\windows\pchealth\helpctr\binaries\helpsvc.exe	

[Services]

Display Name	Name	State	Start Mode	Service Type	Path	Error Control	Start Name	Tag ID			
Alerter	Alerter	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k localservice		Normal	NT AUTHORITY\LocalService	0		
Application Layer Gateway Service	ALG	Stopped	Manual	Own Process	c:\windows\system32\alg.exe		Normal	NT AUTHORITY\LocalService	0		
Application Management	AppMgmt	Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		
Windows Audio	AudioSrv	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		
Background Intelligent Transfer Service	BITS	Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		
Computer Browser	Browser	Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		
Indexing Service	CiSvc	Stopped	Disabled	Share Process	c:\windows\system32\cisvc.exe	Normal	LocalSystem	0			
ClipBook	ClipSrv	Stopped	Disabled	Own Process	c:\windows\system32\clipsrv.exe	Normal	LocalSystem	0			
COM+ System Application	COMSysApp	Stopped	Manual	Own Process	c:\windows\system32\dllhost.exe /processid:{02d4b3f1-fd88-11d1-960d-00805fc79235}				Normal	LocalSystem	0
Cryptographic Services	CryptSvc	Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		
Distributed File System	Dfs	Stopped	Manual	Own Process	c:\windows\system32\dfssvc.exe		Normal	LocalSystem	0		
DHCP Client	Dhcp	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k networkservice		Normal	NT AUTHORITY\NetworkService	0		
Logical Disk Manager Administrative Service	dmadmin	Stopped	Manual	Share Process	c:\windows\system32\dmadmin.exe /com				Normal	LocalSystem	0
Logical Disk Manager	dmserver	Running	Auto	Share Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		
DNS Client	Dnscache	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k networkservice		Normal	NT AUTHORITY\NetworkService	0		
Error Reporting Service	ERSvc	Stopped	Disabled	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		
HID Input Service	HidServ	Stopped	Manual	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\llsrv.exe		Normal	NT AUTHORITY\NetworkService	0		
TCP/IP NetBIOS Helper	LmHosts	Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k localservice				Normal	NT AUTHORITY\LocalService	0
Messenger	Messenger	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		
NetMeeting Remote Desktop Sharing	mnmsrvc	Stopped	Disabled	Own Process	c:\windows\system32\mnmsrvc.exe		Normal	LocalSystem	0		
Distributed Transaction Coordinator	MSDTC	Stopped	Disabled	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
MyStorage Remote HBA Server	MyStorage Rmt Server	Stopped	Manual	Own Process	c:\program files (x86)\lsilogic\mystorage\mystorage_rmtsvr.exe				Normal	LocalSystem	0
NEC Storage Manager	NEC Storage Manager	Stopped	Manual	Own Process	"c:\program files (x86)\nec\ismsvr\sbin\ismmainnt.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	Stopped	Manual	Share Process	c:\windows\system32\lsass.exe		Normal	LocalSystem	0		
ONC/RPC Portmapper	portmap	Running	Auto	Own Process	c:\program files (x86)\qlogic corporation\sansurfer\portmap.exe				Normal	LocalSystem	0

