

Compaq Computer Corporation

TPC Benchmark™ C
Full Disclosure Report
for
ProLiant 8500-700-96P
using
Microsoft SQL Server 2000 Enterprise Edition
and
Windows 2000 Advanced Server

First Edition
July 25, 2000

COMPAQ

Compaq Computer Corporation (Compaq) believes that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. Compaq 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, Compaq provides no warranty of the pricing information in this document.

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

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

Copyright 2000 Compaq Computer Corporation.

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

Printed in U.S.A., 2000

Compaq, NonStop, ProLiant 8500, and ProLiant are registered trademarks of Compaq Computer Corporation.

Microsoft, Windows 2000 and SQL Server for Windows 2000 are registered trademarks of Microsoft Corporation.

Pentium III Xeon is a registered trademark of Intel.

TPC Benchmark is a trademark of the Transaction Processing Performance Council.

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

Table of Contents

TABLE OF CONTENTS	3
PREFACE	5
TPC BENCHMARK C OVERVIEW	5
ABSTRACT	6
OVERVIEW	6
TPC BENCHMARK C METRICS	6
STANDARD AND EXECUTIVE SUMMARY STATEMENTS	6
AUDITOR	6
GENERAL ITEMS	10
TEST SPONSOR	10
APPLICATION CODE AND DEFINITION STATEMENTS	10
PARAMETER SETTINGS	10
CONFIGURATION ITEMS	10
CLAUSE 1 RELATED ITEMS	12
TABLE DEFINITIONS	12
PHYSICAL ORGANIZATION OF DATABASE	12
<i>Benchmarked Configuration:</i>	12
PRICED CONFIGURATION VS. MEASURED CONFIGURATION	13
INSERT AND DELETE OPERATIONS	13
PARTITIONING	13
REPLICATION, DUPLICATION OR ADDITIONS	13
CLAUSE 2 RELATED ITEMS	14
RANDOM NUMBER GENERATION	14
INPUT/OUTPUT SCREEN LAYOUT	14
PRICED TERMINAL FEATURE VERIFICATION	14
PRESENTATION MANAGER OR INTELLIGENT TERMINAL	14
TRANSACTION STATISTICS	14
QUEUING MECHANISM	15
CLAUSE 3 RELATED ITEMS	16
TRANSACTION SYSTEM PROPERTIES (ACID)	16
ATOMICITY	16
<i>Completed Transactions</i>	16
<i>Aborted Transactions</i>	16
CONSISTENCY	16
ISOLATION	16
DURABILITY	18
<i>Durable Media Failure</i>	18
<i>Instantaneous Interruption and Loss of Memory</i>	18
CLAUSE 4 RELATED ITEMS	20
INITIAL CARDINALITY OF TABLES	20
DATABASE LAYOUT	20
TYPE OF DATABASE	21

DATABASE MAPPING.....	21
180 DAY SPACE	21
CLAUSE 5 RELATED ITEMS	22
THROUGHPUT	22
KEYING AND THINK TIMES	22
RESPONSE TIME FREQUENCY DISTRIBUTION CURVES AND OTHER GRAPHS	23
FIGURE 10. THROUGHPUT VS. TIME DISTRIBUTION.....	27
STEADY STATE DETERMINATION.....	28
WORK PERFORMED DURING STEADY STATE	28
REPRODUCIBILITY.....	28
MEASUREMENT PERIOD DURATION	29
REGULATION OF TRANSACTION MIX.....	29
TRANSACTION STATISTICS.....	29
CHECKPOINT COUNT AND LOCATION.....	29
CLAUSE 6 RELATED ITEMS	30
RTE DESCRIPTIONS	30
EMULATED COMPONENTS.....	30
FUNCTIONAL DIAGRAMS	30
NETWORKS	30
OPERATOR INTERVENTION	30
CLAUSE 7 RELATED ITEMS	31
SYSTEM PRICING.....	31
AVAILABILITY, THROUGHPUT, AND PRICE PERFORMANCE.....	31
COUNTRY SPECIFIC PRICING.....	31
USAGE PRICING.....	31
CLAUSE 9 RELATED ITEMS	32
AUDITOR'S REPORT	32
AVAILABILITY OF THE FULL DISCLOSURE REPORT	32

Preface

The TPC Benchmark C was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry. This full disclosure report is based on the TPC Benchmark C Standard Specifications Version 3.5, released October 25, 1999.

TPC Benchmark C Overview

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

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

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

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

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

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark C test conducted on the Compaq ProLiant 8500-700-96P cluster. The operating system used for the benchmark was Windows 2000 Advanced Server. The DBMS used was Microsoft SQL Server 2000 Enterprise Edition.

TPC Benchmark C Metrics

The standard TPC Benchmark C metrics, tpmC (transactions per minute), price per tpmC (five year capital cost per measured tpmC), and the availability date are reported as:

262,243.60 tpmC
\$20.24 per tpmC

The availability date is September 30, 2000.

Standard and Executive Summary Statements

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

Auditor

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

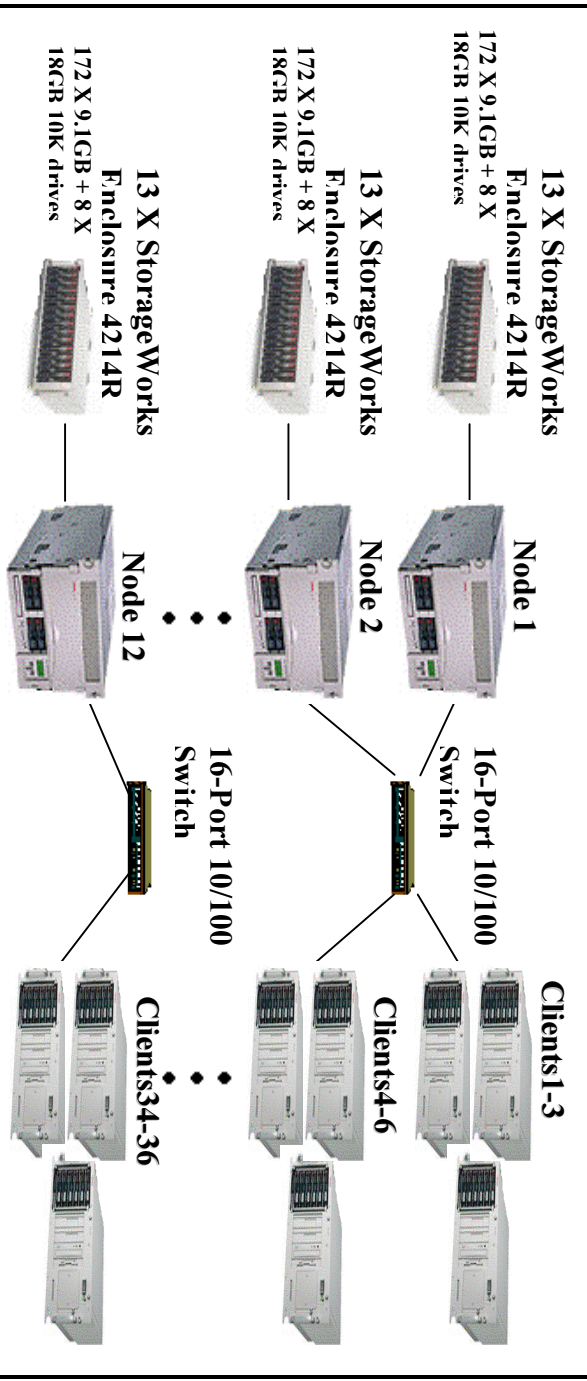
Compaq Computer Corporation

Proliant 8500-700-96F Client/Server

TPC-C Rev. 3.5

Report Date: Jul 25, 2000

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
\$5,305,571	262,243.60	\$20.24	Sep 30, 2000
Processors	Database Manager	Operating System	Other Software
96 Pentium Xeon 700 MHz	Microsoft SQL Server 2000 Enterprise Edition	Windows 2000 Advanced Server	Microsoft Visual C++ 216,000 MKS Toolkit for NT Microsoft COM+
			Number of Users



System Components	Quantity	Description	Quantity	Description
Processor	96	700MHz PentiumIXeon w/ 2MB Cache	34	Pentium III 600Mhz/512K
Memory	192	512MB	12	Pentium III 550Mhz/512K
Disk Controllers	60	Compaq SMART Array Controller 5304	26	Pentium III 733Mhz/512K
Disk Drives	12	Integrated SCSI RAID Controller	144	128MB
Total Storage Tape Drives	2064	9.1GB 10K Ultra2 Universal SCSI Drives	23	Integrated Ultra SCSI Controller
	96	18GB 10K Ultra2 Universal SCSI Drives	13	Integrated SCSI RAID Controller
	1	4/8 GB SLR Tape Drive	361	9.1GB 10K Ultra2 SCSI Drives
			304.9GB	

Compaq Computer Corporation

Client/Server

TPC-C Rev. 3.5

Report Date:

25-Jul-00

Proliant 8500-700-96P

Description	Part Number	Third Party	Brand Pricing	Unit Price	Qty	Extended Price	5 Yr. Maint. Price
Server Hardware							
Proliant 8500R X700-2M 8P 4G	168758-001	1		78,400	12	940,800	
- 8 GB SD RAM - Dual 10/100 Ethernet Controller							
1G-Memory Kit SDRAM	328808-B21	1		4,301	48	206,438	
NC3122 PCI Dual 10/100 Controller	317450-B21	1		263	12	3,158	
Compaq SMART Array Controller 3504 - 4 SCSI Chan.	158939-B21	1		2,463	60	147,773	
StorageWorks Enclosure Model 4214R	103381-001	1		2,912	156	454,272	
Dual I/O Module for StorageWorks 4214R	119829-B21	1		560	12	6,720	
Compaq V500 Color Monitor	325900-001	1		172	12	2,070	
4/8-GB SLR Tape Drive - Internal	295480-B22	1		448	1	448	
Compaq Rack Model 7142	165753-001	1		1,680	15	25,200	
Compaq Rack Coupling Kit	165664-001	1		152	14	2,132	
Compaq Rack SideWall Kit	165652-001	1		208	1	208	
R3000 UPS	242705-001	1		1,786	14	25,010	
ServerNet II PCI Adapter	422869-001	1		890	12	10,685	
ServerNet II 12 Port Switch	452219-001	1		4,984	2	9,968	
LinkSys 16 Port 10/100 Switch DSSX16	DEH4324 Linksys	3		469	9	4,221	See Note 1
9.1 GB Hot-Plug Wide U2 10K 1"	328939-B22	1		469	2064	968,594	
18.2 GB Hot-Plug Wide U2 10K 1"	128418-B22	1		783	96	75,156	
Proliant Enterprise Server - Parts Exchange - 1 Year	FM-HIEPRT-12	1		597	24		14,328
Proliant Storage System - Parts Exchange - 1 Year	FM-STPRT-12	1		255	24		6,120
Onsite 5x8 4 Hr Service Upgrade from Compaq Standard Service		Amburst			1		
Server Software							
Microsoft SQL Server 2000 Enterprise	810-00945 Microsoft	2		15,802	96	1,516,992	230,628
Microsoft Visual C++ 6.0	048-00317 Microsoft	2		549	1	549	125,700
Microsoft Windows 2000 Advance Server	C10-00475 Microsoft	2		2,399	12	28,788	Incl Above
Client Hardware							
Proliant 1600 6/600 - 128MB 100TX Nic	153552-001	1		2,688	17	45,696	
Pentium III 600 MHz Processor Option	153555-B21	1		1,007	17	17,117	
Proliant 1850R 6/550 128MB 100TX Nic	123740-001	1		4,032	6	24,192	
Proliant DL380 733/133 128MB TX NIC	157829-001	1		4,592	13	59,696	
128-Megabyte SD DIMM Kit	313615-B21	1		361	69	24,884	
128 MB 133 DIMM	128277-B21	1		311	39	12,143	
Pentium III 550 512KB Processor	153555-B21	1		899	6	5,394	
Pentium III 733/133 Processor	159756-B21	1		783	13	10,177	
NC3122 PCI Dual 10/100 Controller	317450-B21	1		263	48	12,634	
Compaq V500 Color Monitor	325900-001	1		172	36	6,209	
9.1GB Wide U2 10K	328939-B22	1		469	36	16,894	
Proliant Workgroup Server - Parts Exchange - 1 Year	FM-HLOPRT-12	1	Amburst	259	72		18,648
Onsite 5x8 4 Hr Service Upgrade from Compaq Standard Service					1		18,803
Client Software							
Microsoft Windows 2000 Server	C11-00821 Microsoft	2		738	36	26,568	Incl. Above
User Connectivity							
LinkSys 8 Port 10/100 Switch EZX488R	DEH4162 LinkSys	3		159	40	6,360	See Note 1
ARK 17-port 16-10BaseT 1-10Base2 (BNC) hub	CT1017D1 ArkPC	4		37.00	14890	550,930	See Note 1
Large Purchase and Cash discount	9.0%	Amburst	1				
Total						\$4,917,311	\$388,260
						(\$330,766)	(\$25,967)
Subtotal						557,290	0
Five-Year Cost of Ownership:							\$5,305,571
tpmC Rating:							262,243.60
							\$ / tpmC: \$20.24

Note 1 = 5 Year warranty with 10% Spares -

Note: The Benchmark results and test methodology were audited by Lorna Livingtree and Tom Sawyer of Performance Metrics, Inc.

Numerical Quantities Summary

MQTH, Computed Maximum Qualified Throughput 262,243.60 tpmC

% throughput difference, reported & reproducibility runs 0.4%

Response Times (in seconds)	Average	90%	Maximum
New-Order	0.87	1.66	26.49
Payment	0.70	1.46	21.26
Order-Status	0.67	1.40	13.53
Delivery (interactive portion)	0.11	0.11	5.30
Delivery (deferred portion)	0.60	0.93	15.44
Stock-Level	1.61	2.49	14.66
Menu	0.11	0.11	5.53

Transaction Mix, in percent of total transaction

New-Order	44.82%
Payment	43.06%
Order-Status	4.03%
Delivery	4.05%
Stock-Level	4.04%

Emulation Delay (in seconds)

	Resp. Time	Menu
New-Order	0.10	0.10
Payment	0.10	0.10
Order-Status	0.10	0.10
Delivery (interactive)	0.10	0.10
Stock-Level	0.10	0.10

Keying/Think Times (in seconds)

	Min.	Average	Max.
New-Order	18.00/0.00	18.02/12.28	18.04/126.05
Payment	3.00/0.00	3.02/12.29	3.04/122.92
Order-Status	2.00/0.00	2.02/10.23	2.03/102.51
Delivery (interactive)	2.00/0.00	2.02/5.16	2.03/51.51
Stock-Level	2.00/0.00	2.02/5.15	2.03/51.51

Test Duration

Ramp-up time	74 minutes
Measurement interval	30 minutes
Transactions (all types) completed during measurement interval	17,551,635
Ramp down time	7 minutes

Checkpointing

Number of checkpoints	1
Checkpoint interval	30 minutes

General Items

Test Sponsor

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

This benchmark was sponsored by Compaq Computer Corporation. The benchmark was developed and engineered by Compaq Computer Corporation. Testing took place at Compaq benchmarking laboratories in Houston, Texas.

Application Code and Definition Statements

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

Appendix A contains all source code implemented in this benchmark.

Parameter Settings

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

- Database options
- Recover/commit options
- Consistency locking options
- Operating system and application configuration parameters

This requirement can be satisfied by providing a full list of all parameters.

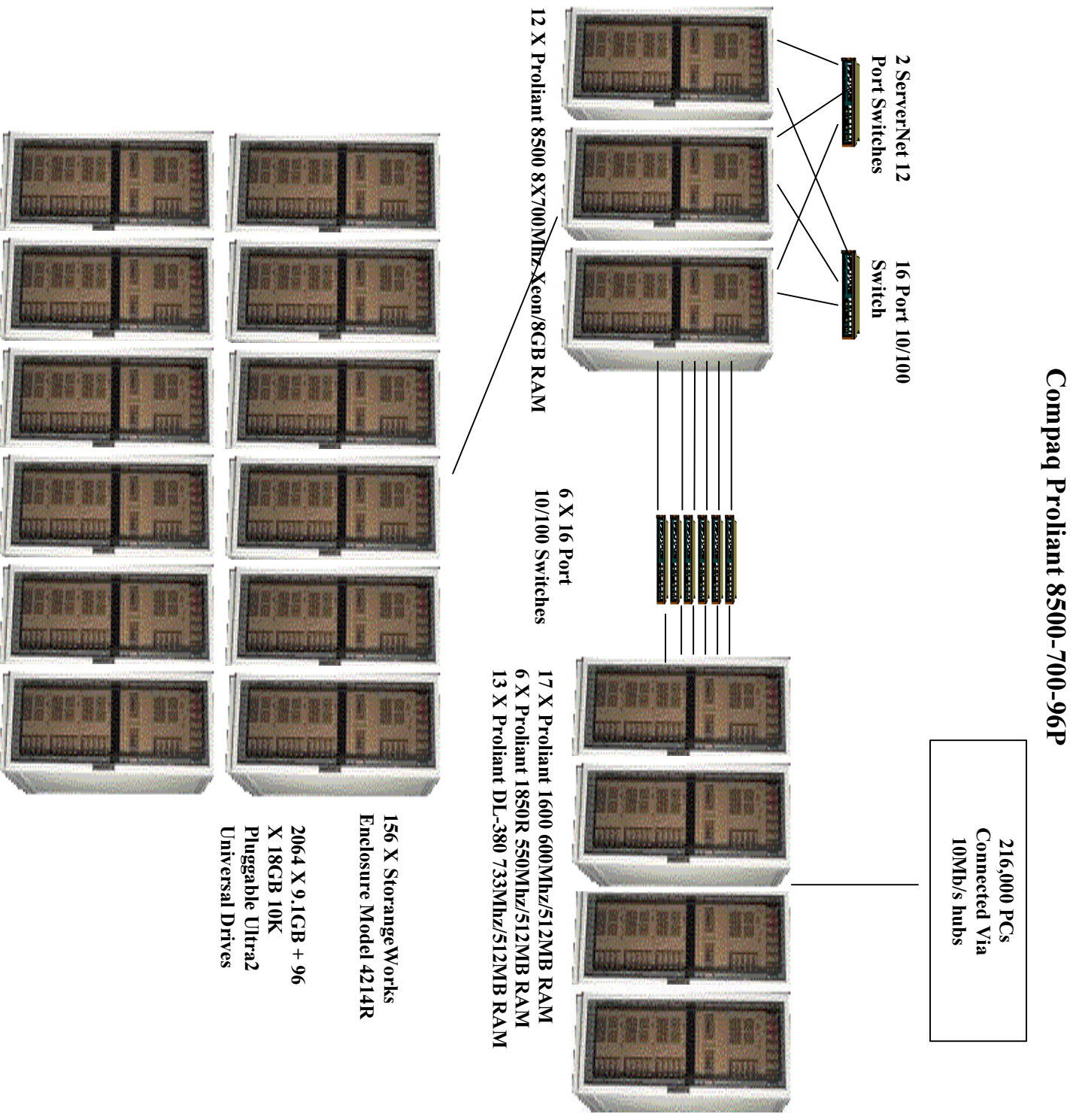
Appendix C contains the tunable parameters for the database, the operating system, and the transaction monitor.

Configuration Items

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

The configuration diagram for the tested and priced systems is included on the following page.

Figure 1. Benchmarked and Priced Configuration



Clause 1 Related Items

Table Definitions

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

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

Physical Organization of Database

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

The tested configuration consisted of: 2064 drives at 9.1GB and 96 drives at 18GB each.

Benchmarked Configuration (on each node):

Embedded Smart Array Controller

(2 9.1GB drives)		
<u>EISA UTILITIES PARTITION</u>	<u>Total Capacity = 36 MB</u>	<u>RAID 1</u>
Compaq System Configuration Utilities		
<u>LOGICAL DRIVE C:</u>	<u>Total Capacity = 8.43 GB</u>	<u>RAID 1</u>
Microsoft Windows 2000 Advanced Server		

SMART-5304 Controller, Slot 7, Array		
(42 9.1GB drives)		
(0% Read Cache, 100% Write Cache)		
<u>LOGICAL DRIVE 0:</u>	<u>Total Capacity = 29.83 GB</u>	<u>RAID 0</u> Cache Enabled
<u>Big_fg</u>		
<u>LOGICAL DRIVE 1:</u>	<u>Total Capacity = 24.41 GB</u>	<u>RAID 0</u> Cache Enabled
<u>Misc_fg</u>		
<u>LOGICAL DRIVE 2:</u>	<u>Total Capacity = 150.83 GB</u>	<u>RAID 0+1</u> Cache Enabled
TpcBackup3		

SMART-5304 Controller, Slot 8, Array A		
(42 9.1GB drives)		
(0% Read Cache, 100% Write Cache)		
<u>LOGICAL DRIVE 3:</u>	<u>Total Capacity = 29.83 GB</u>	<u>RAID 0</u> Cache Enabled
<u>Big_fg</u>		
<u>LOGICAL DRIVE 4:</u>	<u>Total Capacity = 24.41 GB</u>	<u>RAID 0</u> Cache Enabled
<u>Misc_fg</u>		
<u>LOGICAL DRIVE 5:</u>	<u>Total Capacity = 150.83 GB</u>	<u>RAID 0+1</u> Cache Enabled
TpcBackup4		

SMART-5304 Controller, Slot 9, Array A		
(8 18.2GB drives)		
<u>LOGICAL DRIVE 6:</u>	<u>Total Capacity = 67.83 GB</u>	<u>RAID 1</u> Cache Disabled
Tpc log		
SMART-5304 Controller, Slot 9, Array B		
(2 9.1GB drives)		
(50% Read Cache, 50% Write Cache)		
<u>LOGICAL DRIVE 7:</u>	<u>Total Capacity = 8.47 GB</u>	<u>RAID 1</u> Cache Enabled
Root database files, DTC logs		

SMART-5304 Controller, Slot 3, Array A

(42 9.1GB drives)		
(0% Read Cache, 100% Write Cache)		
<u>LOGICAL DRIVE 8:</u>	<u>Total Capacity = 29.83 GB</u>	<u>RAID 0</u> Cache Enabled
Big_fg	<u>Total Capacity = 24.41 GB</u>	<u>RAID 0</u> Cache Enabled
<u>LOGICAL DRIVE 9:</u>		
Misc_fg	<u>Total Capacity = 150.83 GB</u>	<u>RAID 1</u> Cache Enabled
<u>LOGICAL DRIVE 10:</u>		
TpccBackup1		

SMART-5304 Controller, Slot 4, Array A

(42 9.1GB drives)		
(0% Read Cache, 100% Write Cache)		
<u>LOGICAL DRIVE 11:</u>	<u>Total Capacity = 29.83 GB</u>	<u>RAID 0</u> Cache Enabled
Big_fg	<u>Total Capacity = 24.41 GB</u>	<u>RAID 0</u> Cache Enabled
<u>LOGICAL DRIVE 12:</u>		
Misc_fg	<u>Total Capacity = 150.83 GB</u>	<u>RAID 1</u> Cache Enabled
<u>LOGICAL DRIVE 13:</u>		
TpccBackup2		

Priced Configuration vs. Measured Configuration:

The measured and priced configuration only differ in that the measured configuration used disk drives for database backup and the priced configuration used a DAT drive for backup.

Insert and Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restrictions in the SUT database implementation that precludes inserts beyond the limits defined in Clause I.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the minimum key value for these new rows.

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

Partitioning

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

All database tables were partitioned across database server nodes by warehouse ID, 1800 warehouses per node with the exception of the item table, which was replicated across nodes.

Replication, Duplication or Additions

Replication of tables, if used, must be disclosed. Additional and/or duplicated attributes in any table must be disclosed along with a statement on the impact on performance.

The item table was replicated across all database server nodes.

Clause 2 Related Items

Random Number Generation

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

In the Benchmark RTE from Microsoft, each driver engine uses an independent random number sequence. All of the users within a given driver draw from the same sequence.

The Benchmark RTE computes random integers as described in "Random Numbers Generators: Good Ones Are Hard to Find." Communications of the ACM - October 1988 Volume 31 Number 10.

The seeds for each user were captured and verified by the auditor to be unique. In addition, the contents of the database were systematically searched, and randomly sampled by the auditor for patterns that would indicate the random number generator had effected any kind of a discernible pattern; none were found.

Input/Output Screen Layout

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

All screen layouts followed the specifications exactly.

Priced 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).

The terminal attributes were verified by the auditor manually exercising each specification on a representative Compaq Proliant 1600 server.

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 forms applications is listed in Appendix A.

Transaction Statistics

Table 2.1 lists the numerical quantities that Clauses 8.1.3.5 to 8.1.3.11 require.