Protected Storage	ProtectedStorage	Stopped	Manual	Share Process	c:\windows\system32\lsass.exe	Normal	LocalSystem	0			
QLogic Management Suite Java Agent	QLManagementAgentJava	Stopped	Manual	Own Process	c:\progra~2\qlogic~1\sansur~1\qlremote.exe	Normal	LocalSystem	0			
Remote Access Auto Connection Manager		RasAuto	Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0		
Remote Access Connection Manager	RasMan	Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
Remote Desktop Help Session Manager		RDSessMgr	Stopped	Manual	Own Process	c:\windows\system32\sessmgr.exe	Normal	LocalSystem	0		
Routing and Remote Access	RemoteAccess	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
Remote Registry	RemoteRegistry	Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k regsvc	Normal	NT AUTHORITY\LocalService	0			
Remote Procedure Call (RPC) Locator	RpcLocator	Stopped	Manual	Own Process	c:\windows\system32\locator.exe	Normal	NT AUTHORITY\NetworkService	0			
Remote Procedure Call (RPC)	RpcSs	Running	Auto	Share Process	c:\windows\system32\svchost.exe -k rpcss	Normal	LocalSystem	0			
Resultant Set of Policy Provider	RSoPProv	Stopped	Manual	Share Process	c:\windows\system32\rsopprov.exe	Normal	LocalSystem	0			
Special Administration Console Helper	sacsvr	Running	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
Security Accounts Manager	SamSs	Running	Auto	Share Process	c:\windows\system32\lsass.exe	Normal	LocalSystem	0			
Smart Card	SCardSvr	Stopped	Manual	Share Process	c:\windows\system32\scardsvr.exe	Ignore	NT AUTHORITY\LocalService	0			
Task Scheduler	Schedule	Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
Secondary Logon	seclogon	Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Ignore	LocalSystem	0			
System Event Notification	SENS	Running	Auto	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
Shell Hardware Detection	ShellHWDetection	Running	Auto	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Ignore	LocalSystem	0			
Print Spooler	Spooler	Stopped	Disabled	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	Auto	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	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k termsvcs	Normal	LocalSystem	0			
Themes	Themes	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
Telnet	TlntSvr	Stopped	Disabled	Own Process	c:\windows\system32\tlntsvr.exe	Normal	NT AUTHORITY\LocalService	0			
Distributed Link Tracking Server	TrkSvr	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
Distributed Link Tracking Client	TrkWks	Stopped	Disabled	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	NT AUTHORITY\LocalService	0			
Virtual Disk Service	vds	Stopped	Manual	Own Process	c:\windows\system32\vds.exe	Normal	LocalSystem	0			
Volume Shadow Copy	VSS	Stopped	Manual	Own Process	c:\windows\system32\vssvc.exe	Normal	LocalSystem	0			
Windows Time	W32Time	Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
WebClient	WebClient	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k localservice	Normal	NT AUTHORITY\LocalService	0			
WinHTTP Web Proxy Auto-Discovery Service		WinHttpAutoProxySvc		Stopped	Manual	Share Process	c:\windows\system32\svchost.exe -k localservice	Normal	NT AUTHORITY\LocalService	0	
Windows Management Instrumentation	winmgmt	Running	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	Stopped	Disabled	Share Process	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
Wireless Configuration	WZCSVC	Stopped	Disabled	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									
NEC Storage Manager Client	All Users:NEC Storage Manager Client	All Users									
NEC Storage Manager Server	All Users:NEC Storage Manager Server	All Users									
NEC Storage Manager Server\Configuration Setting Command Design	All Users:NEC Storage Manager Server\Configuration Setting Command Design	All Users									
NEC Storage Manager Server\Performance Operation Command Design	All Users:NEC Storage Manager Server\Performance Operation Command Design	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	TPCC-FAME\Administrator:Accessories	TPCC-FAME\Administrator									
Accessories\Accessibility	TPCC-FAME\Administrator:Accessories\Accessibility	TPCC-FAME\Administrator									
Accessories\Entertainment	TPCC-FAME\Administrator:Accessories\Entertainment	TPCC-FAME\Administrator									
Administrative Tools	TPCC-FAME\Administrator:Administrative Tools	TPCC-FAME\Administrator									
MyStorage	TPCC-FAME\Administrator:MyStorage	TPCC-FAME\Administrator									
QLogic Corporation	TPCC-FAME\Administrator:QLogic Corporation	TPCC-FAME\Administrator									
QLogic Corporation\SANblade Control VIX	TPCC-FAME\Administrator:QLogic Corporation\SANblade Control VIX	TPCC-FAME\Administrator									
QLogic Management Suite	TPCC-FAME\Administrator:QLogic Management Suite	TPCC-FAME\Administrator									
Startup	TPCC-FAME\Administrator:Startup	TPCC-FAME\Administrator									

[Startup Programs]

Program	Command	User Name	Location								
desktop	desktop.ini	NT AUTHORITY\SYSTEM	Startup								
CTFMON.EXE	c:\windows\system32\ctfmon.exe	NT AUTHORITY\SYSTEM	HKU\S-1-5-18\SOFTWARE\Microsoft\Windows\CurrentVersion\Run								
desktop	desktop.ini	TPCC-FAME\Administrator	Startup								
CTFMON.EXE	c:\windows\system32\ctfmon.exe	TPCC-FAME\Administrator	HKU\S-1-5-21-2838299447-3930895136-1706246558-500\SOFTWARE\Microsoft\Windows\CurrentVersion\Run								
desktop	desktop.ini	.DEFAULT	Startup								
CTFMON.EXE	c:\windows\system32\ctfmon.exe	.DEFAULT	HKU\.DEFAULT\SOFTWARE\Microsoft\Windows\CurrentVersion\Run								
desktop	desktop.ini	All Users	Common Startup								
IDW Logging Tool	c:\windows\system32\idwlog.exe -3	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	6.0.3790.0
Build	63790
Application Path	C:\Program Files\Internet Explorer
Language	English (United States)
Active Printer	Not Available
Cipher Strength	128-bit
Content Advisor	Disabled
IEAK Install	No