Table 2. 1 Transaction Statistics

Statistic	Value	
New Order	Home warehouse order lines	99.00%
	Remote warehouse order lines	1.00%
	Rolled back transactions	1.00%
Payment	Average items per order	10.00
	Home warehouse payments Remote warehouse payments	85.00% 15.00%

Statistic	Value
Accessed by last name	60.01%
Order Status	59.98%
Transaction Mix	44.82% 43.06% 4.03% 4.05% 4.04%
	New Order Payment Order status Delivery Stock level

Queuing Mechanism

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

The deferred delivery operation is queued by making an entry in an array within the application process (tpcc.dll) running on the middle tier machines. Background threads within the application process asynchronously process the queued delivery transactions.

The source code is listed in Appendix A.

Clause 3 Related Items

Transaction System Properties (ACID)

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

All ACID property tests were successful. The executions are described below.

Atomicity

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

Completed Transactions

A row was selected in a script from the warehouse, district and customer tables, and the balances noted. A payment transaction was started with the same warehouse, district and customer identifiers and a known amount. The payment transaction was committed and the rows were verified to contain correctly updated balances.

Aborted Transactions

A row was selected in a script from the warehouse, district and customer tables, and the balances noted. A payment transaction was started with the same warehouse, district and customer identifiers and a known amount. The payment transaction was rolled back and the rows were verified to contain the original balances.

All atomicity tests were run twice. The first set of atomicity tests were run with database accesses from just a single node. The second set of atomicity tests were run with database accesses from two nodes.

Consistency

Consistency is the property of the application that requires any execution of a database transaction to take the database from one consistent state to another, assuming that the database is initially in a consistent state.

Consistency conditions one through four were tested using a shell script to issue queries to the database. The results of the queries verified that the database was consistent for all four tests.

A run was executed under full load lasting over an hour and included a checkpoint.

The shell script was executed again. The result of the same queries verified that the database remained consistent after the run.

Isolation

Sufficient conditions must be enabled at either the system or application level to ensure the required isolation defined above (clause 3.4.1) is obtained.

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

In addition, the phantom tests and the stock level tests were executed and verified.

For Isolation test seven, case A was followed.

All isolation tests were run twice. The first set of isolation tests were run with database accesses from just a single node. The second set of isolation tests were run with database accesses from two nodes.

Durability

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

Durable Media Failure

Durability from media failure was demonstrated on a database scaled for 10 warehouses running on 2 of the Proliant 8500 nodes (5 warehouses on each). The standard driving mechanism was used to generate the transaction load of 100 users. The fully scaled database under full load would also have passed the following test. Loss of data and loss of log were combined in the same test.

Loss of Data and Log

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

- The database was backed up to extra disks.
- The total number of New Orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.
- The RTE was started with 100 users.
- The test was allowed to run for a minimum of 10 minutes.
- One log disk was removed from the drive cabinet.
- Since the disk was mirrored, processing was not interrupted. This was verified by checking the users status on the RTE.
- One of the data disks was removed from the drive cabinet.
- When Microsoft SQL Server recorded errors about not being able to access the database, the RTE was shut down.
- A dump of the transaction log was taken and the Microsoft SQL Server was shutdown.
- A new log disk was inserted into the log drive cabinet. A new data disk was inserted into the data drive cabinet. Both cabinets were powered back up and system restarted.
- Microsoft SQL Server was started.
- The database was automatically recovered from the database backup and the transaction log dump.
- Consistency condition #3 was executed and verified.
- Step 2 was repeated and the difference between the first and second counts was noted.
- An RTE report was generated for the entire run time giving the number of NEW-ORDERS successfully returned to the RTE.
- The counts in step 14 and 15 were compared and the results verified that all committed transactions had been successfully recovered.
- Samples were taken from the RTE files and used to query the database to demonstrate successful transactions had corresponding rows in the ORDER table.

Instantaneous Interruption and Loss of Memory

Because loss of power erases the contents of memory, the instantaneous interruption and the loss of memory tests were combined into a single test. This test was performed twice, once for a single node and second time for 12 nodes. Both tests were executed on a fully scaled database of 21,600 warehouses under a full load of 216,000 users. The following steps were executed for each test:

- The total number of New Orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table on all database nodes giving the beginning count.
- The RTE was started with 216,000 users.
- The test was allowed to run for a minimum of 10 minutes.
- A checkpoint was performed.
- For the single node and 12 node tests, system crash and loss of memory were induced by switching the power off. In the single node test all nodes were executing transactions,

- but only one node was powered off. In the 12 node test all nodes were powered off. No battery backup or Uninterruptible Power Supply (UPS) were used to preserve the contents of memory.
- The RTE was shutdown.
 - All power was restored and the system restarted.
 - Microsoft SQL Server was restarted and performed an automatic recovery.
 - Consistency condition #3 was executed and verified.
 - Step 1 was repeated and the difference between the first and second counts was noted.
 - An RTE report was generated for the entire run time giving the number of NEW-ORDERS successfully returned to the RTE.
 - The counts in step 10 and 11 were compared and the results verified that all committed transactions had been successfully recovered.
 - Samples were taken from the RTE files and used to query the database to demonstrate successful transactions had corresponding rows in the ORDER table.

Interconnect Loss

Since the benchmark utilized two different types of internode interconnects, two separate interconnect loss tests were performed. The first test involved power loss to a ServerNetII switch. The second test involved power loss to the 10/100 ethernet switch. Both tests were executed on a fully scaled database of 21,600 warehouses under a full load of 216,000 users. The following steps were executed for each test:

- The total number of New Orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table on all database nodes giving the beginning count.
- The RTE was started with 216,000 users.
- The test was allowed to run for a minimum of 10 minutes.
- A checkpoint was performed.
- For the ServerNetII loss test a ServerNetII switch was powered off during the run. For the ethernet switch loss test the power cord was pulled from the switch during the run.
- The RTE was shutdown.
- All power was restored and the system restarted.
- Microsoft SQL Server was restarted and performed an automatic recovery.
- Consistency condition #3 was executed and verified.
- Step 1 was repeated and the difference between the first and second counts was noted.
- An RTE report was generated for the entire run time giving the number of NEW-ORDERS successfully returned to the RTE.
- The counts in step 10 and 11 were compared and the results verified that all committed transactions had been successfully recovered.
- Samples were taken from the RTE files and used to query the database to demonstrate successful transactions had corresponding rows in the ORDER table.

Clause 4 Related Items

Initial Cardinality of Tables

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

Table 4.1 Number of Rows for Database

Table	Cardinality as built
Warehouse	21600
District	216,000
Customer	648,000,000
History	648,000,000
Orders	648,000,000
New Order	194,400,000
Order Line	6,479,903,522
Stock	2,160,000,000
Item	100,000
Deleted Warehouses	0

Database Layout

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

The benchmarked configuration used 12 Compaq Proliant 8500 nodes each configured identically with 1800 warehouses each. Each node used 5 SMART-5304 Array controllers, each with 4 SCSI channels. Each controller is capable of accessing up to 56 disk drives per array, 14 disk drives per each channel, and supports RAID 0, 1, 5 per each logical volume configured. On each node, the data tables were stored on 8 logical volumes of 42 drives each (i.e. 2 logical volumes on each controller array of 42 drives). Each data volume was configured with RAID 0 and the Array Accelerator was enabled for all the volumes. On each node, one logical volume of 8 18GB drives was configured as RAID 0+1 and stored the transaction log. The transaction log volumes had the Array Accelerator disabled. Also on each node, one logical volume of 2 9GB drives was configured as RAID 1 and stored the root database files, along with logs for the Distributed Transaction Coordinator (DTC). For each node, the operating system was stored on 2 9GB drives configured as RAID 1 on the embedded Smart Array Controller. All RAID volumes used hardware RAID.

Section 1.2 of this report details the distribution of database tables across all disks. The code that creates the filegroups and tables is included in Appendix B.

Type of Database

A statement must be provided that describes:

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

Microsoft SQL Server 2000 Enterprise Edition is a relational DBMS.

The interface used was Microsoft SQL Server stored procedures accessed with Remote Procedure Calls embedded in C code.

Database Mapping

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

The item table was replicated across database nodes.

180 Day Space

Details of the 180 day space computations along with proof that the database is configured to sustain 8 hours of growth, for the dynamic tables (Order, Order-Line, and History) must be disclosed.

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

- The free space on the log file was queried using *dbcc sqlperf(logspace)*.
- The before New Order count was measured via the sql command: *select sum(d_next_o_id) SumDNextOID* from district
- Transactions were run against the database with a full load of users.
- The free space was again queried using *dbcc sqlperf(logspace)*.
- The after New Order count was measured via the sql command: *select sum(d_next_o_id) SumDNextOID* from district
- The space used was calculated as the difference between the first and second query.
- The number of NEW-ORDERS was calculated as the difference between the after and before *sum(d_next_o_id)* values covering the entire run.
- The space used was divided by the number of NEW-ORDERS giving a space used per NEW-ORDER transaction.
- The space used per transaction was multiplied by the measured tpmC rate times 480 minutes.

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

The details of both the 8-hour transaction log space requirement and the 180-day space requirement is shown in Appendix D.

Clause 5 Related Items

Throughput

Measured tpmC must be reported

Measured tpmC 262,243.60 tpmC
Price per tpmC \$20.24 per tpmC

Response Times

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

Table 5.2: Response Times

Type	Average	90 th %	Maximum
New-Order	0.87	1.66	26.49
Payment	0.70	1.46	21.26
Order-Status	0.67	1.40	13.53
Interactive Delivery	0.11	0.11	5.30
Deferred Delivery	0.60	0.93	15.44
Stock-Level	1.61	2.49	14.66
Menu	0.11	0.11	5.53

Keying and Think Times

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

Table 5.3: Keying Times

Type	Minimum	Average	Maximum
New-Order	18.00	18.02	18.04
Payment	3.00	3.02	3.04
Order-Status	2.00	2.02	2.03
Interactive Delivery	2.00	2.02	2.03
Stock-Level	2.00	2.02	2.03

Table 5.4: Think Times

Type	Minimum	Average	Maximum
New-Order	0.00	12.28	126.05
Payment	0.00	12.29	122.92
Order-Status	0.00	10.23	102.51
Interactive Delivery	0.00	5.16	51.51
Stock-Level	0.00	5.15	51.51

Response Time Frequency Distribution Curves and Other Graphs

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

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

Think Time frequency distribution curves (see Clause 5.6.3) must be reported for each transaction type.

Keying Time frequency distribution curves (see Clause 5.6.4) must be reported for each transaction type.