[File Versions]

File	Version	Size	Date	Path	Company	
actxprxy.dll	6.0.3790.0	230 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
advpack.dll	6.0.3790.0	240 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
asctrls.ocx	6.0.3790.0	219 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
browselc.dll	6.0.3790.0	62 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
browseui.dll	6.0.3790.1159	2 480 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
cdfview.dll	6.0.3790.0	292 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
comctl32.dll	5.82.3790.1159	1 586 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
dxtrans.dll	6.3.3790.0	562 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
dxtmsft.dll	6.3.3790.0	918 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
iecont.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available	
iecontlc.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available	
iedkcs32.dll	16.0.3790.1159	677 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
iepeers.dll	6.0.3790.0	652 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
iesetup.dll	6.0.3790.0	89 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
ieuinit.inf	Not Available	20 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Not Available
iexplore.exe	6.0.3790.0	102 KB	11/02/2004 14:00:00		C:\Program Files\Internet Explorer	Microsoft Corporation
imgutil.dll	5.2.3790.0	101 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
inetcpl.cpl	6.0.3790.0	589 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation

inetcplc.dll	6.0.3790.0	108 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
inseng.dll	6.0.3790.0	213 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
mlang.dll	6.0.3790.1159	800 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
msencode.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available	
mshta.exe	6.0.3790.1159	60 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
mshtml.dll	6.0.3790.1159	8 050 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
mshtml.tlb	6.0.3790.0	1 319 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
mshtmled.dll	6.0.3790.0	1 376 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
mshtmler.dll	6.0.3790.0	56 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
msident.dll	6.0.3790.0	128 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
msidntld.dll	6.0.3790.0	14 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
msieftp.dll	6.0.3790.0	536 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
msrating.dll	6.0.3790.0	379 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
mstime.dll	6.0.3790.0	1 621 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
occache.dll	6.0.3790.0	201 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
proctexe.ocx	<File Missing>	Not Available	Not Available	Not Available	Not Available	
sendmail.dll	6.0.3790.1159	98 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
shdoclc.dll	6.0.3790.0	588 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
shdocvw.dll	6.0.3790.1159	3 287 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
shfolder.dll	6.0.3790.0	37 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
shlwapi.dll	6.0.3790.1159	724 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
tdc.ocx	1.3.0.3130	177 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
url.dll	6.0.3790.0	45 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
urlmon.dll	6.0.3790.1159	1 248 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
webcheck.dll	6.0.3790.0	665 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation
wininet.dll	6.0.3790.1159	1 472 KB	11/02/2004 14:00:00		C:\WINDOWS\system32	Microsoft Corporation

[Connectivity]

Item Value
Connection Preference Never dial

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 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	Custom
Trusted sites	Medium
Internet	High
Restricted sites	Custom

Microsoft® SQL Serveur™ 2000 setting

Startup Parameters

```
sqlservr -c -x -g192 -T3502 -T3428 -T888 -T825
```

Microsoft® SQL Serveur™ 2000 Configuration Parameters

Mesures\NS5080\sql_version.txt

```
1> 2> 3> 4> 5> 6> 7> 8> 9> 10> 11> 12> 1> 2> 3> 4> 5>
 ------------------- ------------------- ----------------------------
 8.00.883          SP3              Enterprise Edition (64-bit)

(1 row affected)
1> 2> 3>
 -----------------------------
 2004-06-0912:56:07.210

(1 row affected)
1> 2> 3> 4> 5>
```