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

Figure 3. New Order Response Time Distribution

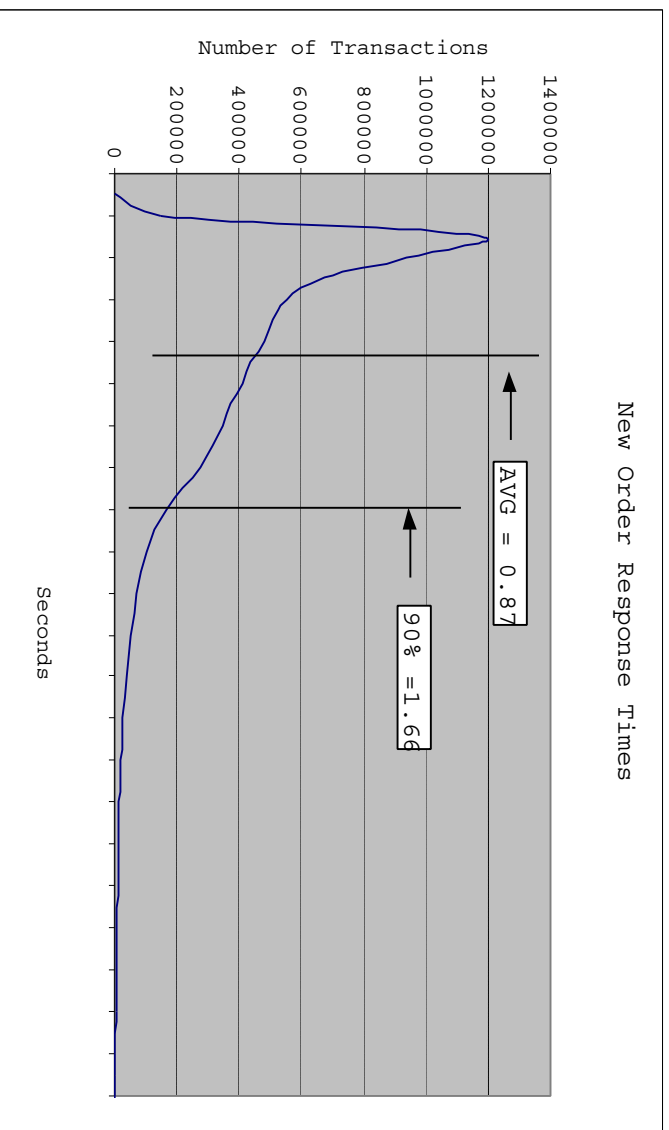


Figure 4. Payment Response Time Distribution

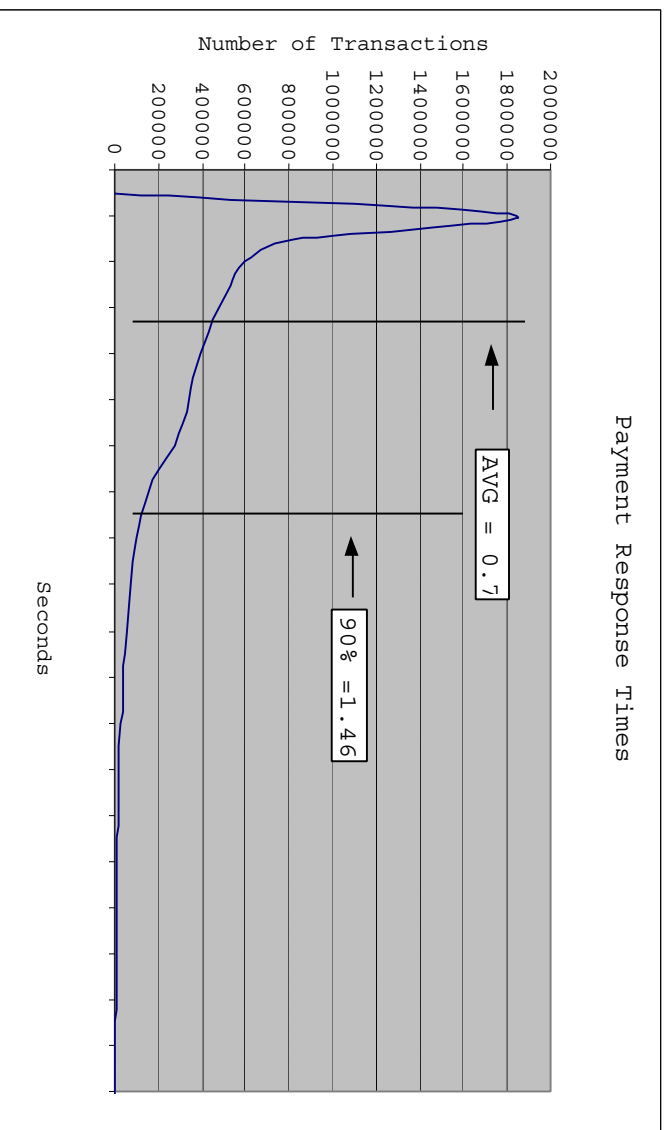


Figure 5. Order Status Response Time Distribution

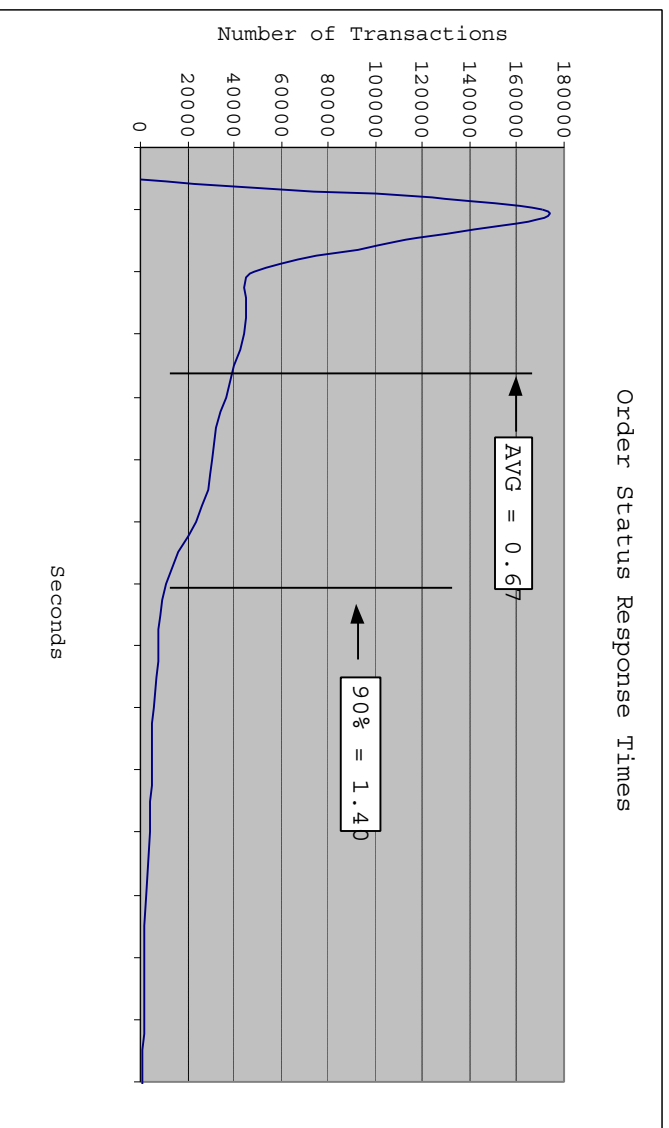


Figure 6. Delivery Response Time Distribution

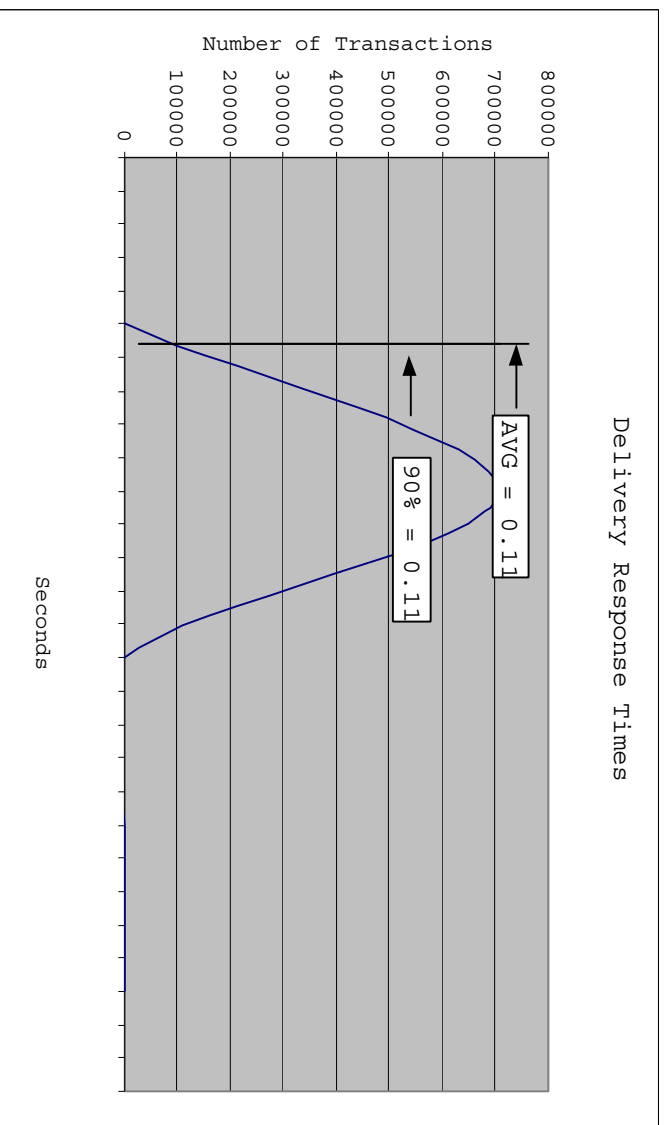


Figure 7. Stock Level Response Time Distribution

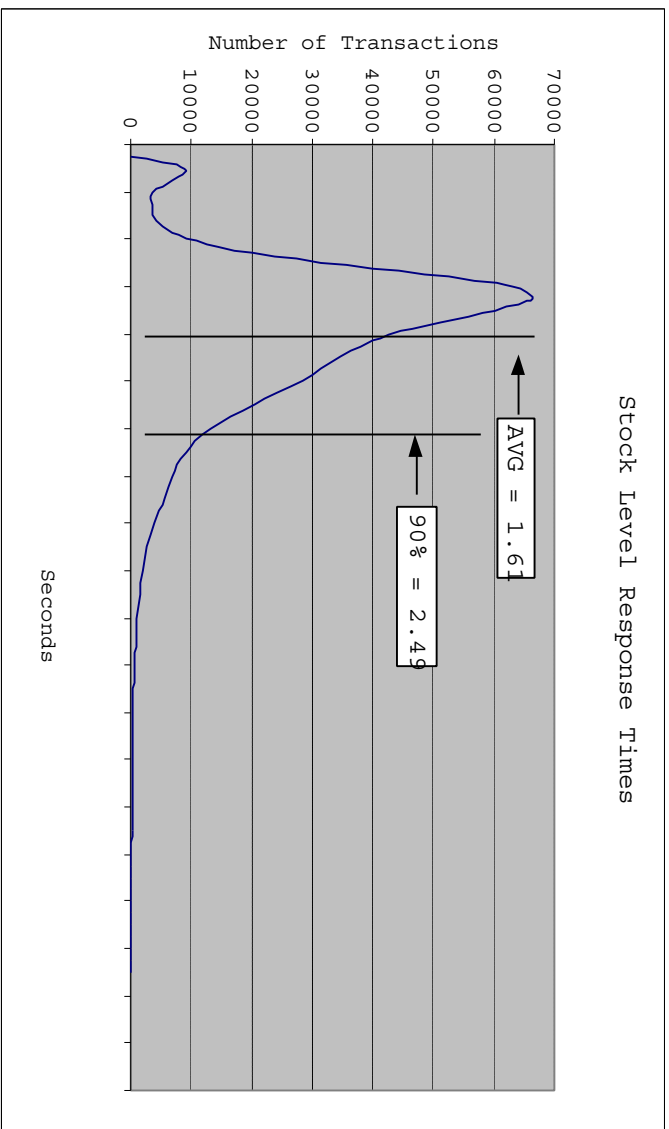


Figure 8. Response Time vs. Throughput

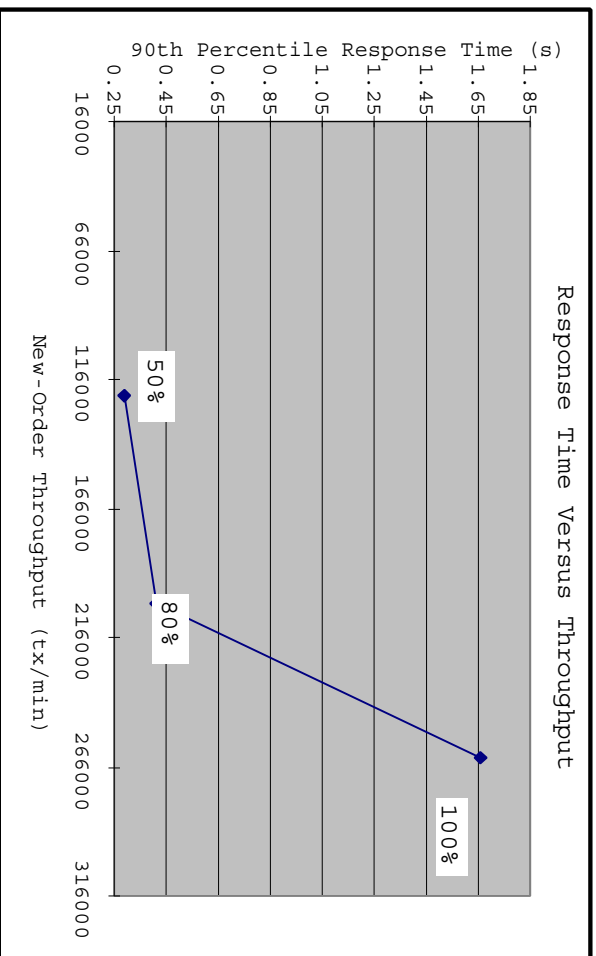


Figure 9. New Order Think Time Distribution

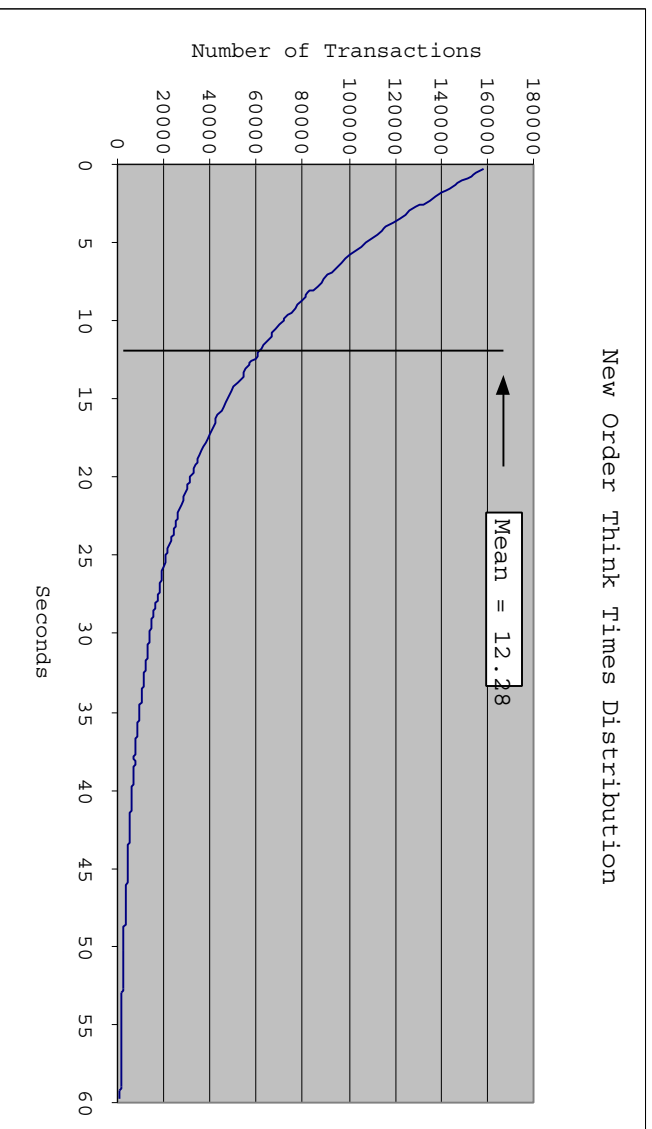
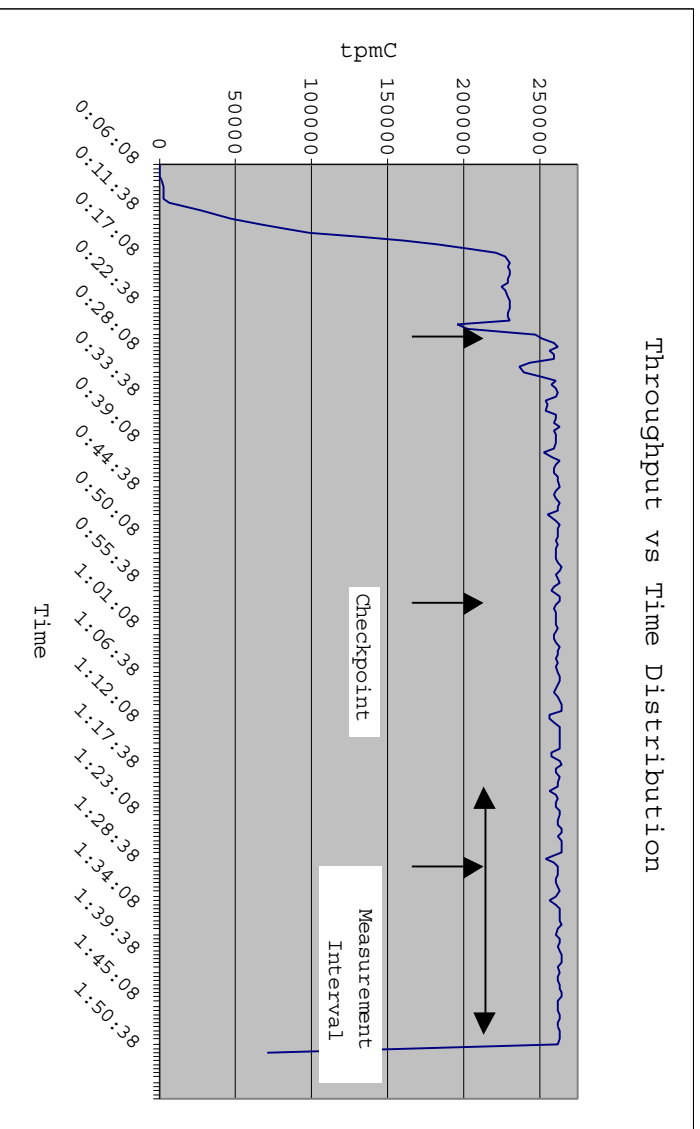


Figure 10. Throughput vs. Time Distribution



Steady State Determination

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

Steady state was determined using real time monitor utilities from the RTE. Steady state was further confirmed by the throughput data collected during the run and graphed in Figure 10.

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.

The RTE generated the required input data to choose a transaction from the menu. This data was timestamped. The input screen for the requested transaction was returned and timestamped. The difference between these two timestamps was the menu response time. The RTE writes to the log file once per transaction on selective fields such as order id. There is one log file per driver engine.

The RTE generated the required input data for the chosen transaction. It waited to complete the minimum required key time before transmitting the input screen. The transmission was timestamped. The return of the screen with the required response data was timestamped. The difference between these two timestamps was the response time for that transaction.

The RTE then waited the required think time interval before repeating the process starting at selecting a transaction from the menu.

The RTE transmissions were sent to application processes running on the client machines through Ethernet LANs. These client application processes handled all screen I/O as well as all requests to the database on the server. The applications communicated with the database server over another Ethernet LAN using ODBC and RPC calls.

To perform checkpoints at specific intervals, we set SQL Server *recovery interval* to 40 and wrote a script to schedule multiple checkpoints at specific intervals. The script included a wait time between each checkpoint equal to the measurement interval which was 30 minutes. The checkpoint script was started manually after the RTE had all users logged in and sending transactions. These manual checkpoints kept SQL Server's automatic checkpoints postponed as long as the script was running.

At each checkpoint, Microsoft SQL Server wrote to disk all memory pages that had been updated but not yet physically written to disk. The positioning of the measurement interval was verified to be clear of the guard zones and is depicted on the graph in Figure 10.

Reproducibility

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

We allowed the database to warm up and to reach a steady state for approximately 43 minutes. The steady state was sustained for a 30-minute (measurement) interval, and was followed by a second measurement interval and then a ramp-down. The repeatable interval result was within 0.4 % of the reported interval result.

Measurement Period Duration

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

The reported measured interval was exactly 30 minutes long.

Regulation of Transaction Mix

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

The RTE was given a weighted random distribution, which was not adjusted during the run.

Transaction Statistics

The percentage of the total mix for each transaction type must be disclosed. The percentage of New-Order transactions rolled back as a result of invalid item number must be disclosed. The average number of order-lines entered per New-Order transaction must be disclosed. The percentage of remote Payment transactions must be per New-Order transaction 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.

Table 5.5: Transaction Statistics

Statistic	Value	
New Order	Home warehouse order lines	99.00%
	Remote warehouse order lines	1.00%
	Rolled back transactions	1.00%
Average items per order	10.00	
Payment	Home warehouse payments	85.00%
	Remote warehouse payments	15.00%
Accessed by last name	60.01%	
Order Status	Accessed by last name	59.98%
Transaction Mix	New Order	44.82%
	Payment	43.06%
	Order status	4.03%
	Delivery	4.05%
Stock level	4.04%	

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.

The initial checkpoint was started 24 minutes after the start of the ramp-up. Subsequent checkpoints occurred every 30 minutes. The checkpoint in the measurement interval lasted approximately 10 minutes. The measurement interval contains the third checkpoint, and is clear of the guard zones.

Clause 6 Related Items

RTE Descriptions

If the RTE is commercially available, then its inputs must be specified. Otherwise, a description must be supplied of what inputs (e.g., scripts) to the RTE had been used.

The RTE used was Microsoft Benchmark RTE. Benchmark is a proprietary tool provided by Microsoft and is not commercially available. The RTE's input are listed in Appendix A.

Emulated Components

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

The driver system consisted of 12 Compaq ProLiant servers. These driver machines were attached to the 36 Compaq ProLiant client machines through a 100Mb/s Ethernet connection to the 10/100 Dual Speed Switches. Since this configuration is the same connectivity of the priced system, no components were being emulated. Therefore, the test described in Clause 6.6.3.4 was not required.

Functional Diagrams

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

The driver system performed the data generation and input functions of the priced display device. It also captured the input and output data and timestamps for post-processing of the reported metrics. No other functionality was included on the driver system.

Section 1.4 of this report contains detailed diagrams of both the benchmark configuration and the priced configuration.

Networks

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

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

In the tested configuration, 12 driver (RTE) machines were connected to the 8-port Dual Speed switches with 100Mb/s Ethernet connections. The 36 client machines were connected to the switch with 100Mb/s connections. The server (SUT) was connected to the switches with a 100Mb/s connection.

A test was run demonstrating that using the 100Mb/s connection from the RTE machines to drive the clients had no enhancement in performance. This was accomplished by comparing TPC-C performance results in both the 100Mb/s and multiple 10Mb/s configurations. This test was accomplished with an auditor present and monitoring the testing procedures.

Operator Intervention

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

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

Clause 7 Related Items

System Pricing

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

The total 5 year price of the entire configuration must be reported, including: hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

The details of the hardware and software are reported in the front of this report as part of the executive summary. All third party quotations are included at the end of this report as Appendix E.

Availability, Throughput, and Price Performance

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

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

- **Maximum Qualified Throughput** **262,243.60 tpmC**
- **Price per tpmC** **\$20.24 per tpmC**
- **Availability** **September 30, 2000**

Country Specific Pricing

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

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

Usage Pricing

For any usage pricing, the sponsor must disclose:

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

The component pricing based on usage is shown below:

- 12 Microsoft Windows 2000 Advanced Server
- 36 Microsoft Windows 2000 Server licenses (includes Microsoft COM+)
- 12 Microsoft SQL Server 2000 Enterprise Edition (Unlimited User Licenses)
- One Microsoft Visual C++
- Compaq Servers include 3 years of support.

Clause 9 Related Items

Auditor's Report

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

This implementation of the TPC Benchmark C was audited by Lorna Livingtree and Tom Sawyer of Performance Metrics, Inc.

Performance Metrics, Inc.
137 Yankton St., Suite 101
Folsom, CA 95630
(phone) (916) 985-1131
(fax) (916) 985-1185
e-mail: lorna@perfmetrics.com

Availability of the Full Disclosure Report

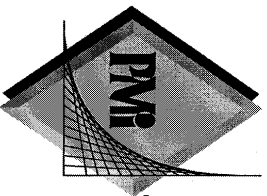
The Full Disclosure Report must be readily available to the public at a reasonable charge, similar to the charges for similar documents by the test sponsor. The report must be made available when results are made public. In order to use the phrase "TPC Benchmark™ C", the Full Disclosure Report must have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

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

Transaction Processing Performance Council
c/o Shanley Public Relations
777 North First Street, Suite 600
San Jose, CA 95112-6311

or

Compaq Computer Corporation
Database Performance Engineering
P.O. Box 692000
Houston, TX 77269-2000



PERFORMANCE METRICS INC.
TPC Certified Auditors

July 22, 2000

Mr. Andy Bond
Mr. Paul Cao
Senior Systems Software Engineers
Compaq Computer Corporation
20555 SH 249
Houston, TX 77070

I have verified the TPC Benchmark™ C client/server for the following configuration

Platform: ProLiant 8500-700-96P (12 node cluster)
Database Manager: Microsoft SQL Server 2000 Version 8 Enterprise Edition
Operating System: Microsoft Windows 2000 Advanced Server
Transaction Monitor: Microsoft COM+

12 Servers: ProLiant 8500 each with:				
CPU's	Memory	Disks (total)	90% Response	TpmsC
8 Pentium III Xeon @ 700 Mhz	Main: 8192MB Cache: 2 MB	172 @ 9.1GB 8 @ 18.2 GB	1.66 sec	262,243.60
36 Clients of three types: 13 ProLiant DL380s each with:				
2 Pentium III @ 733 MHz	Main: 512 MB Cache: 256K	1 @ 9.1 GB	na	Na
6 ProLiant 1850Rs				
2 Pentium III @ 550 MHz	Main: 512 MB Cache: 512	1 @ 9.1 GB	na	na
17 ProLiant 1600s				
2 Pentium III @ 600 MHz	Main: 512 MB Cache: 512	1 @ 9.1 GB	na	na

137 Yankton St. Suite 101, Folsom, CA 95630
(916) 985-1131 fax: (916) 985-1185 email: Lorna@PerfMetrics.com

Page 1

PERFORMANCE METRICS INC.
TPC Certified Auditors

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

- The transactions were correctly implemented.
- The database files were properly sized and populated.
- The database was properly scaled with 21,600 warehouses evenly divided across all 12 nodes.
- The ACID properties were successfully demonstrated. Two nodes were used to successfully demonstrate ACID properties for remote cross-node transactions.
- The update of all columns, of all tables was successfully demonstrated on the 8-node 550 MHz configuration.
- The durability tests for data loss and log loss were demonstrated on a subset of the SUT configured with a database properly populated for 10 warehouses and evenly divided across 2 nodes.
- Input data was generated according to the specified percentages.
- Eight hours of mirrored log space was present on the tested system.
- Eight hours of growth space for the dynamic tables was present on the tested system.
- The data for the 180-day space calculation was verified.
- The controller cache was disabled on the log disk controllers.
- The steady state portion of the test was 30 minutes.
- One checkpoint was taken before the measured interval.
- One checkpoint was taken during the measured interval.
- The checkpoints were verified to be clear of the guard zone.
- The system pricing was checked for major components and maintenance.
- Third party quotes were verified for compliance.

Auditor Notes:

The cluster was tested for durability of transaction against a single node failure, a 12-node failure and a node interconnect failure. All durability tests completed successfully.

Portions of the audit were performed by Tom Sawyer of Performance Metrics Inc.

Sincerely,



Lorna Livingtree
Auditor

137 Yankton St. Suite 101, Folsom, CA 95630
(916) 985-1131 fax: (916) 985-1185 email: Lorna@PerfMetrics.com

Page 2

Appendix A: Source Code

The client source code is listed below.

Methods.h

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

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

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

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

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

```
        if (m_szErrorText != NULL)
            delete [] m_szErrorText;
    };

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

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

static void WriteMessageToEventLog(LPTSTR lpszMsg);

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

    CTPCC_Common();
    ~CTPCC_Common();

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

    HRESULT __stdcall CallSetComplete();

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

// IObjectConstruct
    STDMETHODIMP Construct(IDispatch * pUnk);

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

    struct COM_DATA
    {
```

```

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

};

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

    BEGIN_COM_MAP(CTPCC)
        COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
    END_COM_MAP()

};

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

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

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

////////////////////////////////////
// COrderStatus
class COrderStatus :
    public CTPCC_Common,

```

```

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

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

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

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

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

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

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

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

```

```

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

```

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 ( !strcmp(szTmp, szDBNames[ODBC]) )
            pReg->eDB_Protocol = ODBC;
        else if ( !strcmp(szTmp, szDBNames[DBLIB]) )
            pReg->eDB_Protocol = DBLIB;
    }
}

```

```

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

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

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

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

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

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

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

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

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

```

```

        pReg->szDbUser[0] = 0;

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

        RegCloseKey(hKey);

        return FALSE;
    }

```

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 TXNMN { 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 TXNMN eTxnMon;
    BOOL bCOM_SinglePool;
    DWORD dwMaxConnections;
    DWORD dwMaxPendingDeliveries;
    DWORD dwNumberOfDeliveryThreads;
    char szPath[128];
    char szDbServer[32];
    char szDbName[32];
    char szDbUser[32];
    char szDbPassword[32];
} TPCCREGISTRYDATA, *PTPCCREGISTRYDATA;

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );

```

WEBCLNT.DSP

```

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

# TARGETTYPE "Win32 (x86) Application" 0x0101

CFG=webclnt - Win32 Release

```

```

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

```

```

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

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir ".\Release"
# PROP BASE Intermediate_Dir ".\Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\Release"
# PROP Intermediate_Dir ".\Release"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:I386

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir ".\Debug"
# PROP BASE Intermediate_Dir ".\Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\Debug"
# PROP Intermediate_Dir ".\Debug"

```

```

# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/c
# ADD CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD BASE MTL /nologo /D "_DEBUG" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD BASE RSC /1 0x409 /d "_DEBUG"
# ADD RSC /1 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386

!ENDIF

# Begin Target

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

```

Webclnt.dsw

Microsoft Developer Studio Workspace File, Format Version 6.00
WARNING: DO NOT EDIT OR DELETE THIS WORKSPACE FILE!

```

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

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

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

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

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

```

```

Package=<5>
{{{
}}}

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

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

Package=<5>
{{{
}}}

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

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

```

```

Package=<5>
{{{
}}}

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

#####

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

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

#####

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

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

#####

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

Package=<5>
{{{
}}}

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

#####

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

Package=<5>
{{{
}}}

Package=<4>
{{{

```

```

}}}

#####

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

Package=<5>
{{{
}}}

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

#####

Global:

Package=<5>
{{{
}}}

Package=<3>
{{{
}}}

#####

```

com_all_resource.h

```

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

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

```

db_dblib_dll.dsp

```

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

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

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

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

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ntwdlib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /machine:I386 /out:".bin\tpcc_dblib.dll"

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/ FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d " _DEBUG"
# ADD RSC /l 0x409 /d " _DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ntwdlib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_dblib.dll"
/pdbtype:sept

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

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

```

```

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/ FD /c
# ADD BASE MTL /nologo /D " _DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D " _DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d " _DEBUG"
# ADD RSC /l 0x409 /d " _DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ntwdlib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_dblib.dll"
/pdbtype:sept

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

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

```

```

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

!ENDIF

# Begin Target

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

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

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

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

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

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

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

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

```

db_odbc_dll.dsp

```

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

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

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

```

```

!MESSAGE NMAKE /f "db_odbc_dll.mak" CFG="db_odbc_dll - Win32 IceCAP"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "db_odbc_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_odbc_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_odbc_dll - Win32 IceCAP" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

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

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /1 0x409 /d "NDEBUG"
# ADD RSC /1 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386 /out:".bin\tpcc_odbc.dll"

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c

```

```

# ADD CPP /nologo /MDd /W3 /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD BASE MTL /nologo /D " _DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D " _DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /l 0x409 /d " _DEBUG"
# ADD RSC /l 0x409 /d " _DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".bin/tpcc_odbc.dll"
/pdbtype:sept

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "db_odbc"
# PROP BASE Intermediate_Dir "db_odbc_"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /Gm /GX /ZI /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"ICECAP" /YX /FD /Gh /c
# ADD BASE MTL /nologo /D " _DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D " _DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /l 0x409 /d " _DEBUG"
# ADD RSC /l 0x409 /d " _DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".bin/tpcc_odbc.dll"
/pdbtype:sept
# ADD LINK32 icap.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".bin/tpcc_odbc.dll"
/pdbtype:sept

!ENDIF

# Begin Target

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

# PROP Default_Filter "*.cpp"

```

```

# Begin Source File

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

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

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

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

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

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

```

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 */

```

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_WEBDDL 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_INS_MEMORY "Insufficient Memory to continue."
#define ERR_UNKNOWN "Unknown error."
#define ERR_MSG_BUF_SIZE 512
#define INV_ERROR_CODE -1

class CBaseErr
{
public:
    CBaseErr(LPCTSTR szLoc = NULL)
    {
        m_idMsg = INV_ERROR_CODE;
        if (szLoc)
        {
            m_szLoc = new char[m_szLoc_size];
            strcpy(m_szLoc, szLoc);
        }
        else
            m_szLoc = NULL;

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

```

```

}
CBaseErr(int idMsg, LPCTSTR szLoc = NULL)
{
    m_idMsg = idMsg;
    if (szLoc)
    {
        m_szLoc = new char[m_szLoc_size];
        strcpy(m_szLoc, szLoc);
    }
    else
        m_szLoc = NULL;

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

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

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

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

    j += wsprintf(szTmp+j, "%s\n", ErrorText());
    ::MessageBox(hwnd, szTmp, m_szApp, MB_OK);
}

char *GetApp(void) { return m_szApp; }
char *GetLocation(void) { return m_szLoc; }
virtual int ErrorNum() { return m_idMsg; }
virtual int ErrorType() = 0; // a value which distinguishes the kind of
error that occurred
virtual char *ErrorText() = 0; // a string (i.e., human readable)
representation of the error

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

class CSocketErr : public CBaseErr
{
public:
    enum Action
    {

```

```

        eNone,
        eSend,
        eSocket,
        eBind,
        eConnect,
        eListen,
        eHost,
        eRecv,
    };

    CSocketErr(Action eAction, LPCTSTR szLocation = NULL);
    Action m_eAction;

    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,
    };

    CSystemErr(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;}
};

```

install.c

```

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

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

#include "resource.h"

#define WM_INITTEXT WM_USER+100

HICON hIcon;
HINSTANCE hInst;

DWORD versionExeMS;
DWORD versionExeLS;
DWORD versionExeMM;
DWORD versionDllMS;
DWORD versionDllLS;

// TPC-C registry settings
TPCCREGISTRYDATA Reg;

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

static int iMaxPhysicalMemory; //max physical memory in
MB

static char szLastFileName[64]; // last file we worked on (for
error reporting)

BOOL CALLBACK LicenseDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM
lParam);

```

```

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

BOOL install_com(char *szDllPath);

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

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

    hInst = hInstance;

    InitCommonControls();

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

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

    DestroyIcon(hIcon);
    return 0;
}

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

    switch(uMsg)
    {
        case WM_INITDIALOG:
            hFont = CreateFont(-12, 0, 0, 0, 400, 0, 0, 0, 0, 0,
0, 0, 0, "Arial");
            SendMessage( GetDlgItem(hwnd, IDR_LICENSE1),
WM_SETFONT, (WPARAM)hFont, MAKELPARAM(0, 0) );

```

```

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

BOOL CALLBACK UpdatedDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    switch(uMsg)
    {
        case WM_INITDIALOG:
            switch(lParam)
            {
                case 1:
                case 2:
                    SetDlgItemText(hwnd, IDC_RESULTS,
"TPC-C Web Client Installed");
                    break;
            }
            return TRUE;
        case WM_COMMAND:
            if ( wParam == IDOK )
                EndDialog(hwnd, TRUE);
            break;
        default:
            break;
    }
    return FALSE;
}

BOOL CALLBACK MainDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    PAINTSTRUCT ps;
    MEMORYSTATUS memoryStatus;

```

```

OSVERSIONINFO VI;
char szTmp[256];
static char szDllPath[256];
static char szExePath[256];

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

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

        // set default values
        ZeroMemory( &Reg, sizeof(Reg) );
        Reg.dwNumberOfDeliveryThreads = 4;
        Reg.dwMaxConnections = 100;
        Reg.dwMaxPendingDeliveries = 100;
        Reg.eDB_Protocol = DBLIB;
        Reg.eTxnMon = None;
        strcpy(Reg.szDbServer, "");
        strcpy(Reg.szDbName, "tpcc");
        strcpy(Reg.szDbUser, "sa");
        strcpy(Reg.szDbPassword, "");

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

        ReadTPCCRegistrySettings( &Reg );
        ReadRegistrySettings();

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

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

        SetDlgItemText(hwnd, IDC_PATH, szDllPath);

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

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

```

```

iPoolThreadLimit, FALSE);
SetDlgItemInt(hwnd, ED_IIS_MAX_THREAD_POOL_LIMIT,
iThreadTimeout, FALSE);
SetDlgItemInt(hwnd, ED_IIS_THREAD_TIMEOUT,
iListenBackLog, FALSE);
SetDlgItemInt(hwnd, ED_IIS_LISTEN_BACKLOG,
iAcceptExOutstanding, FALSE);
SetDlgItemInt(hwnd, ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,

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

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

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

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

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

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

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

```



```

else if ( IsDlgButtonChecked(hwnd, IDC_TM_ENCINA) )
    Reg.eTxnMon = ENCINA;

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

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

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

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

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

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

    if (install_com(szDllPath))
    {
        ShowWindow(hwnd, SW_SHOWNA);
        DestroyWindow(hDlg);

```

```

        strcpy( szErrTxt, "Error occured when configuring COM
settings." );
        MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP | MB_OK);
        EndDialog(hwnd, 0);
        return;
    }
}

Sleep(100);

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

EndDialog(hwnd, rc);
return;
}

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

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

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

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

        RegCloseKey(hKey);
    }

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

        RegCloseKey(hKey);
    }
}

static void WriteRegistrySettings(char *szDllPath)

```

```

{
    HKEY    hKey;
    DWORD   dwDisposition;
    char    szTmp[256];
    char    *ptr;
    int     iRc;

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

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

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

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

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

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

        RegFlushKey(hKey);
        RegCloseKey(hKey);
    }

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

        RegFlushKey(hKey);
        RegCloseKey(hKey);
    }
}

```

```

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

            RegFlushKey(hKey);
            RegCloseKey(hKey);
        }

        return;
    }

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

            return TRUE;
        }
        return FALSE;
    }

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

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

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

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

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

```

```

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

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

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

        CloseHandle(hFile);

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

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

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

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

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

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

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

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

```

```

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

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

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

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

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

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

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

        return 1;
    }

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

```

```

int          iRc;

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

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

    return bRc;
}

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

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

    versionExeMS = 0x7FFF;
    versionExeLS = 0x7FFF;
    dwSize = GetFileVersionInfoSize(szExePath, &d);
    if ( dwSize )

```

```

    {
        ptr = (char *)malloc(dwSize);
        GetFileVersionInfo(szExePath, 0, dwSize, ptr);
        VerQueryValue(ptr, "\\",&vs, &dwBytes);

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

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

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

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

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

    CloseServiceHandle(schService);
    return TRUE;
}

ServiceNotRunning:

    CloseServiceHandle(schService);
    return FALSE;
}

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

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

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

```

```

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

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

    CloseServiceHandle(schService);
    return TRUE;

StartWWWebErr:
    CloseServiceHandle(schService);
    return FALSE;
}

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

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

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

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

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

    CloseServiceHandle(schService);
    return TRUE;
}

```

```

StopWWWebErr:
    CloseServiceHandle(schService);
    return FALSE;
}

static void UpdateDialog(HWND hDlg)
{
    MSG msg;

    UpdateWindow(hDlg);
    while( PeekMessage(&msg, hDlg, 0, 0, PM_REMOVE) )
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    Sleep(250);
    return;
}

```

install.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_THREAD_POOL_LIMIT 1015
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE 1017
#define ED_IIS_THREAD_TIMEOUT 1018
#define ED_IIS_LISTEN_BACKLOG 1019
#define IDC_DBLIB          1021
#define IDC_ODBC            1022

#define IDC_CONNECT_POOL    1023
#define ED_USER_CONNECT_DELAY_TIME 1024

// Next default values for new objects
//

```

install.rc

```

//Microsoft Developer Studio generated resource script.

```

```

//
#include "resource.h"

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

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

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

#ifdef !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
#ifdef WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif // WIN32

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

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

```

```

DEFPUSHBUTTON     "OK",IDOK,53,331,50,14
PUSHBUTTON        "Cancel",IDCANCEL,119,331,50,14
EDITTEXT          IDC_PATH,106,26,91,13,ES_AUTOHSCROLL | ES_READONLY
LTEXT             "Number of Delivery Threads:",IDC_STATIC,35,45,115,12
LTEXT             "Max Number of Connections:",IDC_STATIC,35,73,115,12
RTEXT             "Version 4.11",IDC_VERSION,120,4,89,9
LTEXT             "IIS Max Thread Pool Limit:",IDC_STATIC,36,263,115,12
LTEXT             "Web Service Backlog Queue Size:",IDC_STATIC,36,277,115,
12
LTEXT             "IIS Thread Timeout (seconds):",IDC_STATIC,36,291,115,12
LTEXT             "IIS Listen Backlog:",IDC_STATIC,36,307,115,10
GROUPBOX          "Database Interface",IDC_STATIC,35,208,163,27,WS_GROUP
LTEXT             "Installation directory:",IDC_STATIC,35,29,71,10
GROUPBOX          "Transaction Monitor",IDC_STATIC,33,90,165,37
LTEXT             "Server Name:",IDC_STATIC,35,155,56,8
LTEXT             "User ID:",IDC_STATIC,35,168,60,8
LTEXT             "User Password:",IDC_STATIC,35,181,83,8
LTEXT             "Database Name:",IDC_STATIC,35,194,54,8
GROUPBOX          "SQL Server Connection Properties",IDC_STATIC,22,139,187,
102
GROUPBOX          "Web Client Properties",IDC_STATIC,22,15,187,118
GROUPBOX          "IIS Settings",IDC_STATIC,22,247,187,79
LTEXT             "Max Pending Deliveries:",IDC_STATIC,35,59,115,12
END

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

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

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

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

```

```

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
  IDD_DIALOG1, DIALOG
  BEGIN
    LEFTMARGIN, 22
    RIGHTMARGIN, 209
    VERTGUIDE, 35
    VERTGUIDE, 198
    TOPMARGIN, 4
    BOTTOMMARGIN, 345
  END

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

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

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

```

```

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

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

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

```

```

#endif // APSTUDIO_INVOKED

```

```

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

```

```

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

```

```

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

```

```

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

```

```

#ifdef _MAC

```

```

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

```

```

VS_VERSION_INFO VERSIONINFO
FILEVERSION 0,4,20,0
PRODUCTVERSION 0,4,20,0
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x40004L
FILETYPE 0x1L
FILESUBTYPE 0x0L
BEGIN

```

```

  BLOCK "StringFileInfo"

```

```

  BEGIN

```

```

    BLOCK "040904b0"

```

```

    BEGIN

```

```

      VALUE "Comments", "TPC-C Web Client Installer\0"
      VALUE "CompanyName", "Microsoft\0"
      VALUE "FileDescription", "install\0"
      VALUE "FileVersion", "0, 4, 20, 0\0"
      VALUE "InternalName", "install\0"
      VALUE "LegalCopyright", "Copyright © 1999\0"
      VALUE "OriginalFilename", "install.exe\0"
      VALUE "ProductName", "Microsoft install\0"
      VALUE "ProductVersion", "0, 4, 20, 0\0"

```

```

    END

```

```

  END

```

```

  BLOCK "VarFileInfo"

```

```

  BEGIN

```

```

    VALUE "Translation", 0x409, 1200

```

```

  END

```

```

END

```

```

#endif // !_MAC

```

```

////////////////////////////////////////
//
// LICENSE
//
IDR_LICENSE1           LICENSE DISCARDABLE    "license.txt"
////////////////////////////////////////
//
// DBLIB_DLL
//
IDR_DBLIB_DLL          DBLIB_DLL DISCARDABLE
"..\\..\\db_dblib_dll\\bin\\tpcc_dblib.dll"
////////////////////////////////////////
//
// ODBC_DLL
//
IDR_ODBC_DLL           ODBC_DLL DISCARDABLE
"..\\..\\db_odbc_dll\\bin\\tpcc_odbc.dll"
////////////////////////////////////////
//
// TUXEDO_APP
//
IDR_TUXEDO_APP         TUXEDO_APP DISCARDABLE "..\\..\\tuxapp\\bin\\tuxapp.exe"
////////////////////////////////////////
//
// TUXEDO_DLL
//
IDR_TUXEDO_DLL         TUXEDO_DLL DISCARDABLE
"..\\..\\tm_tuxedo_dll\\bin\\tpcc_tuxedo.dll"
////////////////////////////////////////
//
// COM_DLL
//
IDR_COM_DLL            COM_DLL DISCARDABLE
"..\\..\\tm_com_dll\\bin\\tpcc_com.dll"
////////////////////////////////////////
//
// COM_PS_DLL
//
IDR_COMPS_DLL         COM_PS_DLL DISCARDABLE
"..\\..\\tpcc_com_ps\\bin\\tpcc_com_ps.dll"
////////////////////////////////////////
//
// COM_ALL_DLL
//
IDR_COMALL_DLL        COM_ALL_DLL DISCARDABLE
"..\\..\\tpcc_com_all\\bin\\tpcc_com_all.dll"
#endif // English (U.S.) resources

```

```

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

```

install_com.cpp

```

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

#define _WIN32_WINNT 0x0500

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

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

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

    ICatalogObject*     pCatalogObjectApp  = NULL;
    ICatalogObject*     pCatalogObjectCo   = NULL;
    ICatalogObject*     pCatalogObjectItf  = NULL;
    ICatalogObject*     pCatalogObjectMethod = NULL;

    _bstr_t              bstrTemp, bstrTemp2, bstrTemp3,
bstrTemp4;
    _bstr_t              bstrDllPath = szDllPath;
    variant_t            vTmp, vKey;
    long                 lActProp, lCount, lCountCo,
lCountItf, lCountMethod;

```



```

bool                                bTmp;

CoInitializeEx(NULL, COINIT_MULTITHREADED);

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

if (!SUCCEEDED(hr)) goto Error;

bstrTemp = "Applications";

// Attempt to connect to "Applications" in the Catalog
hr = pCOMAdminCat->GetCollection(bstrTemp,

(IDispatch**) &pCatalogCollectionApp);
if (!SUCCEEDED(hr)) goto Error;

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

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

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

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

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

hr = pCOMAdminCat->InstallComponent(bstrTemp,
bstrTemp2,
bstrTemp3,
bstrTemp4);
if (!SUCCEEDED(hr)) goto Error;

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

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

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

// iterate through components in application and set the properties
while (lCountCo > 0)
{

```

```

        hr = pCatalogCollectionCo->get_Item(lCountCo - 1, (IDispatch**)
&pCatalogObjectCo);
        if (!SUCCEEDED(hr)) goto Error;

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

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

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

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

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

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

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

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

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

        hr = pCatalogCollectionItf->get_Count(&lCountItf);
        if (!SUCCEEDED(hr)) goto Error;

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

```

```

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

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

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

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

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

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

                pCatalogObjectMethod->Release();
                pCatalogObjectMethod = NULL;

                lCountMethod--;
        }

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

        pCatalogObjectItf->Release();
        pCatalogObjectItf = NULL;

        lCountItf--;
    }

    pCatalogObjectCo->Release();
    pCatalogObjectCo = NULL;

    lCountCo--;
}

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

pCatalogCollectionApp->Release();
pCatalogCollectionApp = NULL;

pCatalogCollectionCo->Release();

```

```

pCatalogCollectionCo = NULL;

pCatalogCollectionItf->Release();
pCatalogCollectionItf = NULL;

pCatalogCollectionMethod->Release();
pCatalogCollectionMethod = NULL;

Error:
    CoUninitialize();

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

```

install_resource.h

```

//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by install.rc
//
#define IDD_DIALOG1 101
#define IDI_ICON1 102
#define IDR_TPCCDLL 103
#define IDD_DIALOG2 105
#define IDI_ICON2 106
#define IDR_DELIVERY 107
#define IDD_DIALOG3 108
#define IDR_LICENSE1 112
#define IDD_DIALOG4 113
#define IDR_TPCCOBJ1 117
#define IDR_TPCCSTUB1 118
#define IDR_DBLIB_DLL 122
#define IDR_ODBC_DLL 123
#define IDR_TUXEDO_APP 124
#define IDR_TUXEDO_DLL 125
#define IDR_COM_DLL 126
#define IDR_COMPS_DLL 127
#define IDR_COMALL_DLL 128
#define BN_LOG 1001

```

```

#define ED_KEEP 1002
#define ED_THREADS 1003
#define ED_THREADS2 1004
#define IDC_PATH 1007
#define IDC_VERSION 1009
#define IDC_RESULTS 1010
#define IDC_PROGRESS1 1011
#define IDC_STATUS 1012
#define IDC_BUTTON1 1013
#define ED_MAXCONNECTION 1014
#define ED_IIS_MAX_THREAD_POOL_LIMIT 1015
#define ED_MAXDELIVERIES 1016
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE 1017
#define ED_IIS_THREAD_TIMEOUT 1018
#define ED_IIS_LISTEN_BACKLOG 1019
#define IDC_DBLIB 1021
#define IDC_LICENSE 1022
#define IDC_ODBC 1022
#define IDC_CONNECT_POOL 1023
#define ED_DB_SERVER 1023
#define ED_USER_CONNECT_DELAY_TIME 1024
#define ED_DB_USER_ID 1024
#define IDC_MTS 1025
#define IDC_TM_MTS 1025
#define IDC_TM_TUXEDO 1026
#define IDC_TM_NONE 1027
#define ED_DB_PASSWORD 1028
#define ED_DB_NAME 1029
#define IDC_TM_ENCINA 1030

```

```

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

```

isapi_dll.dsp

```

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

```

```

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

```

```

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

```

```

!MESSAGE "isapi_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "isapi_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "isapi_dll - Win32 IceCAP" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

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

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "NDEBUG" /D "WIN32" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ..\common\txnlog\lib\release\rtetime.lib
..\common\txnlog\lib\release\spinlock.lib ..\common\txnlog\lib\release\error.lib
..\common\txnlog\lib\release\txnlog.lib wsock32.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
/nodfaultlib:"LIBCMT" /out:".bin\tpcc.dll"
# SUBTRACT LINK32 /profile /pdb:none /nodfaultlib

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c

```

```

# ADD CPP /nologo /MDd /W3 /GX /ZI /Od /D "_DEBUG" /D "WIN32" /D "_WINDOWS" /FR /YX
/FD /c
# ADD BASE MTL /nologo /D " _DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D " _DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d " _DEBUG"
# ADD RSC /l 0x409 /d " _DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ..\common\txnlog\lib\debug\rtetime.lib
..\common\txnlog\lib\debug\spinlock.lib ..\common\txnlog\lib\debug\error.lib
..\common\txnlog\lib\debug\txnlog.lib wsock32.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib
odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386
/nodfaultlib:"LIBCMTD" /out:".bin\tpcc.dll" /pdbtype:sept
# SUBTRACT LINK32 /profile /pdb:none /nodfaultlib

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "isapi_dll"
# PROP BASE Intermediate_Dir "isapi_dll"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /GX /Zi /Od /D "_DEBUG" /D "WIN32" /D "_WINDOWS" /FR
/YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /GX /Zi /O2 /D "NDEBUG" /D "ICECAP" /D "WIN32" /D
"_WINDOWS" /FR /YX /FD /Gh /c
# ADD BASE MTL /nologo /D " _DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D " _DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d " _DEBUG"
# ADD RSC /l 0x409 /d " _DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc.dll"
/pdbtype:sept
# SUBTRACT BASE LINK32 /profile /pdb:none
# ADD LINK32 icap.lib ..\common\txnlog\lib\release\rtetime.lib
..\common\txnlog\lib\release\spinlock.lib ..\common\txnlog\lib\release\error.lib
..\common\txnlog\lib\release\txnlog.lib wsock32.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug
/machine:I386 /out:".bin\tpcc.dll" /pdbtype:sept
# SUBTRACT LINK32 /profile /pdb:none /map

!ENDIF

# Begin Target

```

```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

isapi_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
#ifdef 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

```

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 );
}

```

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

```

tm_com_dll.dsp

```

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

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

CFG=tm_com_dll - Win32 Debug

```

```

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

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

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /1 0x409 /d "NDEBUG"
# ADD RSC /1 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386 /out:".bin\tpcc_com.dll"

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0

```

```

# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /1 0x409 /d "_DEBUG"
# ADD RSC /1 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_com.dll"
/pdbtype:sept

!ENDIF

# Begin Target

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

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

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



---


tpcc.cpp


---


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

#include <windows.h>
#include <process.h>

```

```

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

#include <sqltypes.h>

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

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

#include "..\..\common\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 //used to log delivery transaction information *txnDelilog = NULL;

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

// configuration settings from registry
TPCCREGISTRYDATA Reg;

DWORD dwNumDeliveryThreads = 4;
CRITICAL_SECTION DelBuffCriticalSection; //critical
section for delivery transactions cache
DELIVERY_TRANSACTION *pDelBuff = NULL;
DWORD dwDelBuffSize =
100; // size of circular buffer for delivery txns
DWORD dwDelBuffFreeCount;
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 occurred in initialization

```



```

*
*           DLL successfully initialized           TRUE
*/
BOOL WINAPI DllMain(HANDLE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
    DWORD i;
    char szEvent[LEN_ERR_STRING] = "\\0";
    char szLogFile[128];
    char szDllName[128];

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

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

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

                TermInit();

                // load DLL for txn monitor
                if (Reg.eTxnMon == TUXEDO)
                {
                    strcpy( szDllName, Reg.szPath );
                    strcat( szDllName,
"tpcc_tuxedo.dll");
                    szDllName );
                    hLibInstanceTm = LoadLibrary(
                    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");
                    szDllName );
                    hLibInstanceTm = LoadLibrary(
                    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");
                        LoadLibrary( szDllName );
                        hLibInstanceDb =
                        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");
    LoadLibrary( szDllName );
    NULL)
    if (hLibInstanceDb ==
    throw new
    CWEBCLNT_ERR( ERR_LOADDLL_FAILED, szDllName, GetLastError() );
    // get function pointer
    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
    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);
    lstrcpy(pVer->lpszExtensionDesc, "TPC-C Server.",
HSE_MAX_EXT_DLL_NAME_LEN);

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

    return TRUE;
}

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

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

    TermDeleteAll();
    return TRUE;
}

/* FUNCTION: HttpExtensionProc

```

```

 *
 * PURPOSE:      This function is the main entry point for the TPCC DLL. The
internet service
 *
 *               calls this function passing in the http string.
 *
 * ARGUMENTS:    EXTENSION_CONTROL_BLOCK  *pECB    structure pointer to
passed in internet
 *
 *               service information.
 *
 * RETURNS:      DWORD              HSE_STATUS_SUCCESS
connection can be dropped if error
 *
 *               HSE_STATUS_SUCCESS_AND_KEEP_CONN    keep connect valid comment sent
 *
 * COMMENTS:     None
 */

DWORD WINAPI HttpExtensionProc(EXTENSION_CONTROL_BLOCK *pECB)
{
    int          iCmd, FormId, TermId, 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 );
}
#endif 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

    int iRetryCnt = 0;
    static int iMaxRetries = 10;

    assert(txnDeliRec != NULL);

Reconnect:

```

```

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

            // will retry connection up to ten times
            if (iRetryCnt++ < iMaxRetries)
            {
                Sleep(5000); // delay for 5 seconds
                goto Reconnect;
            }

            wsprintf( szTmp, "Delivery Txn thread terminating after %d
retries.", iMaxRetries );
            WriteMessageToEventLog( szTmp );
            goto ErrorExit;
        }
        catch (...)
        {
            WriteMessageToEventLog(TEXT("Unhandled exception caught in
DeliveryWorkerThread. Delivery Txn thread terminating."));
            goto ErrorExit;
        }

        while (TRUE)
        {
            try
            {
                //while delivery thread running, i.e. user has not
                requested termination
                while (TRUE)
                {
                    // need to wait for multiple objects:
                    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];
        sprintf( szTmp, "Error in Delivery Txn thread. %s",
e->ErrorText() );
        WriteMessageToEventLog( szTmp );

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

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

```

```

    }
}

ErrorExit:
    delete pTxn;
    _endthread();
}

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

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

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

        dwDelBuffFreeCount--;
        dwDelBuffFreeIndex++;
        if (dwDelBuffFreeIndex == dwDelBuffSize)
            dwDelBuffFreeIndex = 0;          // 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>"
New\ "><PRE>"
"__TIME__" <BR>"
("__TIMESTAMP__") <BR>"
ACTION="\tpcc.dll" METHOD="GET">"
NAME="\STATUSID\" VALUE="\0\">"
NAME="\ERROR\" VALUE="\0\">"
NAME="\FORMID\" VALUE="\1\">"
NAME="\TERMID\" VALUE="\0\">"
NAME="\SYNCID\" VALUE="\0\">"
NAME="\VERSION\" VALUE="\\" WEBCLIENT_VERSION "\\">"
);
    sprintf( szTmp, "Configuration Settings: <BR><font face=\Courier
New\" color=\blue\"><PRE>"
<B>%s</B><BR>"
<B>%s</B><BR>"
<B>%d</B><BR>"
<B>%d</B><BR>"
<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>"
NAME="\db_server\" SIZE=20 VALUE="\%s\"><BR>"
NAME="\db_user\" SIZE=20 VALUE="\%s\"><BR>"
"DB Server = <INPUT
"DB User ID = <INPUT

```

```

"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
"DB Server
"DB User ID
"DB Password
"DB Name
"\"/PRE></font>"
, Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, Reg.szDbName );
strcat( szBuffer, szTmp);
sprintf( szTmp, "Please enter your Warehouse and District for this
session:<BR>"
"\"<font face=\"Courier New\"
color=\"blue\"><PRE>" );
strcat( szBuffer, szTmp);
strcat( szBuffer, "Warehouse ID = <INPUT NAME=\"w_id\" SIZE=4><BR>"
"District ID = <INPUT
NAME=\"d_id\" SIZE=2><BR>"
"\"/PRE></font><HR>"
"\"<INPUT TYPE=\"submit\">"
NAME=\"CMD\" VALUE=\"Submit\">"
"\"/FORM></BODY></HTML>");
}
/* FUNCTION: SubmitCmd
*
* PURPOSE: This function allocated a new terminal id in the Term structure
array.
*
*/
void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer)
{
int iNewTerm;
char *ptr = pECB->lpszQueryString;
char szVersion[32] = { 0 };
char szServer[32] = { 0 };
char szUser[32] = "sa";
char szPassword[32] = { 0 };
char szDatabase[32] = "tpcc";
// validate version field; the version field ensures that the RTE is
synchronized with the web client

```

```

GetKeyValue(&ptr, "VERSION", szVersion, sizeof(szVersion),
ERR_VERSION_MISMATCH);
if ( strcmp( szVersion, WEBCIENT_VERSION ) )
throw new CWEBCLNT_ERR( ERR_VERSION_MISMATCH );
if (Reg.eTxnMon == None)
{
// parse Server name
GetKeyValue(&ptr, "db_server", szServer, sizeof(szServer),
ERR_NO_SERVER_SPECIFIED);
// parse User name
GetKeyValue(&ptr, "db_user", szUser, sizeof(szUser), NO_ERR);
// parse Password
GetKeyValue(&ptr, "db_passwd", szPassword, sizeof(szPassword),
NO_ERR);
// parse Database name
GetKeyValue(&ptr, "db_name", szDatabase, sizeof(szDatabase),
NO_ERR);
}
// parse warehouse ID
int w_id = GetIntKeyValue(&ptr, "w_id", ERR_HTML_ILL_FORMED,
ERR_W_ID_INVALID);
if ( w_id < 1 )
throw new CWEBCLNT_ERR( ERR_W_ID_INVALID );
// parse district ID
int d_id = GetIntKeyValue(&ptr, "d_id", ERR_HTML_ILL_FORMED,
ERR_D_ID_INVALID);
if ( d_id < 1 || d_id > 10 )
throw new CWEBCLNT_ERR( ERR_D_ID_INVALID );
iNewTerm = TermAdd();
Term.pClientData[iNewTerm].w_id = w_id;
Term.pClientData[iNewTerm].d_id = d_id;
try
{
if (Reg.eTxnMon == TUXEDO)
Term.pClientData[iNewTerm].pTxn = pCTPCC_TUXEDO_new();
else if (Reg.eTxnMon == ENCINA)
Term.pClientData[iNewTerm].pTxn = pCTPCC_ENCINA_new();
else if (Reg.eTxnMon == COM)
Term.pClientData[iNewTerm].pTxn = pCTPCC_COM_new(
Reg.bCOM_SinglePool );
else if (Reg.eDB_Protocol == ODBC)
Term.pClientData[iNewTerm].pTxn = pCTPCC_ODBC_new(
szServer, szUser, szPassword, szMyComputerName, szDatabase );
else if (Reg.eDB_Protocol == DBLIB)
Term.pClientData[iNewTerm].pTxn = pCTPCC_DBLIB_new(
szServer, szUser, szPassword, szMyComputerName, szDatabase );
}
catch (...)
{
TermDelete(iNewTerm);
throw; // pass exception upward
}
MakeMainMenuForm(iNewTerm, Term.pClientData[iNewTerm].iSyncId, szBuffer);
}

```



```

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

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

    EnterCriticalSection(&TermCriticalSection);

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

    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>"
              , iTotals );
}

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_TOO_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)
    vsprintf( 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 CWBCLNT_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 CWBCLNT_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 CWBCLNT_ERR(err)
*                                else
*                                    return 0
*                                else if (non-numeric char found) then
*                                    if (NotIntErr != NO_ERR) then
*                                        throw CWBCLNT_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 CWBCLNT_ERR( NoKeyErr );
        return 0;
    }

    *pQueryString = ptr;
    return atoi(ptr0);