Mesures\NS5080\sp_config.txt

```
 name                              minimum     maximum     config_value
                         run_value
 --------------------------------- ----------- ----------- ------------
                         -----------
 affinity mask                    -2147483648  2147483647         255
                               255
 affinity64 mask                    -2147483648  2147483647          0
                                 0
 allow updates                              0          1           0
```

awe enabled	0	1	0	0	0
c2 audit mode	0	1	0	0	0
cost threshold for parallelism	0	32767	0	0	0
Cross DB Ownership Chaining	0	1	0	0	0
cursor threshold	-1	2147483647	-1	-1	-1
default full-text language	0	2147483647	1033	1033	1033
default language	0	9999	0	0	0
fill factor (%)	0	100	0	0	0
index create memory (KB)	704	2147483647	0	0	0
lightweight pooling	0	1	1	1	1
locks	5000	2147483647	0	0	0
max degree of parallelism	0	32	1	1	1
max server memory (MB)	4	2147483647	125600	125600	125600
max text repl size (B)	0	2147483647	65536	65536	65536
max worker threads	32	32767	560	560	560
media retention	0	365	0	0	0
min memory per query (KB)	512	2147483647	512	512	512
min server memory (MB)	0	2147483647	0	0	0
nested triggers	0	1	1	1	1
network packet size (B)	512	65536	4096	4096	4096
open objects	0	2147483647	0	0	0
priority boost	0	1	1	1	1
query governor cost limit	0	2147483647	0	0	0
query wait (s)	-1	2147483647	2147483647	2147483647	2147483647
recovery interval (min)	0	32767	118	118	118
remote access	0	1	1	1	1
remote login timeout (s)	0	2147483647	20	20	20
remote proc trans	0	1	0	0	0
remote query timeout (s)	0	2147483647	600	600	600
scan for startup procs	0	1	0	0	0
set working set size	0	1	0	0	0
show advanced options	0	1	1	1	1
two digit year cutoff	1753	9999	2049	2049	2049
user connections	0	32767	0	0	0
user options	0	32767	0	0	0

Appendix D: Database space calculation table.

60 Day Space

Note: Numbers are in MBytes unless otherwise specified

Warehouses:	14 000	tpmC	172 368	tpmC/W	12,31

Table	Rows	Data	Clustered Index	Non Clustered Index	Free Space	8H Space	Total Space
					2%		
Warehouse	14 000	1,46	0,05	0,05	5,00		6,56
District	140 000	15,22	0,53	0,53	0,30		16,59
Customer.1	420 000 000	74 901,58	1 605,36	21 880,86	1 498,03		99 885,83
Customer.2	420 000 000	200 271,61	0,00	0,00	4 005,43		204 277,04
History	420 000 000	23 231,51	1 602,17	1 602,17		4 933,57	31 369,42
Orders	420 000 000	13 217,93	1 605,36	0,00		8 654,29	23 477,58
New Order	126 000 000	2 162,93	481,61	480,65		884,04	4 009,24
Order Line	4 200 000 000	260 353,09	16 055,58	16 021,73		51 638,86	344 069,26
Stock	1 400 000 000	425 910,95	5 350,53	5 340,58	8 518,22		445 120,27
Item	100 000	9,35	0,38	0,38	0,19		10,30
Totals		1 000 075,61	26 701,58	45 326,96	14 027,17	66 110,76	1 152 242,09

DB File Group	Count	Size	Needed	Overhead	Not Needed
MSSQL_misc_fg	12	453 120,00	402 958,94	4 029,59	46 131,47
MSSQL_cs_fg	12	552 960,00	545 006,11	5 450,06	2 503,83
MSSQL_cust_fg	12	253 440,00	204 277,04	2 042,77	47 120,19
Totals		1 006 080,00	947 965,05	9 479,65	48 635,30

Dynamic Space	296 803	Sum of Data for History, Orders and Order Line
Static space	753 481	Data + Index + 2% Space + Overhead - Dynamic space
Free space	-92 839	Total Seg.Size - Dynamic Space - Static Space - Not Needed
Daily growth	58 468	(Dynamic space / (W * 62,5)) * tpmC
Daily spread	0	Free space - 1,5 * Daily growth (zero if negative)
60 day	4 261 553	Static space + 60 (daily growth + daily spread)
60 day (GB)	4 161,67	Excludes OS, Paging and RDBMS Logs
8 Hour Log	436 785	Need double for mirroring
os, file sys, swap	0,63	

	Disk size (GB)	Priced Qty	Priced (GB)	Needed(GB)	Extra(GB)
Database	33,26	180	5 986,80	4 161,67	1 825,13
System OS&Swap	33,26	15	498,90	0,63	498,27
Mirrored Log	33,26	28	931,28	853,10	78,18
Total number of disks		223			
Total (GB)			7 417	5 015	2 402

Appendix E: Third party price quotations.