ErrorNoKey:
    if (NoKeyErr != NO_ERR)
        throw new CWBCLNT_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=\"TERMINID\" VALUE=\"%d\">"
"INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
" BOLD>An Error Occurred</BOLD><BR><BR>"
"%s"
"<BR><BR><HR>"
"INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
"INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Payment..\">"
"INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Delivery..\">"
"INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-
Status..\">"
"INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
"INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Exit..\">"
"/FORM></BODY></HTML>"
, iType, iErrorNum, MAIN_MENU_FORM, iTermId, 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=\"TERMINID\" 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=\"TERMINID\" VALUE=\"%d\">"
"INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
"PRE><font face=\"Courier\">"
"Stock-Level<BR>"
"Warehouse: %4.4d District: %2.2d<BR> <BR>",
STOCK_LEVEL_FORM, iTermId, Term.pClientData[iTermId].iSyncId,
Term.pClientData[iTermId].w_id, Term.pClientData[iTermId].d_id);

    if ( bInput )
    {
        strcpy(szForm+c,
"Stock Level Threshold: <INPUT NAME=\"TT*\"
SIZE=2><BR> <BR>"
"low stock:  </font><BR> <BR> <BR> <BR> <BR> <BR>"
" <BR> <BR> <BR> <BR> <BR> <BR> <BR></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></PRE><HR>"

```



```

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

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

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

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

    PDELIVERY_DATA      pDelivery;

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

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

    if (dwNumDeliveryThreads)
    {
        //post delivery info
        if ( PostDeliveryInfo(pDelivery->w_id, pDelivery->o_carrier_id)

                pDelivery->exec_status_code = eDeliveryFailed;
            else
                pDelivery->exec_status_code = eOK;
        }
    else // delivery is done synchronously if no delivery threads configured
        Term.pClientData[iTermId].pTxn->Delivery();

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

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

```

```

 *
 * ARGUMENTS:   EXTENSION_CONTROL_BLOCK      *pECB      passed in structure
pointer from inetsrv.
 *
 *              int
iTermId      client browser terminal id
 *
 */

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

    PSTOCK_LEVEL_DATA      pStockLevel;

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

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

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

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

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

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

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

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

```

```

        "Qty10*", "Qty11*", "Qty12*", "Qty13*", "Qty14*" );

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

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

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

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

        _strupr( szTmp );
        if ( strlen(pPaymentData->c_last) > LAST_NAME_LEN )
            throw new CWBCLNT_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 CWBCLNT_ERR( ERR_PAYMENT_CID_AND_CLT );
    }

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

```

```

        throw new CWBCLNT_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 CWBCLNT_ERR(
ERR_ORDERSTATUS_MISSING_CID_CLT );

        _strupr( szTmp );
        if ( strlen(pOrderStatusData->c_last) > LAST_NAME_LEN )
            throw new CWBCLNT_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 CWBCLNT_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 CWBCLNT_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 *dotp;
    BOOL bValid;

    if ( *ptr == 0 )
        return FALSE;

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

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

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

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

```

tpcc.def

LIBRARY TPCC.DLL

EXPORTS

GetExtensionVersion @1

```
HttpExtensionProc @2
TerminateExtension @3
```

tpcc.h

```
/* FILE: TPC.C.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;
//synchronization id
int iTickCount;
//time of last access;

CTPCC_BASE *pTxn;

} CLIENTDATA, *PCLIENTDATA;

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

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

enum WEBERROR
{
    NO_ERR,
    ERR_COMMAND_UNDEFINED,
    ERR_D_ID_INVALID,
    ERR_DELIVERY_CARRIER_ID_RANGE,
    ERR_DELIVERY_CARRIER_INVALID,
    ERR_DELIVERY_MISSING_OCD_KEY,
    ERR_DELIVERY_THREAD_FAILED,
    ERR_GETPROCADDR_FAILED,
    ERR_HTML_ILL_FORMED,
    ERR_INVALID_SYNC_CONNECTION,
    ERR_INVALID_TERMID,
    ERR_LOADDLL_FAILED,
    ERR_MAX_CONNECTIONS_EXCEEDED,
    ERR_MEM_ALLOC_FAILED,
    ERR_MISSING_REGISTRY_ENTRIES,
    ERR_NEWORDER_CUSTOMER_INVALID,
    ERR_NEWORDER_CUSTOMER_KEY,
    ERR_NEWORDER_DISTRICT_INVALID,
    ERR_NEWORDER_FORM_MISSING_DID,
    ERR_NEWORDER_ITEMID_INVALID,
    ERR_NEWORDER_ITEMID_RANGE,
    ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
    ERR_NEWORDER_MISSING_IID_KEY,
    ERR_NEWORDER_MISSING_QTY_KEY,
    ERR_NEWORDER_MISSING_SUPPW_KEY,
    ERR_NEWORDER_NOITEMS_ENTERED,
    ERR_NEWORDER_QTY_INVALID,
    ERR_NEWORDER_QTY_RANGE,
    ERR_NEWORDER_QTY_WITHOUT_SUPPW,
    ERR_NEWORDER_SUPPW_INVALID,
    ERR_NO_SERVER_SPECIFIED,
    ERR_ORDERSTATUS_CID_AND_CLT,
    ERR_ORDERSTATUS_CID_INVALID,
```

```

ERR_ORDERSTATUS_CLT_RANGE,
ERR_ORDERSTATUS_DID_INVALID,
ERR_ORDERSTATUS_MISSING_CID_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_TOO_LONG,
ERR_PAYMENT_MISSING_CDI_KEY,
ERR_PAYMENT_MISSING_CID_CLT,
ERR_PAYMENT_MISSING_CID_KEY,
ERR_PAYMENT_MISSING_CLT,
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(LPCTSTR 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(LPCTSTR lpszQueryString, NEW_ORDER_DATA *pNewOrderData);
void GetPaymentData(LPCTSTR lpszQueryString, PAYMENT_DATA *pPaymentData);
void GetOrderStatusData(LPCTSTR lpszQueryString, ORDER_STATUS_DATA *pOrderStatusData);
BOOL PostDeliveryInfo(short w_id, short o_carrier_id);
BOOL IsNumeric(char *ptr);
BOOL IsDecimal(char *ptr);

```

```
void DeliveryWorkerThread(void *ptr);
```

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

#ifdef _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
#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
    END
    #endif // APSTUDIO_INVOKED

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

    #ifndef APSTUDIO_INVOKED
    ///////////////////////////////////////////////////////////////////
    //
    // Generated from the TEXTINCLUDE 3 resource.
    //
    ///////////////////////////////////////////////////////////////////
    #endif // not APSTUDIO_INVOKED

```

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 definitions of structures specific to TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_com.h"

#include "..\..\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "..\..\tpcc_com_all\src\tpcc_com_all_i.c"

// wrapper routine for class constructor
_declspec( dllexport ) CTPCC_COM* CTPCC_COM_new(BOOL bSinglePool)
{
    return new CTPCC_COM(bSinglePool);
}

CTPCC_COM::CTPCC_COM(BOOL bSinglePool)
{
    HRESULT hr = NULL;

```

```

    long lRet = 0;
    ULONG ulTmpSize = 0;

    m_pTxn                = NULL;
    m_pNewOrder            = NULL;
    m_pPayment             = NULL;
    m_pStockLevel         = NULL;
    m_pOrderStatus        = NULL;

    m_bSinglePool         = bSinglePool;

    ulTmpSize = (ULONG) sizeof(COM_DATA);
    VariantInit(&m_vTxn);
    m_vTxn.vt = VT_SAFEARRAY;

    m_vTxn.parray = SafeArrayCreateVector(VT_UI1, ulTmpSize, ulTmpSize);
    if (!m_vTxn.parray)
        throw new CCOMERR( E_FAIL );

    memset((void*)m_vTxn.parray->pvData, 0, ulTmpSize);
    m_pTxn = (COM_DATA*)m_vTxn.parray->pvData;

    hr = CoInitializeEx(NULL, COINIT_MULTITHREADED);
    if (FAILED(hr))
    {
        throw new CCOMERR( hr );
    }

    // create components
    if (m_bSinglePool)
    {
        hr = CoCreateInstance(CLSID_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 );
}

```

tpcc_com.h

```

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

#pragma once

#include <stdio.h>
#include "..\..\tpcc_com_ps\src\tpcc_com_ps.h"

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

```

```

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

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

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

    int m_hr;
    int m_iErrorType;
    int m_iError;

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

    int ErrorNum() {return m_hr;}

    char *ErrorText()
    {
        if (m_hr == S_OK)
            sprintf( m_szErrorText, "Error: Class %d",
                error # %d", m_iErrorType, m_iError );
        else
            sprintf( m_szErrorText, "Error: COM HRESULT
                %x", m_hr );
        return m_szErrorText;
    }
};

class DllDecl CTPCC_COM : public CTPCC_BASE
{
private:
    BOOL m_bSinglePool;

    // COM Interface pointers
    ITPCC* m_pNewOrder;
    ITPCC* m_pPayment;
    ITPCC* m_pStockLevel;
    ITPCC* m_pOrderStatus;
};

```

```

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

VARIANT m_vTxn;

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

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

    void NewOrder ();
    void Payment ();
    void StockLevel ();
    void OrderStatus ();
    void Delivery () { throw new CCOMERR(E_NOTIMPL); }

} // not supported

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

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

typedef CTPCC_COM* (TYPE_CTPCC_COM) (BOOL);

```

tpcc_com_all.cpp

```

/* FILE: TPCCOM_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 definitions 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
            (LPCWSTR *)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. GetProcAddress 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)
        sprintf( szTmp+strlen(szTmp), " Error=%d", m_SystemErr );

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

CTPCC_Common::CTPCC_Common()
{
    m_pTxn = NULL;
    m_bCanBePooled = TRUE;
}

CTPCC_Common::~CTPCC_Common()
{
    if (m_pTxn)
        delete m_pTxn;
}

HRESULT CTPCC_Common::CallSetComplete()
{
    IObjectContext* pObjectContext = NULL;

    // get our object context
    HRESULT hr = CoGetObjectContext( IID_IObjectContext, (void
**) &pObjectContext );
    pObjectContext->SetComplete();
    ReleaseInterface(pObjectContext);
    return hr;
}

//
// called by the ctor activator
//
STDMETHODIMP CTPCC_Common::Construct(IDispatch * pUnk)
{
    // Code to access construction string, if needed later...
    // if (!pUnk)
    //     return E_UNEXPECTED;
    // IObjectConstructString * pString = NULL;
    // HRESULT hr = pUnk->QueryInterface(IID_IObjectConstructString,
(void **) &pString);
    // pString->Release();

    try

```

```

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

    return S_OK;
}

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

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

    m_pTxn->OrderStatus();

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

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

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

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

```

tpcc_com_all.def

; tpcc_com_all.def : Declares the module parameters.

```

LIBRARY          "tpcc_com_all.dll"

EXPORTS
    DllCanUnloadNow      @1 PRIVATE
    DllGetClassObject    @2 PRIVATE
    DllRegisterServer    @3 PRIVATE

```

DllUnregisterServer @4 PRIVATE

tpcc_com_all.dsp

```

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

```

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

```
CFG=tpcc_com_all - Win32 Debug
```

```
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
```

```
!MESSAGE use the Export Makefile command and run
```

```
!MESSAGE
```

```
!MESSAGE NMAKE /f "tpcc_com_all.mak".
```

```
!MESSAGE
```

```
!MESSAGE You can specify a configuration when running NMAKE
```

```
!MESSAGE by defining the macro CFG on the command line. For example:
```

```
!MESSAGE
```

```
!MESSAGE NMAKE /f "tpcc_com_all.mak" CFG="tpcc_com_all - Win32 Debug"
```

```
!MESSAGE
```

```
!MESSAGE Possible choices for configuration are:
```

```
!MESSAGE
```

```
!MESSAGE "tpcc_com_all - Win32 Release" (based on "Win32 (x86) Dynamic-Link
```

```
Library")
```

```
!MESSAGE "tpcc_com_all - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
```

```
!MESSAGE
```

```
# Begin Project
```

```
# PROP AllowPerConfigDependencies 0
```

```
# PROP Scc_ProjName ""
```

```
# PROP Scc_LocalPath ""
```

```
CPP=cl.exe
```

```
MTL=midl.exe
```

```
RSC=rc.exe
```

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

```
# PROP BASE Use_MFC 0
```

```
# PROP BASE Use_Debug_Libraries 0
```

```
# PROP BASE Output_Dir "Release"
```

```
# PROP BASE Intermediate_Dir "Release"
```

```
# PROP BASE Target_Dir ""
```

```
# PROP Use_MFC 0
```

```
# PROP Use_Debug_Libraries 0
```

```
# PROP Output_Dir ".\bin"
```

```
# PROP Intermediate_Dir ".\obj"
```

```
# PROP Ignore_Export_Lib 0
```

```
# PROP Target_Dir ""
```

```
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
```

```
# ADD CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D " _WINDOWS" /YX /FD /c
```

```
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
```

```
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
```

```
# ADD BASE RSC /l 0x409 /d "NDEBUG"
```

```
# ADD RSC /l 0x409 /d "NDEBUG"
```

```
BSC32=bscmake.exe
```

```
# ADD BASE BSC32 /nologo
```

```
# ADD BSC32 /nologo
```

```
LINK32=link.exe
```



```

# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ..\db\dblib_dll\bin\tpcc_dblib.lib ..\db\odbc_dll\bin\tpcc_odbc.lib
kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib /nologo /subsystem:windows
/dll /machine:I386

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /1 0x409 /d "_DEBUG"
# ADD RSC /1 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
!ENDIF

# Begin Target

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

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

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

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

SOURCE=.\src\tpcc_com_all.idl

```

```

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

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

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

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

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

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

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

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

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

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

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

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

!ENDIF

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

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

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

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

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

```

tpcc_com_all.h

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

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

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jun 12 18:15:19 2000
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf (OptLev=i2), Wl, Zp8, env=Win32 (32b run), ms_ext, c_ext
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

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

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

#ifdef __tpcc_com_all_h__
#define __tpcc_com_all_h__

/* Forward Declarations */

#ifdef __TPCC_FWD_DEFINED__
#define __TPCC_FWD_DEFINED__

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

#endif /* __TPCC_FWD_DEFINED__ */

#ifdef __NewOrder_FWD_DEFINED__
#define __NewOrder_FWD_DEFINED__

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

#endif /* __NewOrder_FWD_DEFINED__ */

#ifdef __OrderStatus_FWD_DEFINED__
#define __OrderStatus_FWD_DEFINED__

#ifdef __cplusplus
```

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

#endif /* __OrderStatus_FWD_DEFINED__ */

#ifdef __Payment_FWD_DEFINED__
#define __Payment_FWD_DEFINED__

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

#endif /* __Payment_FWD_DEFINED__ */

#ifdef __StockLevel_FWD_DEFINED__
#define __StockLevel_FWD_DEFINED__

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

#endif /* __StockLevel_FWD_DEFINED__ */

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

#ifdef __cplusplus
extern "C"{
#endif

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

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

extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_s_ifspec;

#ifdef __TPCCLib_LIBRARY_DEFINED__
#define __TPCCLib_LIBRARY_DEFINED__

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

```

EXTERN_C const IID LIBID_TPCLib;

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 /* __TPCLib_LIBRARY_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

```

tpcc_com_all.idl

```

/* FILE: TPCC.IDL Microsoft TPC-C Kit Ver. 4.20.000
 * Copyright Microsoft, 1999
 */

```

```

* All Rights Reserved
*
* not yet audited
*
* PURPOSE: IDL source for TPCC.dll. This file is processed by the MIDL
tool to
* produce the type library (TPCC.tlb) and
marshalling code.
*
* Change history:
* 4.20.000 - first version
*/

interface TPCC;
interface NewOrder;
interface OrderStatus;
interface Payment;
interface StockLevel;

import "oidl.idl";
import "ocidl.idl";
import "..\tpcc_com_ps\src\tpcc_com_ps.idl";

[
    uuid(122A3117-2520-11D3-BA71-00C04FBFE08B),
    version(1.0),
    helpstring("TPC-C 1.0 Type Library")
]
library TPCLib
{
    importlib("stdole32.tlb");
    importlib("stdole2.tlb");

    [
        uuid(122A3128-2520-11D3-BA71-00C04FBFE08B),
        helpstring("All Txns Class")
    ]
    coclass TPCC
    {
        [default] interface ITPCC;
    };

    [
        uuid(975BAABF-84A7-11D2-BA47-00C04FBFE08B),
        helpstring("NewOrder Class")
    ]
    coclass NewOrder
    {
        [default] interface ITPCC;
    };

    [
        uuid(266836AD-A50D-11D2-BA4E-00C04FBFE08B),
        helpstring("OrderStatus Class")
    ]
    coclass OrderStatus
    {
        [default] interface ITPCC;
    };
}

```

```

[
    uuid(CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B),
    helpstring("Payment Class")
]
coclass Payment
{
    [default] interface ITPCC;
};

[
    uuid(2668369E-A50D-11D2-BA4E-00C04FBFE08B),
    helpstring("StockLevel Class")
]
coclass StockLevel
{
    [default] interface ITPCC;
};

```

tpcc_com_all.rc

```

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

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

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

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
#include _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

#ifdef _MAC
//
// Version
//

VS_VERSION_INFO VERSIONINFO
FILEVERSION 1,0,0,1
PRODUCTVERSION 1,0,0,1
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x4L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904B0"
        BEGIN
            VALUE "CompanyName", "\0"
            VALUE "FileDescription", "tpcc_com_all Module\0"
            VALUE "FileVersion", "1, 0, 0, 1\0"
            VALUE "InternalName", "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

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

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

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

```

tpcc_com_all.rgs

```

HKCR
{
    TPCC.AllTxns.1 = s 'All Txns Class'
    {
        CLSID = s '{122A3128-2520-11D3-BA71-00C04FBFE08B}'
    }
    TPCC.AllTxns = s 'TPCC Class'
    {
        CurVer = s 'TPCC.AllTxns.1'
    }
    NoRemove CLSID
    {
        ForceRemove {122A3128-2520-11D3-BA71-00C04FBFE08B} = s 'TPCC
Class'
        {
            ProgID = s 'TPCC.AllTxns.1'
            VersionIndependentProgID = s 'TPCC.AllTxns'
            InprocServer32 = s '%MODULE%'
            {
                val ThreadingModel = s 'Both'
            }
        }
    }
}

```

tpcc_com_all.i.c

```

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

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

```

```

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

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jun 12 18:15:19 2000
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), Wl, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
        DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#ifdef !defined(_M_IA64) && !defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

```

```

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

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

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

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

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

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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif /* !defined(_M_IA64) && !defined(_M_AXP64) */

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

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

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

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jun 12 18:15:19 2000 */
/*
Compiler settings for .\src\tpcc_com.all.idl:
Oicf (OptLev=i2), Wl, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext, robust
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if defined(_M_IA64) || defined(_M_AXP64)

#ifdef __cplusplus
extern "C" {
#endif

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

```

```

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

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

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

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

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

```

```

#ifdef /* defined(_M_IA64) || defined(_M_AXP64)*/

```

tpcc_com_no.rgs

```

HKCR
{
    TPCC.NewOrder.1 = s 'NewOrder Class'
    {
        CLSID = s '{975BAABF-84A7-11D2-BA47-00C04FBFE08B}'
    }
    TPCC.NewOrder = s 'NewOrder Class'
    {
        CurVer = s 'TPCC.NewOrder.1'
    }
    NoRemove CLSID
    {
        ForceRemove {975BAABF-84A7-11D2-BA47-00C04FBFE08B} = s 'NewOrder
Class'
        {
            ProgID = s 'TPCC.NewOrder.1'
            VersionIndependentProgID = s 'TPCC.NewOrder'
            InprocServer32 = s '%MODULE%'
            {
                val ThreadingModel = s 'Both'
            }
        }
    }
}

```

tpcc_com_os.rgs

```

HKCR
{
    TPCC.OrderStatus.1 = s 'OrderStatus Class'
    {
        CLSID = s '{266836AD-A50D-11D2-BA4E-00C04FBFE08B}'
    }
    TPCC.OrderStatus = s 'OrderStatus Class'
    {
        CurVer = s 'TPCC.OrderStatus.1'
    }
    NoRemove CLSID
    {
        ForceRemove {266836AD-A50D-11D2-BA4E-00C04FBFE08B} = s
'OrderStatus Class'
        {
            ProgID = s 'TPCC.OrderStatus.1'
            VersionIndependentProgID = s 'TPCC.OrderStatus'
            InprocServer32 = s '%MODULE%'
            {
                val ThreadingModel = s 'Both'
            }
        }
    }
}

```

```

}

```

tpcc_com_pay.rgs

```

HKCR
{
    TPCC.Payment.1 = s 'Payment Class'
    {
        CLSID = s '{CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B}'
    }
    TPCC.Payment = s 'Payment Class'
    {
        CurVer = s 'TPCC.Payment.1'
    }
    NoRemove CLSID
    {
        ForceRemove {CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B} = s 'Payment
Class'
        {
            ProgID = s 'TPCC.Payment.1'
            VersionIndependentProgID = s 'TPCC.Payment'
            InprocServer32 = s '%MODULE%'
            {
                val ThreadingModel = s 'Both'
            }
        }
    }
}

```

tpcc_com_ps.def

```

LIBRARY "tpcc_com_ps"

DESCRIPTION 'Proxy/Stub DLL'

EXPORTS
    DllGetClassObject @1 PRIVATE
    DllCanUnloadNow @2 PRIVATE
    GetProxyDllInfo @3 PRIVATE
    DllRegisterServer @4 PRIVATE
    DllUnregisterServer @5 PRIVATE

```

tpcc_com_ps.dsp

```

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

```

```

# TARGETTYPE "Win32 (x86) Application" 0x0101

```

```

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

```

```

!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_ps.mak" CFG="tpcc_com_ps - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpcc_com_ps - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "tpcc_com_ps - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

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

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS_WINNT=0x0400 /D
"REGISTER_PROXY_DLL" /FD /c
# SUBTRACT CPP /YX
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib
/nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpcrt4.lib oleaut32.lib uuid.lib
/nologo /entry:"DllMain" /subsystem:windows /dll /pdb:none /machine:I386
/def:".src\tpcc_com_ps.def"
# Begin Custom Build - Copying tpcc_com_ps.h
InputPath=. \bin\tpcc_com_ps.dll
SOURCE="$(InputPath)"

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

# End Custom Build

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"

```

```

# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD CPP /nologo /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WIN32_WINNT=0x0400 /D
"REGISTER_PROXY_DLL" /FD /c
# ADD BASE MTL /nologo /D "DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "DEBUG"
# ADD RSC /l 0x409 /d "DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpcrt4.lib oleaut32.lib uuid.lib
/nologo /entry:"DllMain" /dll /debug /machine:IX86 /def:".src\tpcc_com_ps.def"
/pdbtype:sept
# SUBTRACT LINK32 /pdb:none
# Begin Custom Build - Copying tpcc_com_ps.h
InputPath=. \bin\tpcc_com_ps.dll
SOURCE="$(InputPath)"

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

# End Custom Build

!ENDIF

# Begin Target

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

# PROP Default_Filter ""
# Begin Source File

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

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

SOURCE=. \src\tpcc_com_ps.idl

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

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

```



```

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

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

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

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

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

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

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

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

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

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

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

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

!ENDIF

# End Source File
# Begin Source File

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

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

```

tpcc_com_ps.h

```

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

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

```

```

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jun 12 18:15:12 2000
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
    DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

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

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

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

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

#ifndef _tpcc_com_ps_h_
#define _tpcc_com_ps_h_

/* Forward Declarations */

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

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

#ifdef __cplusplus
extern "C"{
#endif

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

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

extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

```

```

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

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

EXTERN_C const IID IID_ITPCC;

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

MIDL_INTERFACE("FEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
ITPCC : public IUnknown
{
public:
    virtual HRESULT __stdcall NewOrder(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

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

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

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

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

    virtual HRESULT __stdcall CallSetComplete( void ) = 0;

};

#else /* C style interface */

typedef struct ITPCCVtbl
{
    BEGIN_INTERFACE

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

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

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

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

    HRESULT ( STDMETHODCALLTYPE __RPC_FAR *Payment )(
        ITPCC __RPC_FAR * This,

```

```

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

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

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

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

    HRESULT ( STDMETHODCALLTYPE __RPC_FAR *CallSetComplete )(
        ITPCC __RPC_FAR * This);

    END_INTERFACE
} ITPCCVtbl;

interface ITPCC
{
    CONST_VTBL struct ITPCCVtbl __RPC_FAR *lpVtbl;
};

#ifndef COBJMACROS

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

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

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

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

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

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

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

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

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

#endif /* COBJMACROS */

```

```

#endif /* C style interface */

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

void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE_pRpcMessage,
    DWORD *_pdwStubPhase);

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

void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE_pRpcMessage,
    DWORD *_pdwStubPhase);

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

void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE_pRpcMessage,
    DWORD *_pdwStubPhase);

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

void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE_pRpcMessage,
    DWORD *_pdwStubPhase);

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

```

```

void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE_pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC __RPC_FAR * This);

void __RPC_STUB ITPCC_CallSetComplete_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE_pRpcMessage,
    DWORD *_pdwStubPhase);

#endif /* __ITPCC_INTERFACE_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

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

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

```

tpcc_com_ps.idl

```

/* FILE: ITPCC.IDL
 * Microsoft TPC-C Kit Ver. 4.20.000
 * Copyright Microsoft, 1999
 *
 * All Rights Reserved
 *
 * not yet audited
 *
 * PURPOSE: Defines the interface used by TPCC. This interface can be
 * implemented by C++ components.
 *
 * Change history:
 * 4.20.000 - first version
 */

// Forward declare all types defined
interface ITPCC;

```

```

import "oidl.idl";
import "ocidl.idl";

[
    object,
    oleautomation,
    uuid(FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B),
    helpstring("ITPCC Interface"),
    pointer_default(unique)
]
interface ITPCC : IUnknown
{
    HRESULT STDMETHODCALLTYPE NewOrder
        (
            [in] VARIANT txn_in,
            [out] VARIANT *txn_out
        );

    HRESULT STDMETHODCALLTYPE Payment
        (
            [in] VARIANT txn_in,
            [out] VARIANT *txn_out
        );

    HRESULT STDMETHODCALLTYPE Delivery
        (
            [in] VARIANT txn_in,
            [out] VARIANT *txn_out
        );

    HRESULT STDMETHODCALLTYPE StockLevel
        (
            [in] VARIANT txn_in,
            [out] VARIANT *txn_out
        );

    HRESULT STDMETHODCALLTYPE OrderStatus
        (
            [in] VARIANT txn_in,
            [out] VARIANT *txn_out
        );

    HRESULT STDMETHODCALLTYPE CallSetComplete
        (
            [in] VARIANT txn_in,
            [out] VARIANT *txn_out
        );
}; // interface ITPCC

```

tpcc_com_ps_i.c

```

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

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

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

/* File created by MIDL compiler version 5.03.0280 */

```

```

/* at Mon Jun 12 18:15:12 2000
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=i2), Wl, Zp8, env=Win32 (32b run), ms_ext, c_ext
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if !defined(_M_IA64) && !defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

```

```

#ifdef __cplusplus
}
#endif

#endif /* !defined(_M_IA64) && !defined(_M_AXP64) */

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

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

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

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jun 12 18:15:12 2000 */
/*
 * Compiler settings for .\src\tpcc_com_ps.idl:
 * Oicf (OptLev=i2), Wl, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext, robust
 * error checks: allocation ref bounds_check enum stub_data
 * VC __declspec() decoration level:
 *     __declspec(uuid()), __declspec(selectany), __declspec(novtable)
 *     DECLSPEC_UUID(), MIDL_INTERFACE()
 */
//@@MIDL_FILE_HEADING( )

#ifdef _M_IA64 || defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

```

```

} IID;

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif /* defined(_M_IA64) || defined(_M_AXP64) */

```

tpcc_com_ps_p.c

```

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

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

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jun 12 18:15:12 2000 */
/*
 * Compiler settings for .\src\tpcc_com_ps.idl:
 * Oicf (OptLev=i2), Wl, Zp8, env=Win32 (32b run), ms_ext, c_ext
 * error checks: allocation ref bounds_check enum stub_data
 * VC __declspec() decoration level:
 *     __declspec(uuid()), __declspec(selectany), __declspec(novtable)
 *     DECLSPEC_UUID(), MIDL_INTERFACE()
 */
//@@MIDL_FILE_HEADING( )

#ifndef _M_IA64 && !defined(_M_AXP64)
#define USE_STUBLESS_PROXY

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

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

```

```

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 997
#define PROC_FORMAT_STRING_SIZE 193
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 1

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

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

extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

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

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

```

```

0
};

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo =
{
    &Object_StubDesc,
    __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 *)-1 /* ITPCC::NewOrder */ ,
    (void *)-1 /* ITPCC::Payment */ ,
    (void *)-1 /* ITPCC::Delivery */ ,
    (void *)-1 /* ITPCC::StockLevel */ ,
    (void *)-1 /* ITPCC::OrderStatus */ ,
    (void *)-1 /* ITPCC::CallSetComplete */
};

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

extern const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ];

static const MIDL_STUB_DESC Object_StubDesc =
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.3.280 */
    0,
    UserMarshalRoutines,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* Reserved3 */
    0, /* Reserved4 */
    0 /* Reserved5 */
};

```

```

#pragma data_seg(".rdata")

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

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

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this stub because it uses these
features:
#error -Oif or -Oicf, [wire_marshall] or [user_marshall] 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 */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 8 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
                                NdrFcShort( 0x20 ), /* MIPS Stack size/offset = 32 */
#endif
#else
                                NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#else
                                NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 10 */ NdrFcShort( 0x0 ), /* 0 */
/* 12 */ NdrFcShort( 0x8 ), /* 8 */
/* 14 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has
return, */
                                0x3,          /* 3 */

        /* Parameter txn_in */

```

```

/* 16 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 18 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
                                NdrFcShort( 0x8 ), /* MIPS Stack size/offset = 8 */
#endif
#else
                                NdrFcShort( 0x8 ), /* PPC Stack size/offset = 8 */
#endif
#else
                                NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 20 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

        /* Parameter txn_out */

/* 22 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=16 */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 24 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
                                NdrFcShort( 0x18 ), /* MIPS Stack size/offset = 24 */
#endif
#else
                                NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
                                NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 26 */ NdrFcShort( 0x3da ), /* Type Offset=986 */

        /* Return value */

/* 28 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 30 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
                                NdrFcShort( 0x1c ), /* MIPS Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 32 */ 0x8, /* FC_LONG */
                                0x0, /* 0 */

        /* Procedure Payment */

/* 34 */ 0x33, /* FC_AUTO_HANDLE */
                                0x6c, /* Old Flags: object, Oi2 */
/* 36 */ NdrFcLong( 0x0 ), /* 0 */
/* 40 */ NdrFcShort( 0x4 ), /* 4 */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)

```

```

/* 42 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
                NdrFcShort( 0x20 ), /* MIPS Stack size/offset = 32 */
#endif
#else
                NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#else
                NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 44 */ NdrFcShort( 0x0 ), /* 0 */
/* 46 */ NdrFcShort( 0x8 ), /* 8 */
/* 48 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has
return, */
                0x3, /* 3 */

/* Parameter txn_in */

/* 50 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 52 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
                NdrFcShort( 0x8 ), /* MIPS Stack size/offset = 8 */
#endif
#else
                NdrFcShort( 0x8 ), /* PPC Stack size/offset = 8 */
#endif
#else
                NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 54 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Parameter txn_out */

/* 56 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 58 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
                NdrFcShort( 0x18 ), /* MIPS Stack size/offset = 24 */
#endif
#else
                NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
                NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 60 */ NdrFcShort( 0x3da ), /* Type Offset=986 */

/* Return value */

/* 62 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 64 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
                NdrFcShort( 0x1c ), /* MIPS Stack size/offset = 28 */
#endif
#endif
#endif

```

```

#else
                NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#else
                NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 66 */ 0x8, /* FC_LONG */
                0x0, /* 0 */

/* Procedure Delivery */

/* 68 */ 0x33, /* FC_AUTO_HANDLE */
                0x6c, /* Old Flags: object, Oi2 */
/* 70 */ NdrFcLong( 0x0 ), /* 0 */
/* 74 */ NdrFcShort( 0x5 ), /* 5 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 76 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
                NdrFcShort( 0x20 ), /* MIPS Stack size/offset = 32 */
#endif
#else
                NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#else
                NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 78 */ NdrFcShort( 0x0 ), /* 0 */
/* 80 */ NdrFcShort( 0x8 ), /* 8 */
/* 82 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has
return, */
                0x3, /* 3 */

/* Parameter txn_in */

/* 84 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 86 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
                NdrFcShort( 0x8 ), /* MIPS Stack size/offset = 8 */
#endif
#else
                NdrFcShort( 0x8 ), /* PPC Stack size/offset = 8 */
#endif
#else
                NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 88 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Parameter txn_out */

/* 90 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 92 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
                NdrFcShort( 0x18 ), /* MIPS Stack size/offset = 24 */
#endif
#endif
#endif

```



```

#else
                                NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
                                NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 94 */ NdrFcShort( 0x3da ), /* Type Offset=986 */
/* Return value */
/* 96 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 98 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
                                NdrFcShort( 0x1c ), /* MIPS Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 100 */ 0x8, /* FC_LONG */
0x0, /* 0 */
/* Procedure StockLevel */
/* 102 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 104 */ NdrFcLong( 0x0 ), /* 0 */
/* 108 */ NdrFcShort( 0x6 ), /* 6 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 110 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
                                NdrFcShort( 0x20 ), /* MIPS Stack size/offset = 32 */
#endif
#else
                                NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#else
                                NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 112 */ NdrFcShort( 0x0 ), /* 0 */
/* 114 */ NdrFcShort( 0x8 ), /* 8 */
/* 116 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has
return, */
0x3, /* 3 */
/* Parameter txn_in */
/* 118 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 120 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
                                NdrFcShort( 0x8 ), /* MIPS Stack size/offset = 8 */
#endif
#endif
#else

```

```

                                NdrFcShort( 0x8 ), /* PPC Stack size/offset = 8 */
#endif
#else
                                NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 122 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */
/* Parameter txn_out */
/* 124 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 126 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
                                NdrFcShort( 0x18 ), /* MIPS Stack size/offset = 24 */
#endif
#else
                                NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
                                NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 128 */ NdrFcShort( 0x3da ), /* Type Offset=986 */
/* Return value */
/* 130 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 132 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
                                NdrFcShort( 0x1c ), /* MIPS Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 134 */ 0x8, /* FC_LONG */
0x0, /* 0 */
/* Procedure OrderStatus */
/* 136 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 138 */ NdrFcLong( 0x0 ), /* 0 */
/* 142 */ NdrFcShort( 0x7 ), /* 7 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 144 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
                                NdrFcShort( 0x20 ), /* MIPS Stack size/offset = 32 */
#endif
#else
                                NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#else
                                NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */

```

```

#endif
/* 146 */ NdrFcShort( 0x0 ), /* 0 */
/* 148 */ NdrFcShort( 0x8 ), /* 8 */
/* 150 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has
return, */
0x3, /* 3 */

/* Parameter txn_in */

/* 152 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifndef ALPHA_
#ifndef PPC_
#ifndef MIPS_
/* 154 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
NdrFcShort( 0x8 ), /* MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 156 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Parameter txn_out */

/* 158 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=16 */
#ifndef ALPHA_
#ifndef PPC_
#ifndef MIPS_
/* 160 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
NdrFcShort( 0x18 ), /* MIPS Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 162 */ NdrFcShort( 0x3da ), /* Type Offset=986 */

/* Return value */

/* 164 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef ALPHA_
#ifndef PPC_
#ifndef MIPS_
/* 166 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
NdrFcShort( 0x1c ), /* MIPS Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 168 */ 0x8, /* FC_LONG */
0x0, /* 0 */

```

```

/* Procedure CallSetComplete */

/* 170 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */

/* 172 */ NdrFcLong( 0x0 ), /* 0 */
/* 176 */ NdrFcShort( 0x8 ), /* 8 */
#ifndef ALPHA_
/* 178 */ NdrFcShort( 0x8 ), /* x86, MIPS, PPC Stack size/offset = 8 */
#else
NdrFcShort( 0x10 ), /* Alpha Stack size/offset = 16 */
#endif
/* 180 */ NdrFcShort( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x8 ), /* 8 */
/* 184 */ 0x4, /* Oi2 Flags: has return, */
0x1, /* 1 */

/* Return value */

/* 186 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef ALPHA_
/* 188 */ NdrFcShort( 0x4 ), /* x86, MIPS, PPC Stack size/offset = 4 */
#else
NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 190 */ 0x8, /* FC_LONG */
0x0, /* 0 */

}
};

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
0,
{
NdrFcShort( 0x0 ), /* 0 */

/* 2 */
0x12, 0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0x3b0 ), /* Offset= 944 (948) */
/* 6 */
0x2b, /* FC_NON ENCAPSULATED_UNION */
0x9, /* FC_ULONG */
/* 8 */ 0x7, /* Corr desc: FC_USHORT */
0x0, /* */

/* 10 */ NdrFcShort( 0xffff8 ), /* -8 */
/* 12 */ NdrFcShort( 0x2 ), /* Offset= 2 (14) */
/* 14 */ NdrFcShort( 0x10 ), /* 16 */
/* 16 */ NdrFcShort( 0x2b ), /* 43 */
/* 18 */ NdrFcLong( 0x3 ), /* 3 */
/* 22 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 24 */ NdrFcLong( 0x11 ), /* 17 */
/* 28 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE */
/* 30 */ NdrFcLong( 0x2 ), /* 2 */
/* 34 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 36 */ NdrFcLong( 0x4 ), /* 4 */
/* 40 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT */
/* 42 */ NdrFcLong( 0x5 ), /* 5 */
/* 46 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 48 */ NdrFcLong( 0xb ), /* 11 */
/* 52 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 54 */ NdrFcLong( 0xa ), /* 10 */
/* 58 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 60 */ NdrFcLong( 0x6 ), /* 6 */

```

```

/* 64 */ NdrFcShort( 0xd6 ), /* Offset= 214 (278) */
/* 66 */ NdrFcLong( 0x7 ), /* 7 */
/* 70 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 72 */ NdrFcLong( 0x8 ), /* 8 */
/* 76 */ NdrFcShort( 0xd0 ), /* Offset= 208 (284) */
/* 78 */ NdrFcLong( 0xd ), /* 13 */
/* 82 */ NdrFcShort( 0xe2 ), /* Offset= 226 (308) */
/* 84 */ NdrFcLong( 0x9 ), /* 9 */
/* 88 */ NdrFcShort( 0xee ), /* Offset= 238 (326) */
/* 90 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 94 */ NdrFcShort( 0xfa ), /* Offset= 250 (344) */
/* 96 */ NdrFcLong( 0x24 ), /* 36 */
/* 100 */ NdrFcShort( 0x308 ), /* Offset= 776 (876) */
/* 102 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 106 */ NdrFcShort( 0x302 ), /* Offset= 770 (876) */
/* 108 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 112 */ NdrFcShort( 0x300 ), /* Offset= 768 (880) */
/* 114 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 118 */ NdrFcShort( 0x2fe ), /* Offset= 766 (884) */
/* 120 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 124 */ NdrFcShort( 0x2fc ), /* Offset= 764 (888) */
/* 126 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 130 */ NdrFcShort( 0x2fa ), /* Offset= 762 (892) */
/* 132 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 136 */ NdrFcShort( 0x2f8 ), /* Offset= 760 (896) */
/* 138 */ NdrFcLong( 0x400b ), /* 16395 */
/* 142 */ NdrFcShort( 0x2e6 ), /* Offset= 742 (884) */
/* 144 */ NdrFcLong( 0x400a ), /* 16394 */
/* 148 */ NdrFcShort( 0x2e4 ), /* Offset= 740 (888) */
/* 150 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 154 */ NdrFcShort( 0x2ea ), /* Offset= 746 (900) */
/* 156 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 160 */ NdrFcShort( 0x2e0 ), /* Offset= 736 (896) */
/* 162 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 166 */ NdrFcShort( 0x2e2 ), /* Offset= 738 (904) */
/* 168 */ NdrFcLong( 0x400d ), /* 16397 */
/* 172 */ NdrFcShort( 0x2e0 ), /* Offset= 736 (908) */
/* 174 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 178 */ NdrFcShort( 0x2de ), /* Offset= 734 (912) */
/* 180 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 184 */ NdrFcShort( 0x2dc ), /* Offset= 732 (916) */
/* 186 */ NdrFcLong( 0x400c ), /* 16396 */
/* 190 */ NdrFcShort( 0x2da ), /* Offset= 730 (920) */
/* 192 */ NdrFcLong( 0x10 ), /* 16 */
/* 196 */ NdrFcShort( 0x8002 ), /* Simple arm type: FC_CHAR */
/* 198 */ NdrFcLong( 0x12 ), /* 18 */
/* 202 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 204 */ NdrFcLong( 0x13 ), /* 19 */
/* 208 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 210 */ NdrFcLong( 0x16 ), /* 22 */
/* 214 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 216 */ NdrFcLong( 0x17 ), /* 23 */
/* 220 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 222 */ NdrFcLong( 0xe ), /* 14 */
/* 226 */ NdrFcShort( 0x2be ), /* Offset= 702 (928) */
/* 228 */ NdrFcLong( 0x400e ), /* 16398 */
/* 232 */ NdrFcShort( 0x2c4 ), /* Offset= 708 (940) */
/* 234 */ NdrFcLong( 0x4010 ), /* 16400 */
/* 238 */ NdrFcShort( 0x2c2 ), /* Offset= 706 (944) */
/* 240 */ NdrFcLong( 0x4012 ), /* 16402 */
/* 244 */ NdrFcShort( 0x280 ), /* Offset= 640 (884) */
/* 246 */ NdrFcLong( 0x4013 ), /* 16403 */
/* 250 */ NdrFcShort( 0x27e ), /* Offset= 638 (888) */

```

```

/* 252 */ NdrFcLong( 0x4016 ), /* 16406 */
/* 256 */ NdrFcShort( 0x278 ), /* Offset= 632 (888) */
/* 258 */ NdrFcLong( 0x4017 ), /* 16407 */
/* 262 */ NdrFcShort( 0x272 ), /* Offset= 626 (888) */
/* 264 */ NdrFcLong( 0x0 ), /* 0 */
/* 268 */ NdrFcShort( 0x0 ), /* Offset= 0 (268) */
/* 270 */ NdrFcLong( 0x1 ), /* 1 */
/* 274 */ NdrFcShort( 0x0 ), /* Offset= 0 (274) */
/* 276 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (275) */
/* 278 */
/* 280 */ NdrFcShort( 0x8 ), /* 8 */
/* 282 */ 0xb, /* FC_HYPER */
/* 284 */ 0x5b, /* FC_END */
/* 286 */ NdrFcShort( 0xc ), /* Offset= 12 (298) */
/* 288 */
/* 290 */ NdrFcShort( 0x2 ), /* 2 */
/* 292 */ 0x9, /* Corr desc: FC_ULONG */
/* 294 */ NdrFcShort( 0xfffc ), /* -4 */
/* 296 */ 0x6, /* FC_SHORT */
/* 298 */ 0x5b, /* FC_END */
/* 300 */ NdrFcShort( 0x8 ), /* 8 */
/* 302 */ NdrFcShort( 0xffffffff2 ), /* Offset= -14 (288) */
/* 304 */ 0x8, /* FC_LONG */
/* 306 */ 0x5c, /* FC_PAD */
/* 308 */ 0x5b, /* FC_END */
/* 310 */ NdrFcLong( 0x0 ), /* 0 */
/* 314 */ NdrFcShort( 0x0 ), /* 0 */
/* 316 */ NdrFcShort( 0x0 ), /* 0 */
/* 318 */ 0xc0, /* 192 */
/* 320 */ 0x0, /* 0 */
/* 322 */ 0x0, /* 0 */
/* 324 */ 0x0, /* 0 */
/* 326 */ 0x46, /* 70 */
/* 328 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 332 */ NdrFcShort( 0x0 ), /* 0 */
/* 334 */ NdrFcShort( 0x0 ), /* 0 */
/* 336 */ 0xc0, /* 192 */
/* 338 */ 0x0, /* 0 */
/* 340 */ 0x0, /* 0 */
/* 342 */ 0x0, /* 0 */

```

```

/* 344 */                                0x46,                                /* 70 */
/* 346 */ NdrFcShort( 0x2 ), /* FC_UP [pointer_deref] */
/* 348 */                                /* Offset= 2 (348) */
/* 350 */ NdrFcShort( 0x1fc ), /* FC_UP */
/* 352 */                                /* Offset= 508 (858) */
/* 354 */ NdrFcShort( 0x18 ), /* FC_ENCAPSULATED_UNION */
/* 356 */ NdrFcShort( 0xa ), /* FC_PP */
/* 358 */ NdrFcLong( 0x8 ), /* FC_PAD */
/* 362 */ NdrFcShort( 0x58 ), /* FC_FIXED_OFFSET */
/* 364 */ NdrFcLong( 0xd ), /* FC_VARIABLE_REPEAT */
/* 368 */ NdrFcShort( 0x78 ), /* FC_FIXED_OFFSET */
/* 370 */ NdrFcLong( 0x9 ), /* FC_FIXED_OFFSET */
/* 374 */ NdrFcShort( 0x94 ), /* FC_FIXED_OFFSET */
/* 376 */ NdrFcLong( 0xc ), /* FC_FIXED_OFFSET */
/* 380 */ NdrFcShort( 0xbc ), /* FC_FIXED_OFFSET */
/* 382 */ NdrFcLong( 0x24 ), /* FC_FIXED_OFFSET */
/* 386 */ NdrFcShort( 0x114 ), /* FC_FIXED_OFFSET */
/* 388 */ NdrFcLong( 0x800d ), /* FC_FIXED_OFFSET */
/* 392 */ NdrFcShort( 0x130 ), /* FC_FIXED_OFFSET */
/* 394 */ NdrFcLong( 0x10 ), /* FC_FIXED_OFFSET */
/* 398 */ NdrFcShort( 0x148 ), /* FC_FIXED_OFFSET */
/* 400 */ NdrFcLong( 0x2 ), /* FC_FIXED_OFFSET */
/* 404 */ NdrFcShort( 0x160 ), /* FC_FIXED_OFFSET */
/* 406 */ NdrFcLong( 0x3 ), /* FC_FIXED_OFFSET */
/* 410 */ NdrFcShort( 0x178 ), /* FC_FIXED_OFFSET */
/* 412 */ NdrFcLong( 0x14 ), /* FC_FIXED_OFFSET */
/* 416 */ NdrFcShort( 0x190 ), /* FC_FIXED_OFFSET */
/* 418 */ NdrFcShort( 0xffffffff ), /* FC_FIXED_OFFSET */
/* 420 */
/* 422 */ NdrFcShort( 0x4 ), /* FC_CARRY */
/* 424 */ 0x19, /* FC_CARRY */
/* 426 */ NdrFcShort( 0x0 ), /* FC_CARRY */
/* 428 */
/* 430 */                                0x4b,                                /* FC_PP */
/* 432 */ NdrFcShort( 0x4 ), /* FC_PP */
/* 434 */ NdrFcShort( 0x0 ), /* FC_PAD */
/* 436 */ NdrFcShort( 0x1 ), /* FC_PAD */
/* 438 */ NdrFcShort( 0x0 ), /* FC_PAD */
/* 440 */ NdrFcShort( 0x0 ), /* FC_PAD */
/* 442 */ 0x12, 0x0, /* FC_UP */
/* 444 */ NdrFcShort( 0xffffffff6e ), /* FC_UP */
/* 446 */                                0x5b,                                /* FC_END */
/* 448 */ 0x5c, /* FC_LONG */
/* 450 */                                0x5b,                                /* FC_END */
/* 452 */                                0x16,                                /* FC_PSTRUCT */
/* 454 */ NdrFcShort( 0x8 ), /* FC_PSTRUCT */
/* 456 */                                0x3,                                /* FC_PSTRUCT */
/* 458 */                                0x8,                                /* FC_PSTRUCT */
/* 460 */                                0x3,                                /* FC_PSTRUCT */
/* 462 */                                0x5c,                                /* FC_PSTRUCT */
/* 464 */                                0x5b,                                /* FC_PSTRUCT */
/* 466 */                                0x5c,                                /* FC_PSTRUCT */
/* 468 */                                0x5b,                                /* FC_PSTRUCT */
/* 470 */                                0x5c,                                /* FC_PSTRUCT */
/* 472 */                                0x21,                                /* FC_BOGUS_ARRAY */
/* 474 */ 0x19, /* FC_BOGUS_ARRAY */
/* 476 */ NdrFcShort( 0x0 ), /* FC_BOGUS_ARRAY */
/* 478 */ NdrFcLong( 0xffffffff ), /* FC_BOGUS_ARRAY */
/* 480 */ 0x4c, /* FC_BOGUS_ARRAY */
/* 482 */ NdrFcShort( 0xffff50 ), /* FC_BOGUS_ARRAY */
/* 484 */ 0x5c, /* FC_BOGUS_ARRAY */
/* 486 */                                0x5b,                                /* FC_BOGUS_ARRAY */
/* 488 */                                0x1a,                                /* FC_BOGUS_STRUCT */
/* 490 */ NdrFcShort( 0x8 ), /* FC_BOGUS_STRUCT */
/* 492 */ NdrFcShort( 0x0 ), /* FC_BOGUS_STRUCT */
/* 494 */ NdrFcShort( 0x6 ), /* FC_BOGUS_STRUCT */
/* 496 */ 0x8, /* FC_BOGUS_STRUCT */
/* 498 */ 0x5c, /* FC_BOGUS_STRUCT */
/* 500 */                                0x5b,                                /* FC_BOGUS_STRUCT */
/* 502 */ NdrFcShort( 0xfffffe0 ), /* FC_BOGUS_STRUCT */
/* 504 */                                0x21,                                /* FC_BOGUS_ARRAY */
/* 506 */ NdrFcShort( 0x0 ), /* FC_BOGUS_ARRAY */
/* 508 */ 0x19, /* FC_BOGUS_ARRAY */
/* 510 */ NdrFcShort( 0x0 ), /* FC_BOGUS_ARRAY */
/* 512 */ NdrFcLong( 0xffffffff ), /* FC_BOGUS_ARRAY */
/* 514 */ 0x4c, /* FC_BOGUS_ARRAY */
/* 516 */ NdrFcShort( 0xffff40 ), /* FC_BOGUS_ARRAY */
/* 518 */ NdrFcShort( 0x5c ), /* FC_BOGUS_ARRAY */
/* 520 */                                0x5b,                                /* FC_BOGUS_ARRAY */
/* 522 */                                0x1a,                                /* FC_BOGUS_STRUCT */
/* 524 */ NdrFcShort( 0x8 ), /* FC_BOGUS_STRUCT */
/* 526 */ NdrFcShort( 0x0 ), /* FC_BOGUS_STRUCT */
/* 528 */ NdrFcShort( 0x6 ), /* FC_BOGUS_STRUCT */
/* 530 */ 0x8, /* FC_BOGUS_STRUCT */
/* 532 */                                0x36,                                /* FC_POINTER */
/* 534 */                                0x36,                                /* FC_POINTER */