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
http://www.microsoft.com/

Microsoft

June 16, 2004

Groupe Bull
Olivier PAULY
68, route de Versailles
LOUVECIENNES, 78434

Mr. PAULY:

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	8	$132,328
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	8	$5,904
P72-00264	**Windows Server 2003, Enterprise Edition** ***For 64-bit Itanium-based Systems*** *Server license only - No CALs* *Discount Schedule: Open Program - No Level* *Unit Price reflects a 40% discount from the retail unit price of $3,999.*	$2,399	1	$2,399
254-00170	**Visual C++ Standard** *No discounts applied*	$109	1	$109

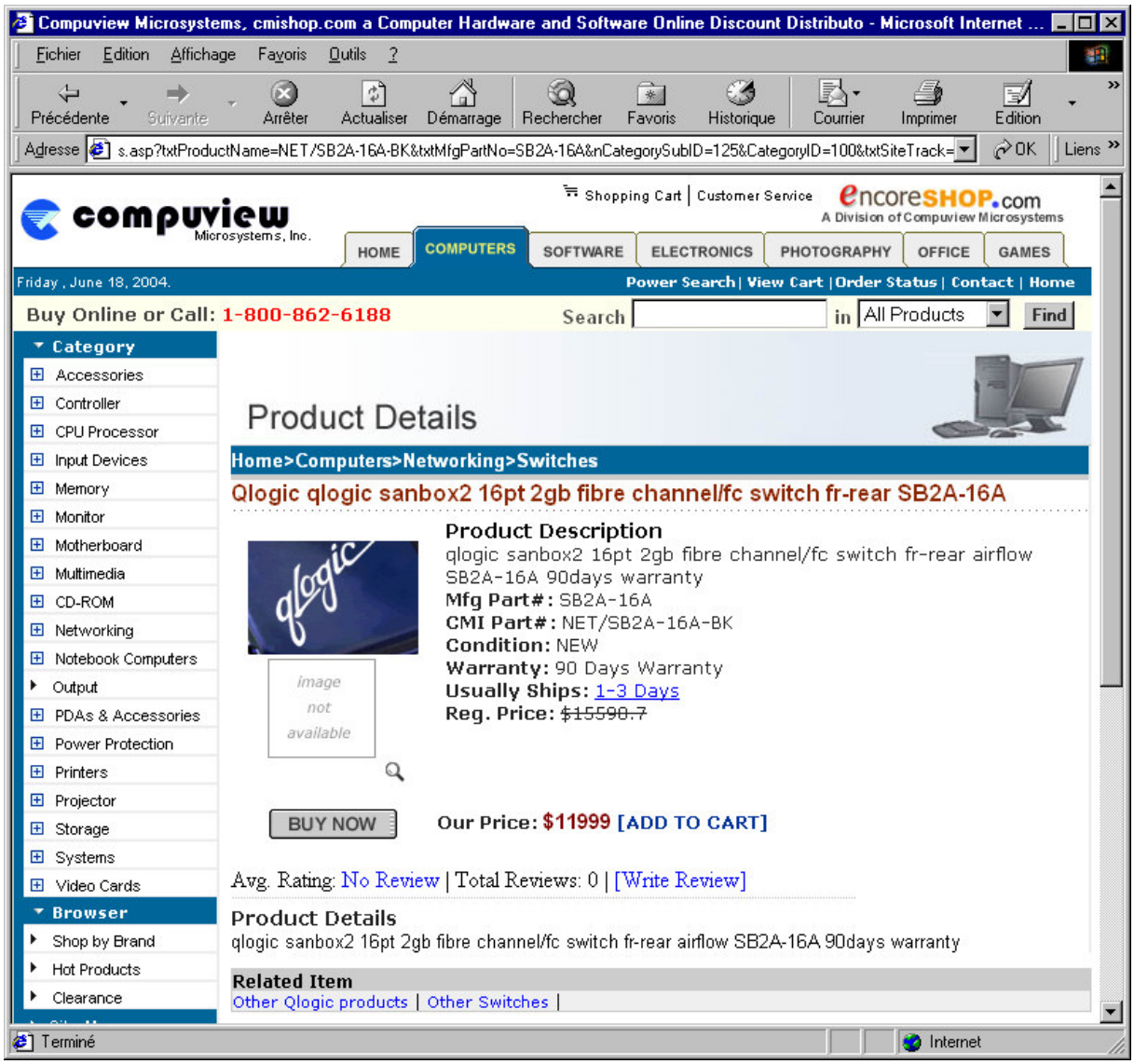
All products are currently orderable through Microsoft's normal distribution channels.