```

```

/* 454 */                                0x4b,                                /* FC_PP */
/* 456 */                                0x5c,                                /* FC_PAD */
/* 458 */ NdrFcShort( 0x4 ), /* FC_NO_REPEAT */
/* 460 */ NdrFcShort( 0x4 ), /* FC_NO_REPEAT */
/* 462 */ 0x11, 0x0, /* FC_RP */
/* 464 */ NdrFcShort( 0xffffffff4 ), /* FC_RP */
/* 466 */                                0x5b,                                /* FC_END */
/* 468 */ 0x8, /* FC_LONG */
/* 470 */                                0x5b,                                /* FC_END */
/* 472 */                                0x21,                                /* FC_BOGUS_ARRAY */
/* 474 */ 0x3, /* FC_BOGUS_ARRAY */
/* 476 */ NdrFcShort( 0x0 ), /* FC_BOGUS_ARRAY */
/* 478 */ 0x19, /* FC_BOGUS_ARRAY */
/* 480 */ 0x0, /* FC_BOGUS_ARRAY */
/* 482 */ 0x4c, /* FC_BOGUS_ARRAY */
/* 484 */ NdrFcShort( 0xffff50 ), /* FC_BOGUS_ARRAY */
/* 486 */ 0x5c, /* FC_BOGUS_ARRAY */
/* 488 */                                0x5b,                                /* FC_BOGUS_ARRAY */
/* 490 */                                0x1a,                                /* FC_BOGUS_STRUCT */
/* 492 */ NdrFcShort( 0x8 ), /* FC_BOGUS_STRUCT */
/* 494 */ NdrFcShort( 0x0 ), /* FC_BOGUS_STRUCT */
/* 496 */ NdrFcShort( 0x6 ), /* FC_BOGUS_STRUCT */
/* 498 */ 0x8, /* FC_BOGUS_STRUCT */
/* 500 */                                0x36,                                /* FC_POINTER */
/* 502 */ 0x5c, /* FC_POINTER */
/* 504 */                                0x5b,                                /* FC_POINTER */
/* 506 */                                0x21,                                /* FC_BOGUS_ARRAY */
/* 508 */ NdrFcShort( 0x0 ), /* FC_BOGUS_ARRAY */
/* 510 */ 0x19, /* FC_BOGUS_ARRAY */
/* 512 */ NdrFcShort( 0x0 ), /* FC_BOGUS_ARRAY */
/* 514 */ NdrFcLong( 0xffffffff ), /* FC_BOGUS_ARRAY */
/* 516 */ 0x4c, /* FC_BOGUS_ARRAY */
/* 518 */ NdrFcShort( 0xffff40 ), /* FC_BOGUS_ARRAY */
/* 520 */ 0x5c, /* FC_BOGUS_ARRAY */
/* 522 */                                0x5b,                                /* FC_BOGUS_ARRAY */
/* 524 */                                0x1a,                                /* FC_BOGUS_STRUCT */
/* 526 */ NdrFcShort( 0x8 ), /* FC_BOGUS_STRUCT */
/* 528 */ NdrFcShort( 0x0 ), /* FC_BOGUS_STRUCT */
/* 530 */ NdrFcShort( 0x6 ), /* FC_BOGUS_STRUCT */
/* 532 */ 0x8, /* FC_BOGUS_STRUCT */
/* 534 */                                0x36,                                /* FC_POINTER */

```

```

/* 532 */ 0x5c,          /* FC_PAD */
/* 534 */          0x5b,          /* FC_END */
/* 536 */ NdrFcShort( 0xffffffe0 ), /* Offset= -32 (504) */
/* 538 */          0x1b,          /* FC_CARRY */
/* 540 */ NdrFcShort( 0x4 ), /* 4 */
/* 542 */ 0x19,          /* Corr desc: field pointer, FC_ULONG */
/* 544 */ NdrFcShort( 0x0 ), /* 0 */
/* 546 */          0x4b,          /* FC_PP */
/* 548 */          0x5c,          /* FC_PAD */
/* 550 */ NdrFcShort( 0x4 ), /* 4 */
/* 552 */ NdrFcShort( 0x0 ), /* 0 */
/* 554 */ NdrFcShort( 0x1 ), /* 1 */
/* 556 */ NdrFcShort( 0x0 ), /* 0 */
/* 558 */ NdrFcShort( 0x0 ), /* 0 */
/* 560 */ 0x12, 0x0,      /* FC_UP */
/* 562 */ NdrFcShort( 0x182 ), /* Offset= 386 (948) */
/* 564 */          0x5b,          /* FC_END */
/* 566 */ 0x5c,          /* FC_LONG */
/* 568 */          0x5b,          /* FC_END */
/* 570 */ NdrFcShort( 0x8 ), /* 8 */
/* 572 */ NdrFcShort( 0x0 ), /* 0 */
/* 574 */ NdrFcShort( 0x6 ), /* Offset= 6 (580) */
/* 576 */ 0x8,          /* FC_LONG */
/* 578 */ 0x5c,          /* FC_POINTER */
/* 580 */          0x5b,          /* FC_PAD */
/* 582 */ NdrFcShort( 0x11, 0x0, /* FC_PP */
/* 584 */ NdrFcShort( 0xffffffd4 ), /* Offset= -44 (538) */
/* 586 */          0x2f,          /* FC_IP */
/* 588 */          0x5a,          /* FC_CONSTANT_IID */
/* 590 */ NdrFcLong( 0x2f ), /* 47 */
/* 592 */ NdrFcShort( 0x0 ), /* 0 */
/* 594 */ NdrFcShort( 0x0 ), /* 0 */
/* 596 */ 0xc0,          /* 192 */
/* 598 */ 0x0,          /* 0 */
/* 600 */ 0x0,          /* 0 */
/* 602 */ 0x0,          /* 0 */
/* 604 */ 0x46,          /* 70 */
/* 606 */          0x1b,          /* FC_CARRY */
/* 608 */          0x0,          /* 0 */
/* 610 */ NdrFcShort( 0x1 ), /* 1 */
/* 612 */ 0x19,          /* Corr desc: field pointer, FC_ULONG */

```

```

/* 608 */ NdrFcShort( 0x4 ), /* 4 */
/* 610 */ 0x1,          /* FC_BYTE */
/* 612 */          0x5b,          /* FC_END */
/* 614 */          0x1a,          /* FC_BOGUS_STRUCT */
/* 616 */          0x3,          /* 3 */
/* 618 */ NdrFcShort( 0x10 ), /* 16 */
/* 620 */ NdrFcShort( 0x0 ), /* 0 */
/* 622 */ 0xa,          /* Offset= 10 (628) */
/* 624 */ 0x8,          /* FC_LONG */
/* 626 */ 0x4c,          /* FC_LONG */
/* 628 */          0x8,          /* FC_LONG */
/* 630 */          0x0,          /* FC_EMBEDDED_COMPLEX */
/* 632 */ NdrFcShort( 0xffffffd8 ), /* Offset= -40 (584) */
/* 634 */ 0x36,          /* FC_POINTER */
/* 636 */          0x5b,          /* FC_END */
/* 638 */          0x12, 0x0,      /* FC_UP */
/* 640 */ NdrFcShort( 0xffffffe4 ), /* Offset= -28 (602) */
/* 642 */          0x1b,          /* FC_CARRY */
/* 644 */          0x3,          /* 3 */
/* 646 */ NdrFcShort( 0x4 ), /* 4 */
/* 648 */ 0x19,          /* Corr desc: field pointer, FC_ULONG */
/* 650 */ NdrFcShort( 0x0 ), /* 0 */
/* 652 */ NdrFcShort( 0x0 ), /* 0 */
/* 654 */ 0x12, 0x0,      /* FC_UP */
/* 656 */ NdrFcShort( 0xfffffd4 ), /* Offset= -44 (612) */
/* 658 */          0x5b,          /* FC_END */
/* 660 */ 0x5c,          /* FC_LONG */
/* 662 */          0x5b,          /* FC_PAD */
/* 664 */          0x1a,          /* FC_BOGUS_STRUCT */
/* 666 */          0x3,          /* 3 */
/* 668 */ NdrFcShort( 0x8 ), /* 8 */
/* 670 */ NdrFcShort( 0x0 ), /* 0 */
/* 672 */ NdrFcShort( 0x6 ), /* Offset= 6 (674) */
/* 674 */ 0x8,          /* FC_LONG */
/* 676 */ 0x5c,          /* FC_POINTER */
/* 678 */          0x36,          /* FC_PAD */
/* 680 */          0x5b,          /* FC_END */
/* 682 */          0x11, 0x0,      /* FC_PP */
/* 684 */ NdrFcShort( 0xffffffd4 ), /* Offset= -44 (632) */
/* 686 */          0x1d,          /* FC_SMFARRAY */
/* 688 */          0x0,          /* 0 */
/* 690 */ NdrFcShort( 0x8 ), /* 8 */

```

```

/* 682 */ 0x2,          /* FC_CHAR */
/* 684 */          0x5b,          /* FC_END */
/* 686 */          0x15,          /* FC_STRUCT */
/* 688 */ 0x8,          0x3,          /* 3 */
/* 690 */ NdrFcShort( 0x10 ), /* 16 */
/* 692 */ 0x6,          /* FC_SHORT */
/* 694 */ 0x4c,          /* FC_EMBEDDED_COMPLEX */
/* 696 */ 0x0,          /* 0 */
/* 698 */ NdrFcShort( 0xfffffff1 ), /* Offset= -15 (678) */
/* 700 */ 0x5b,          /* FC_END */
/* 702 */          0x1a,          /* FC_BOGUS_STRUCT */
/* 704 */          0x3,          /* 3 */
/* 706 */ NdrFcShort( 0x18 ), /* 24 */
/* 708 */ NdrFcShort( 0x0 ), /* 0 */
/* 710 */ NdrFcShort( 0xa ), /* Offset= 10 (712) */
/* 712 */ 0x8,          /* FC_LONG */
/* 714 */          0x36,          /* FC_POINTER */
/* 716 */          /* FC_EMBEDDED_COMPLEX */
/* 718 */ NdrFcShort( 0xffffffe8 ), /* Offset= -24 (684) */
/* 720 */ 0x5c,          /* FC_PAD */
/* 722 */          0x5b,          /* FC_END */
/* 724 */          0x11, 0x0,      /* FC_RP */
/* 726 */ NdrFcShort( 0xfffffff0c ), /* Offset= -244 (470) */
/* 728 */          0x1b,          /* FC_CARRAY */
/* 730 */          0x0,          /* 0 */
/* 732 */ NdrFcShort( 0x1 ), /* 1 */
/* 734 */ 0x19,          /* Corr desc: field pointer, FC_ULONG */
/* 736 */          0x0,          /* */
/* 738 */ NdrFcShort( 0x0 ), /* 0 */
/* 740 */ 0x1,          /* FC_BYTE */
/* 742 */          0x5b,          /* FC_END */
/* 744 */          0x16,          /* FC_PSTRUCT */
/* 746 */          0x3,          /* 3 */
/* 748 */ NdrFcShort( 0x8 ), /* 8 */
/* 750 */          0x4b,          /* FC_PP */
/* 752 */          0x5c,          /* FC_PAD */
/* 754 */          0x46,          /* FC_NO_REPEAT */
/* 756 */          0x5c,          /* FC_PAD */
/* 758 */          0x16,          /* FC_PSTRUCT */
/* 760 */          0x3,          /* 3 */
/* 762 */          0x4b,          /* FC_PP */
/* 764 */          0x5c,          /* FC_PAD */
/* 766 */          0x46,          /* FC_NO_REPEAT */
/* 768 */          0x5c,          /* FC_PAD */
/* 770 */          0x46,          /* FC_NO_REPEAT */
/* 772 */          0x5c,          /* FC_PAD */
/* 774 */          0x8,          /* FC_LONG */
/* 776 */          0x5b,          /* FC_END */
/* 778 */          0x1b,          /* FC_CARRAY */
/* 780 */          0x3,          /* 3 */
/* 782 */ NdrFcShort( 0x4 ), /* 4 */
/* 784 */ 0x19,          /* Corr desc: field pointer, FC_ULONG */
/* 786 */          0x0,          /* */
/* 788 */ NdrFcShort( 0x0 ), /* 0 */
/* 790 */ 0x8,          /* FC_LONG */
/* 792 */          0x5b,          /* FC_END */
/* 794 */          0x16,          /* FC_PSTRUCT */
/* 796 */          0x3,          /* 3 */
/* 798 */ NdrFcShort( 0x8 ), /* 8 */
/* 800 */          0x4b,          /* FC_PP */
/* 802 */          0x5c,          /* FC_PAD */
/* 804 */          0x46,          /* FC_NO_REPEAT */
/* 806 */          0x5c,          /* FC_PAD */
/* 808 */          0x16,          /* FC_PSTRUCT */
/* 810 */          0x3,          /* 3 */
/* 812 */          0x4b,          /* FC_PP */
/* 814 */          0x5c,          /* FC_PAD */
/* 816 */          0x46,          /* FC_NO_REPEAT */
/* 818 */          0x5c,          /* FC_PAD */
/* 820 */          0x16,          /* FC_PSTRUCT */
/* 822 */          0x3,          /* 3 */
/* 824 */          0x4b,          /* FC_PP */
/* 826 */          0x5c,          /* FC_PAD */
/* 828 */          0x46,          /* FC_NO_REPEAT */
/* 830 */          0x5c,          /* FC_PAD */
/* 832 */          0x16,          /* FC_PSTRUCT */
/* 834 */          0x3,          /* 3 */
/* 836 */          0x4b,          /* FC_PP */
/* 838 */          0x5c,          /* FC_PAD */
/* 840 */          0x46,          /* FC_NO_REPEAT */
/* 842 */          0x5c,          /* FC_PAD */
/* 844 */          0x16,          /* FC_PSTRUCT */
/* 846 */          0x3,          /* 3 */
/* 848 */          0x4b,          /* FC_PP */
/* 850 */          0x5c,          /* FC_PAD */
/* 852 */          0x46,          /* FC_NO_REPEAT */
/* 854 */          0x5c,          /* FC_PAD */
/* 856 */          0x16,          /* FC_PSTRUCT */
/* 858 */          0x3,          /* 3 */
/* 860 */          0x4b,          /* FC_PP */
/* 862 */          0x5c,          /* FC_PAD */
/* 864 */          0x46,          /* FC_NO_REPEAT */
/* 866 */          0x5c,          /* FC_PAD */
/* 868 */          0x16,          /* FC_PSTRUCT */
/* 870 */          0x3,          /* 3 */
/* 872 */          0x4b,          /* FC_PP */
/* 874 */          0x5c,          /* FC_PAD */
/* 876 */          0x46,          /* FC_NO_REPEAT */
/* 878 */          0x5c,          /* FC_PAD */
/* 880 */          0x16,          /* FC_PSTRUCT */
/* 882 */          0x3,          /* 3 */
/* 884 */          0x4b,          /* FC_PP */
/* 886 */          0x5c,          /* FC_PAD */
/* 888 */          0x46,          /* FC_NO_REPEAT */
/* 890 */          0x5c,          /* FC_PAD */
/* 892 */          0x16,          /* FC_PSTRUCT */
/* 894 */          0x3,          /* 3 */
/* 896 */          0x4b,          /* FC_PP */
/* 898 */          0x5c,          /* FC_PAD */
/* 900 */          0x46,          /* FC_NO_REPEAT */
/* 902 */          0x5c,          /* FC_PAD */
/* 904 */          0x16,          /* FC_PSTRUCT */
/* 906 */          0x3,          /* 3 */
/* 908 */          0x4b,          /* FC_PP */
/* 910 */          0x5c,          /* FC_PAD */
/* 912 */          0x46,          /* FC_NO_REPEAT */
/* 914 */          0x5c,          /* FC_PAD */
/* 916 */          0x16,          /* FC_PSTRUCT */
/* 918 */          0x3,          /* 3 */
/* 920 */          0x4b,          /* FC_PP */
/* 922 */          0x5c,          /* FC_PAD */
/* 924 */          0x46,          /* FC_NO_REPEAT */
/* 926 */          0x5c,          /* FC_PAD */
/* 928 */          0x16,          /* FC_PSTRUCT */
/* 930 */          0x3,          /* 3 */
/* 932 */          0x4b,          /* FC_PP */
/* 934 */          0x5c,          /* FC_PAD */
/* 936 */          0x46,          /* FC_NO_REPEAT */
/* 938 */          0x5c,          /* FC_PAD */
/* 940 */          0x16,          /* FC_PSTRUCT */
/* 942 */          0x3,          /* 3 */
/* 944 */          0x4b,          /* FC_PP */
/* 946 */          0x5c,          /* FC_PAD */
/* 948 */          0x46,          /* FC_NO_REPEAT */
/* 950 */          0x5c,          /* FC_PAD */
/* 952 */          0x16,          /* FC_PSTRUCT */
/* 954 */          0x3,          /* 3 */
/* 956 */          0x4b,          /* FC_PP */
/* 958 */          0x5c,          /* FC_PAD */
/* 960 */          0x46,          /* FC_NO_REPEAT */
/* 962 */          0x5c,          /* FC_PAD */
/* 964 */          0x16,          /* FC_PSTRUCT */
/* 966 */          0x3,          /* 3 */
/* 968 */          0x4b,          /* FC_PP */
/* 970 */          0x5c,          /* FC_PAD */
/* 972 */          0x46,          /* FC_NO_REPEAT */
/* 974 */          0x5c,          /* FC_PAD */
/* 976 */          0x16,          /* FC_PSTRUCT */
/* 978 */          0x3,          /* 3 */
/* 980 */          0x4b,          /* FC_PP */
/* 982 */          0x5c,          /* FC_PAD */
/* 984 */          0x46,          /* FC_NO_REPEAT */
/* 986 */          0x5c,          /* FC_PAD */
/* 988 */          0x16,          /* FC_PSTRUCT */
/* 990 */          0x3,          /* 3 */
/* 992 */          0x4b,          /* FC_PP */
/* 994 */          0x5c,          /* FC_PAD */
/* 996 */          0x46,          /* FC_NO_REPEAT */
/* 998 */          0x5c,          /* FC_PAD */

```

```

/* 752 */ NdrFcShort( 0x0 ), /* 0 */
/* 754 */ 0x6,          /* FC_SHORT */
/* 756 */          0x5b,          /* FC_END */
/* 758 */          0x16,          /* FC_PSTRUCT */
/* 760 */          0x3,          /* 3 */
/* 762 */ NdrFcShort( 0x8 ), /* 8 */
/* 764 */          0x4b,          /* FC_PP */
/* 766 */          0x5c,          /* FC_PAD */
/* 768 */          0x46,          /* FC_NO_REPEAT */
/* 770 */          0x5c,          /* FC_PAD */
/* 772 */          0x46,          /* FC_NO_REPEAT */
/* 774 */          0x5c,          /* FC_PAD */
/* 776 */          0x46,          /* FC_NO_REPEAT */
/* 778 */          0x5c,          /* FC_PAD */
/* 780 */          0x46,          /* FC_NO_REPEAT */
/* 782 */          0x5c,          /* FC_PAD */
/* 784 */          0x12, 0x0,      /* FC_UP */
/* 786 */ NdrFcShort( 0xffffffe8 ), /* Offset= -24 (746) */
/* 788 */          0x5b,          /* FC_END */
/* 790 */          0x8,          /* FC_LONG */
/* 792 */          0x5b,          /* FC_END */
/* 794 */          0x1b,          /* FC_CARRAY */
/* 796 */          0x3,          /* 3 */
/* 798 */ NdrFcShort( 0x4 ), /* 4 */
/* 800 */ 0x19,          /* Corr desc: field pointer, FC_ULONG */
/* 802 */          0x0,          /* */
/* 804 */ NdrFcShort( 0x0 ), /* 0 */
/* 806 */ 0x8,          /* FC_LONG */
/* 808 */          0x5b,          /* FC_END */
/* 810 */          0x16,          /* FC_PSTRUCT */
/* 812 */          0x3,          /* 3 */
/* 814 */ NdrFcShort( 0x8 ), /* 8 */
/* 816 */          0x4b,          /* FC_PP */
/* 818 */          0x5c,          /* FC_PAD */
/* 820 */          0x46,          /* FC_NO_REPEAT */
/* 822 */          0x5c,          /* FC_PAD */
/* 824 */          0x46,          /* FC_NO_REPEAT */
/* 826 */          0x5c,          /* FC_PAD */
/* 828 */          0x46,          /* FC_NO_REPEAT */
/* 830 */          0x5c,          /* FC_PAD */
/* 832 */          0x46,          /* FC_NO_REPEAT */
/* 834 */          0x5c,          /* FC_PAD */
/* 836 */          0x46,          /* FC_NO_REPEAT */
/* 838 */          0x5c,          /* FC_PAD */
/* 840 */          0x46,          /* FC_NO_REPEAT */
/* 842 */          0x5c,          /* FC_PAD */
/* 844 */          0x46,          /* FC_NO_REPEAT */
/* 846 */          0x5c,          /* FC_PAD */
/* 848 */          0x46,          /* FC_NO_REPEAT */
/* 850 */          0x5c,          /* FC_PAD */
/* 852 */          0x46,          /* FC_NO_REPEAT */
/* 854 */          0x5c,          /* FC_PAD */
/* 856 */          0x46,          /* FC_NO_REPEAT */
/* 858 */          0x5c,          /* FC_PAD */
/* 860 */          0x46,          /* FC_NO_REPEAT */
/* 862 */          0x5c,          /* FC_PAD */
/* 864 */          0x46,          /* FC_NO_REPEAT */
/* 866 */          0x5c,          /* FC_PAD */
/* 868 */          0x46,          /* FC_NO_REPEAT */
/* 870 */          0x5c,          /* FC_PAD */
/* 872 */          0x46,          /* FC_NO_REPEAT */
/* 874 */          0x5c,          /* FC_PAD */
/* 876 */          0x46,          /* FC_NO_REPEAT */
/* 878 */          0x5c,          /* FC_PAD */
/* 880 */          0x46,          /* FC_NO_REPEAT */
/* 882 */          0x5c,          /* FC_PAD */
/* 884 */          0x46,          /* FC_NO_REPEAT */
/* 886 */          0x5c,          /* FC_PAD */
/* 888 */          0x46,          /* FC_NO_REPEAT */
/* 890 */          0x5c,          /* FC_PAD */
/* 892 */          0x46,          /* FC_NO_REPEAT */
/* 894 */          0x5c,          /* FC_PAD */
/* 896 */          0x46,          /* FC_NO_REPEAT */
/* 898 */          0x5c,          /* FC_PAD */
/* 900 */          0x46,          /* FC_NO_REPEAT */
/* 902 */          0x5c,          /* FC_PAD */
/* 904 */          0x46,          /* FC_NO_REPEAT */
/* 906 */          0x5c,          /* FC_PAD */
/* 908 */          0x46,          /* FC_NO_REPEAT */
/* 910 */          0x5c,          /* FC_PAD */
/* 912 */          0x46,          /* FC_NO_REPEAT */
/* 914 */          0x5c,          /* FC_PAD */
/* 916 */          0x46,          /* FC_NO_REPEAT */
/* 918 */          0x5c,          /* FC_PAD */
/* 920 */          0x46,          /* FC_NO_REPEAT */
/* 922 */          0x5c,          /* FC_PAD */
/* 924 */          0x46,          /* FC_NO_REPEAT */
/* 926 */          0x5c,          /* FC_PAD */
/* 928 */          0x46,          /* FC_NO_REPEAT */
/* 930 */          0x5c,          /* FC_PAD */
/* 932 */          0x46,          /* FC_NO_REPEAT */
/* 934 */          0x5c,          /* FC_PAD */
/* 936 */          0x46,          /* FC_NO_REPEAT */
/* 938 */          0x5c,          /* FC_PAD */
/* 940 */          0x46,          /* FC_NO_REPEAT */
/* 942 */          0x5c,          /* FC_PAD */
/* 944 */          0x46,          /* FC_NO_REPEAT */
/* 946 */          0x5c,          /* FC_PAD */
/* 948 */          0x46,          /* FC_NO_REPEAT */
/* 950 */          0x5c,          /* FC_PAD */
/* 952 */          0x46,          /* FC_NO_REPEAT */
/* 954 */          0x5c,          /* FC_PAD */
/* 956 */          0x46,          /* FC_NO_REPEAT */
/* 958 */          0x5c,          /* FC_PAD */
/* 960 */          0x46,          /* FC_NO_REPEAT */
/* 962 */          0x5c,          /* FC_PAD */
/* 964 */          0x46,          /* FC_NO_REPEAT */
/* 966 */          0x5c,          /* FC_PAD */
/* 968 */          0x46,          /* FC_NO_REPEAT */
/* 970 */          0x5c,          /* FC_PAD */
/* 972 */          0x46,          /* FC_NO_REPEAT */
/* 974 */          0x5c,          /* FC_PAD */
/* 976 */          0x46,          /* FC_NO_REPEAT */
/* 978 */          0x5c,          /* FC_PAD */
/* 980 */          0x46,          /* FC_NO_REPEAT */
/* 982 */          0x5c,          /* FC_PAD */
/* 984 */          0x46,          /* FC_NO_REPEAT */
/* 986 */          0x5c,          /* FC_PAD */
/* 988 */          0x46,          /* FC_NO_REPEAT */
/* 990 */          0x5c,          /* FC_PAD */
/* 992 */          0x46,          /* FC_NO_REPEAT */
/* 994 */          0x5c,          /* FC_PAD */
/* 996 */          0x46,          /* FC_NO_REPEAT */
/* 998 */          0x5c,          /* FC_PAD */