This quote is valid for the next 90 days.

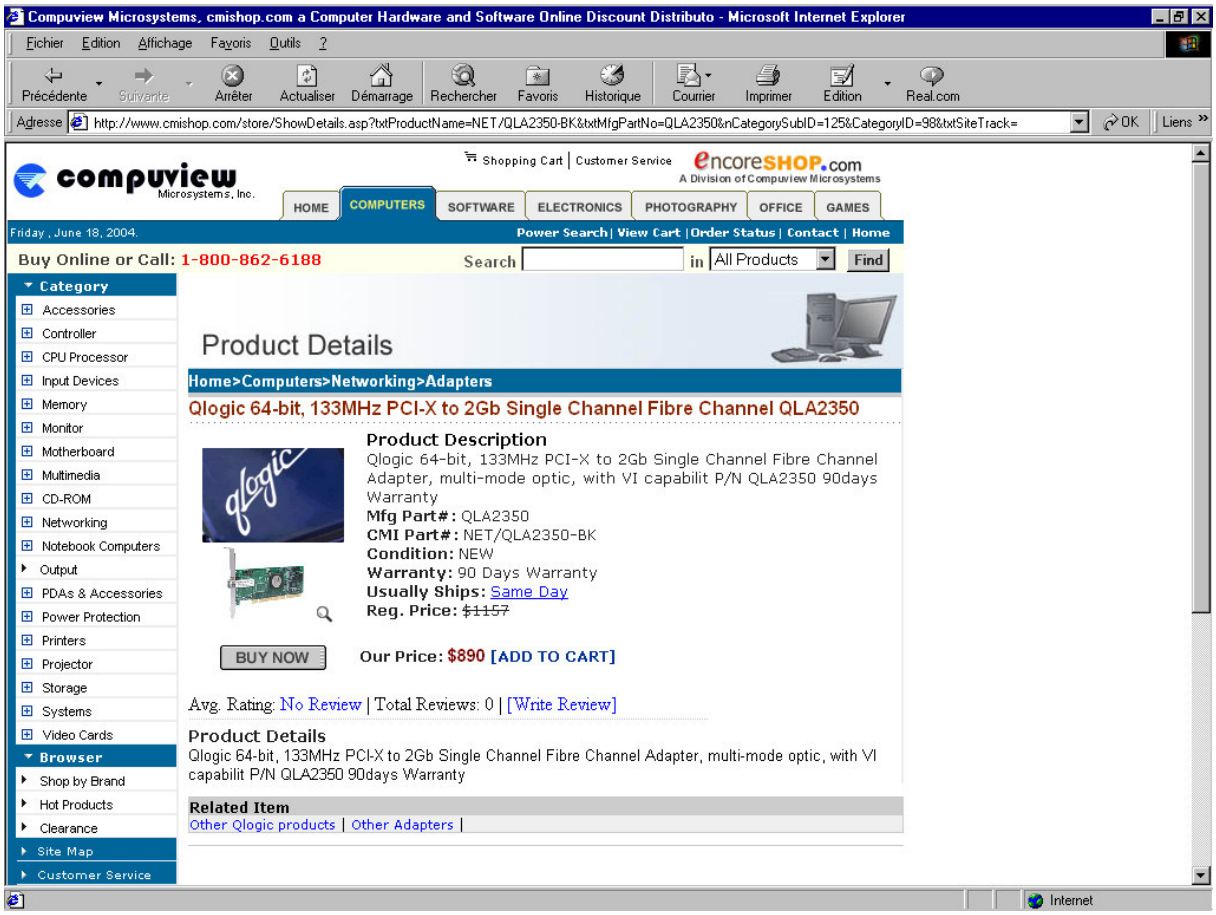
If we can be of any further assistance, please contact Jamie Reding at (425) 703-0510 or jamiere@microsoft.com.

Reference ID: PColpA0416062988
Please include this Reference ID in any correspondence regarding this price quote.

Qlogic SANbox2-16



Qlogic QLA2350-CK



Lsi LQI7202XP-LC