```

```

                                0x16,          /* FC_PSTRUCT */
/* 818 */ NdrFcShort( 0x8 ), /* 8 */
/* 820 */
                                0x4b,          /* FC_PP */
                                0x5c,          /* FC_PAD */
/* 822 */
                                0x46,          /* FC_NO_REPEAT */
                                0x5c,          /* FC_PAD */
/* 824 */ NdrFcShort( 0x4 ), /* 4 */
/* 826 */ NdrFcShort( 0x4 ), /* 4 */
/* 828 */ 0x12, 0x0, /* FC_UP */
/* 830 */ NdrFcShort( 0xffffffe8 ), /* Offset= -24 (806) */
/* 832 */
                                0x5b,          /* FC_END */
/* 834 */ 0x8, /* FC_LONG */
/* 836 */
                                0x5b,          /* FC_END */
/* 838 */ NdrFcShort( 0x8 ), /* 8 */
/* 840 */ 0x8, /* FC_LONG */
/* 842 */ 0x5c, /* FC_PAD */
/* 844 */
                                0x5b,          /* FC_END */
/* 846 */ NdrFcShort( 0x8 ), /* 8 */
/* 848 */ 0x7, /* Corr desc: FC_USHORT */
/* 850 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 852 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 854 */ NdrFcShort( 0xfffffee ), /* Offset= -18 (836) */
/* 856 */ 0x5c, /* FC_PAD */
/* 858 */
                                0x5b,          /* FC_END */
/* 860 */ NdrFcShort( 0x28 ), /* 40 */
/* 862 */ NdrFcShort( 0xfffffee ), /* Offset= -18 (844) */
/* 864 */ NdrFcShort( 0x0 ), /* Offset= 0 (864) */
/* 866 */ 0x6, /* FC_SHORT */
/* 868 */ 0x38, /* FC_ALIGNM4 */
/* 870 */ 0x8, /* FC_LONG */
/* 872 */ 0x0, /* FC_LONG */
/* 874 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 876 */ NdrFcShort( 0xfffffd7 ), /* Offset= -521 (352) */
/* 878 */ 0x5b, /* FC_END */
/* 880 */
                                0x12, 0x0, /* FC_UP */
/* 882 */ NdrFcShort( 0xfffffe6 ), /* Offset= -266 (612) */
/* 884 */
                                0x12, 0x8, /* FC_UP [simple_pointer] */
/* 886 */ 0x1, /* FC_BYTE */
/* 888 */ 0x5c, /* FC_PAD */
/* 890 */
                                0x12, 0x8, /* FC_UP [simple_pointer] */
/* 892 */ 0x1, /* FC_BYTE */
/* 894 */ 0x5c, /* FC_PAD */
/* 896 */
                                0x12, 0x8, /* FC_UP [simple_pointer] */
/* 898 */ 0xc, /* FC_DOUBLE */
/* 900 */ 0x5c, /* FC_PAD */
/* 902 */ NdrFcShort( 0xfffffd90 ), /* Offset= -624 (278) */
/* 904 */
                                0x12, 0x10, /* FC_UP [pointer_deref] */
/* 906 */ NdrFcShort( 0xfffffd92 ), /* Offset= -622 (284) */
/* 908 */
                                0x12, 0x10, /* FC_UP [pointer_deref] */
/* 910 */ NdrFcShort( 0xfffffda6 ), /* Offset= -602 (308) */
/* 912 */
                                0x12, 0x10, /* FC_UP [pointer_deref] */
/* 914 */ NdrFcShort( 0xfffffdb4 ), /* Offset= -588 (326) */
/* 916 */
                                0x12, 0x10, /* FC_UP [pointer_deref] */
/* 918 */ NdrFcShort( 0xfffffdc2 ), /* Offset= -574 (344) */
/* 920 */
                                0x12, 0x10, /* FC_UP [pointer_deref] */
/* 922 */ NdrFcShort( 0x2 ), /* Offset= 2 (924) */
/* 924 */
                                0x12, 0x0, /* FC_UP */
/* 926 */ NdrFcShort( 0x16 ), /* Offset= 22 (948) */
/* 928 */
                                0x15,          /* FC_PSTRUCT */
                                0x7,          /* 7 */
/* 930 */ NdrFcShort( 0x10 ), /* 16 */
/* 932 */ 0x6, /* FC_SHORT */
/* 934 */ 0x1, /* FC_BYTE */
/* 936 */ 0x8, /* FC_ALIGNM4 */
/* 938 */ 0xb, /* FC_LONG */
/* 940 */ 0x5b, /* FC_ALIGNM8 */
/* 942 */ 0x0, /* FC_HYPER */
/* 944 */ 0x12, 0x0, /* FC_UP */
/* 946 */ NdrFcShort( 0xfffffff2 ), /* Offset= -14 (928) */
/* 948 */ 0x2, /* FC_UP [simple_pointer] */
/* 950 */ 0x5c, /* FC_CHAR */
/* 952 */ 0x1a, /* FC_PAD */
/* 954 */ 0x7, /* FC_BOGUS_STRUCT */
/* 956 */ 0x1, /* 7 */
/* 958 */ NdrFcShort( 0x20 ), /* 32 */
/* 960 */ NdrFcShort( 0x0 ), /* 0 */
/* 962 */ NdrFcShort( 0x0 ), /* Offset= 0 (954) */
/* 964 */ 0x8, /* FC_LONG */
/* 966 */ 0x8, /* FC_LONG */
/* 968 */ 0x6, /* FC_SHORT */

```

```

/* 886 */ 0x6, /* FC_SHORT */
/* 888 */ 0x5c, /* FC_PAD */
/* 890 */ 0x8, /* FC_UP [simple_pointer] */
/* 892 */ 0x5c, /* FC_PAD */
/* 894 */ 0xa, /* FC_UP [simple_pointer] */
/* 896 */ 0x5c, /* FC_PAD */
/* 898 */ 0xc, /* FC_UP [simple_pointer] */
/* 900 */ 0x5c, /* FC_PAD */
/* 902 */ 0x12, 0x0, /* FC_UP */
/* 904 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 906 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 908 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 910 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 912 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 914 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 916 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 918 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 920 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 922 */ 0x2, /* FC_UP [pointer_deref] */
/* 924 */ 0x12, 0x0, /* FC_UP */
/* 926 */ 0x16, /* FC_UP [pointer_deref] */
/* 928 */ 0x15, /* FC_PSTRUCT */
/* 930 */ 0x7, /* FC_PSTRUCT */
/* 932 */ 0x6, /* FC_SHORT */
/* 934 */ 0x1, /* FC_BYTE */
/* 936 */ 0x8, /* FC_ALIGNM4 */
/* 938 */ 0xb, /* FC_LONG */
/* 940 */ 0x5b, /* FC_ALIGNM8 */
/* 942 */ 0x12, 0x0, /* FC_UP */
/* 944 */ 0x12, 0x8, /* FC_UP [simple_pointer] */
/* 946 */ 0x5c, /* FC_CHAR */
/* 948 */ 0x1a, /* FC_PAD */
/* 950 */ 0x7, /* FC_BOGUS_STRUCT */
/* 952 */ 0x1, /* 7 */
/* 954 */ 0x1, /* FC_BOGUS_STRUCT */
/* 956 */ 0x8, /* FC_LONG */
/* 958 */ 0x6, /* FC_SHORT */

```

```

                0x6,                /* FC_SHORT */
/* 960 */ 0x6,                /* FC_SHORT */
                0x6,                /* FC_SHORT */
/* 962 */ 0x4c,                /* FC_EMBEDDED_COMPLEX */
                0x0,                /* 0 */
/* 964 */ NdrFcShort( 0xfffffc42 ), /* Offset= -958 (6) */
/* 966 */ 0x5c,                /* FC_PAD */
                0x5b,                /* FC_END */
/* 968 */ 0xb4,                /* FC_USER_MARSHAL */
                0x83,                /* 131 */
/* 970 */ NdrFcShort( 0x0 ), /* 0 */
/* 972 */ NdrFcShort( 0x10 ), /* 16 */
/* 974 */ NdrFcShort( 0x0 ), /* 0 */
/* 976 */ NdrFcShort( 0xfffffc32 ), /* Offset= -974 (2) */
/* 978 */
                0x11, 0x4,          /* FC_RP [allocated_on_stack] */
/* 980 */ NdrFcShort( 0x6 ), /* Offset= 6 (986) */
/* 982 */
                0x13, 0x0,          /* FC_OP */
/* 984 */ NdrFcShort( 0xfffffcdc ), /* Offset= -36 (948) */
/* 986 */ 0xb4,                /* FC_USER_MARSHAL */
                0x83,                /* 131 */
/* 988 */ NdrFcShort( 0x0 ), /* 0 */
/* 990 */ NdrFcShort( 0x10 ), /* 16 */
/* 992 */ NdrFcShort( 0x0 ), /* 0 */
/* 994 */ NdrFcShort( 0xfffffff4 ), /* Offset= -12 (982) */
                0x0
    }
};

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

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

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

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

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{
    if(! _tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }

    return 0;
}

```

```

}

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

#endif /* !defined(_M_IA64) && !defined(_M_AXP64) */

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

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

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jun 12 18:15:12 2000
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=i2), Wl, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext, robust
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#ifdef _M_IA64 || defined(_M_AXP64)
#define USE_STUBLESS_PROXY

/* verify that the <rpcproxy.h> version is high enough to compile this file*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 475
#endif

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

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 979
#define PROC_FORMAT_STRING_SIZE 253
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 1

typedef struct _MIDL_TYPE_FORMAT_STRING
{

```



```

short      Pad;
unsigned char  Format[ TYPE_FORMAT_STRING_SIZE ];
} MIDL_TYPE_FORMAT_STRING;

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

extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[] =
{
0,
44,
88,
132,
176,
220
};

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

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

```

```

};

CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
&ITPCC_ProxyInfo,
&IID_ITPCC,
IUnknown_QueryInterface_Proxy,
IUnknown_AddRef_Proxy,
IUnknown_Release_Proxy ,
(void *)-1 /* ITPCC::NewOrder */ ,
(void *)-1 /* ITPCC::Payment */ ,
(void *)-1 /* ITPCC::Delivery */ ,
(void *)-1 /* ITPCC::StockLevel */ ,
(void *)-1 /* ITPCC::OrderStatus */ ,
(void *)-1 /* ITPCC::CallSetComplete */
};

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

extern const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ];

static const MIDL_STUB_DESC Object_StubDesc =
{
0,
NdrOleAllocate,
NdrOleFree,
0,
0,
0,
0,
0,
0,
__MIDL_TypeFormatString.Format,
1, /* -error bounds_check flag */
0x50002, /* Ndr library version */
0,
0x5030118, /* MIDL Version 5.3.280 */
0,
UserMarshalRoutines,
0, /* notify & notify_flag routine table */
0x1, /* MIDL flag */
0, /* Reserved3 */
0, /* Reserved4 */
0 /* Reserved5 */
};

#pragma data_seg(".rdata")

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

```

```

        ,VARIANT_UserFree
    }
};

#if !defined(__RPC_WIN64__)
#error Invalid build platform for this stub.
#endif

static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
{
    0,
    {
        /* Procedure NewOrder */

        0x33,          /* FC_AUTO_HANDLE */
        0x6c,          /* Old Flags: object, Oi2 */
/* 2 */ NdrFcLong( 0x0 ), /* 0 */
/* 6 */ NdrFcShort( 0x3 ), /* 3 */
#ifdef ALPHA
/* 8 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
        NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 10 */ NdrFcShort( 0x0 ), /* 0 */
/* 12 */ NdrFcShort( 0x8 ), /* 8 */
/* 14 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
/* 16 */ 0xa, /* 10 */
/* 18 */ 0x7, /* Ext Flags: new corr desc, clt
corr check, srv corr check, */
/* 20 */ NdrFcShort( 0x20 ), /* 32 */
/* 22 */ NdrFcShort( 0x0 ), /* 0 */
/* 24 */ NdrFcShort( 0x0 ), /* 0 */

        /* Parameter txn_in */

/* 26 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef ALPHA
/* 28 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
        NdrFcShort( 0x8 ), /* axp64 Stack size/offset = 8 */
#endif
/* 30 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

        /* Parameter txn_out */

/* 32 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=24 */
#ifdef ALPHA
/* 34 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
        NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 36 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

        /* Return value */

/* 38 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */

```

```

#ifdef ALPHA
/* 40 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
        NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 42 */ 0x8, /* FC_LONG */
/* 0 */

        /* Procedure Payment */

/* 44 */ 0x33, /* FC_AUTO_HANDLE */
/* 46 */ NdrFcLong( 0x0 ), /* 0 */
/* 50 */ NdrFcShort( 0x4 ), /* 4 */
#ifdef ALPHA
/* 52 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
        NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 54 */ NdrFcShort( 0x0 ), /* 0 */
/* 56 */ NdrFcShort( 0x8 ), /* 8 */
/* 58 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
/* 60 */ 0xa, /* 10 */
/* 62 */ 0x7, /* Ext Flags: new corr desc, clt
corr check, srv corr check, */
/* 64 */ NdrFcShort( 0x20 ), /* 32 */
/* 66 */ NdrFcShort( 0x0 ), /* 0 */
/* 68 */ NdrFcShort( 0x0 ), /* 0 */

        /* Parameter txn_in */

/* 70 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef ALPHA
/* 72 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
        NdrFcShort( 0x8 ), /* axp64 Stack size/offset = 8 */
#endif
/* 74 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

        /* Parameter txn_out */

/* 76 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=24 */
#ifdef ALPHA
/* 78 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
        NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 80 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

        /* Return value */

/* 82 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef ALPHA
/* 84 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
        NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 86 */ 0x8, /* FC_LONG */
/* 0 */

```

```

/* Procedure Delivery */

/* 88 */ 0x33, /* FC_AUTO_HANDLE */
/* 90 */ NdrFcLong( 0x0 ), /* 0 */
/* 94 */ NdrFcShort( 0x5 ), /* 5 */
#ifdef ALPHA
/* 96 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
NdrFcShort( 0x30 ), /* xpp64 Stack size/offset = 48 */
#endif
/* 98 */ NdrFcShort( 0x0 ), /* 0 */
/* 100 */ NdrFcShort( 0x8 ), /* 8 */
/* 102 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
/* 104 */ 0xa, /* 10 */
/* 107 */ 0x7, /* Ext Flags: new corr desc, clt
corr check, srv corr check, */
/* 106 */ NdrFcShort( 0x20 ), /* 32 */
/* 108 */ NdrFcShort( 0x20 ), /* 32 */
/* 110 */ NdrFcShort( 0x0 ), /* 0 */
/* 112 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 114 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef ALPHA
/* 116 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
NdrFcShort( 0x8 ), /* xpp64 Stack size/offset = 8 */
#endif
/* 118 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

/* Parameter txn_out */

/* 120 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=24 */
#ifdef ALPHA
/* 122 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
NdrFcShort( 0x20 ), /* xpp64 Stack size/offset = 32 */
#endif
/* 124 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Return value */

/* 126 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef ALPHA
/* 128 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
NdrFcShort( 0x28 ), /* xpp64 Stack size/offset = 40 */
#endif
/* 130 */ 0x8, /* FC_LONG */
/* 131 */ 0x0, /* 0 */

/* Procedure StockLevel */

/* 132 */ 0x33, /* FC_AUTO_HANDLE */
/* 134 */ NdrFcLong( 0x0 ), /* 0 */
/* 138 */ NdrFcShort( 0x6 ), /* 6 */

```

```

#ifdef ALPHA
/* 140 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
NdrFcShort( 0x30 ), /* xpp64 Stack size/offset = 48 */
#endif
/* 142 */ NdrFcShort( 0x0 ), /* 0 */
/* 144 */ NdrFcShort( 0x8 ), /* 8 */
/* 146 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
/* 148 */ 0xa, /* 10 */
/* 151 */ 0x7, /* Ext Flags: new corr desc, clt
corr check, srv corr check, */
/* 150 */ NdrFcShort( 0x20 ), /* 32 */
/* 152 */ NdrFcShort( 0x20 ), /* 32 */
/* 154 */ NdrFcShort( 0x0 ), /* 0 */
/* 156 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 158 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef ALPHA
/* 160 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
NdrFcShort( 0x8 ), /* xpp64 Stack size/offset = 8 */
#endif
/* 162 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

/* Parameter txn_out */

/* 164 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=24 */
#ifdef ALPHA
/* 166 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
NdrFcShort( 0x20 ), /* xpp64 Stack size/offset = 32 */
#endif
/* 168 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Return value */

/* 170 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef ALPHA
/* 172 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
NdrFcShort( 0x28 ), /* xpp64 Stack size/offset = 40 */
#endif
/* 174 */ 0x8, /* FC_LONG */
/* 175 */ 0x0, /* 0 */

/* Procedure OrderStatus */

/* 176 */ 0x33, /* FC_AUTO_HANDLE */
/* 178 */ NdrFcLong( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x7 ), /* 7 */
#ifdef ALPHA
/* 184 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
NdrFcShort( 0x30 ), /* xpp64 Stack size/offset = 48 */
#endif
/* 186 */ NdrFcShort( 0x0 ), /* 0 */
/* 188 */ NdrFcShort( 0x8 ), /* 8 */

```

```

/* 190 */ 0x47,          /* Oi2 Flags:  srv must size, clt must size, has
return, has ext, */
                                0x3,          /* 3 */
/* 192 */ 0xa,          /* 10 */
                                0x7,          /* Ext Flags:  new corr desc, clt
corr check, srv corr check, */
/* 194 */ NdrFcShort( 0x20 ), /* 32 */
/* 196 */ NdrFcShort( 0x20 ), /* 32 */
/* 198 */ NdrFcShort( 0x0 ),  /* 0 */
/* 200 */ NdrFcShort( 0x0 ),  /* 0 */

    /* Parameter txn_in */

/* 202 */ NdrFcShort( 0x8b ), /* Flags:  must size, must free, in, by val, */
#ifdef _ALPHA
/* 204 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
                                NdrFcShort( 0x8 ), /* axp64 Stack size/offset = 8 */
#endif
/* 206 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

    /* Parameter txn_out */

/* 208 */ NdrFcShort( 0x6113 ), /* Flags:  must size, must free, out, simple
ref, srv alloc size=24 */
#ifdef _ALPHA
/* 210 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
                                NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 212 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

    /* Return value */

/* 214 */ NdrFcShort( 0x70 ), /* Flags:  out, return, base type, */
#ifdef _ALPHA
/* 216 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
                                NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 218 */ 0x8,          /* FC_LONG */
                                0x0,          /* 0 */

    /* Procedure CallSetComplete */

/* 220 */ 0x33,          /* FC_AUTO_HANDLE */
                                0x6c,          /* Old Flags:  object, Oi2 */
/* 222 */ NdrFcLong( 0x0 ),  /* 0 */
/* 226 */ NdrFcShort( 0x8 ), /* 8 */
/* 228 */ NdrFcShort( 0x10 ), /* ia64, axp64 Stack size/offset = 16 */
/* 230 */ NdrFcShort( 0x0 ), /* 0 */
/* 232 */ NdrFcShort( 0x8 ), /* 8 */
/* 234 */ 0x44,          /* Oi2 Flags:  has return, has ext, */
                                0x1,          /* 1 */
/* 236 */ 0xa,          /* 10 */
                                0x1,          /* Ext Flags:  new corr desc, */
/* 238 */ NdrFcShort( 0x0 ), /* 0 */
/* 240 */ NdrFcShort( 0x0 ), /* 0 */
/* 242 */ NdrFcShort( 0x0 ), /* 0 */
/* 244 */ NdrFcShort( 0x0 ), /* 0 */

    /* Return value */

```

```

/* 246 */ NdrFcShort( 0x70 ), /* Flags:  out, return, base type, */
/* 248 */ NdrFcShort( 0x8 ), /* ia64, axp64 Stack size/offset = 8 */
/* 250 */ 0x8,          /* FC_LONG */
                                0x0,          /* 0 */

                                0x0

    }
};

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
    0,
    {
                                NdrFcShort( 0x0 ), /* 0 */
/* 2 */
                                0x12, 0x0,          /* FC_UP */
/* 4 */ NdrFcShort( 0x39e ), /* Offset= 926 (930) */
/* 6 */
                                0x2b,          /* FC_NON_ENCAPSULATED_UNION */
                                0x9,          /* FC_ULONG */
/* 8 */ 0x7,          /* Corr desc: FC_USHORT */
                                0x0,          /* */
/* 10 */ NdrFcShort( 0xffff8 ), /* -8 */
/* 12 */ NdrFcShort( 0x1 ), /* Corr flags:  early, */
/* 14 */ NdrFcShort( 0x2 ), /* Offset= 2 (16) */
/* 16 */ NdrFcShort( 0x10 ), /* 16 */
/* 18 */ NdrFcShort( 0x2b ), /* 43 */
/* 20 */ NdrFcLong( 0x3 ), /* 3 */
/* 24 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 26 */ NdrFcLong( 0x11 ), /* 17 */
/* 30 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE */
/* 32 */ NdrFcLong( 0x2 ), /* 2 */
/* 36 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 38 */ NdrFcLong( 0x4 ), /* 4 */
/* 42 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT */
/* 44 */ NdrFcShort( 0x5 ), /* 5 */
/* 48 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 50 */ NdrFcLong( 0xb ), /* 11 */
/* 54 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 56 */ NdrFcLong( 0xa ), /* 10 */
/* 60 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 62 */ NdrFcLong( 0x6 ), /* 6 */
/* 66 */ NdrFcShort( 0xd6 ), /* Offset= 214 (280) */
/* 68 */ NdrFcLong( 0x7 ), /* 7 */
/* 72 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 74 */ NdrFcLong( 0x8 ), /* 8 */
/* 78 */ NdrFcShort( 0xd0 ), /* Offset= 208 (286) */
/* 80 */ NdrFcLong( 0xd ), /* 13 */
/* 84 */ NdrFcShort( 0xe4 ), /* Offset= 228 (312) */
/* 86 */ NdrFcLong( 0x9 ), /* 9 */
/* 90 */ NdrFcShort( 0xf0 ), /* Offset= 240 (330) */
/* 92 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 96 */ NdrFcShort( 0xfc ), /* Offset= 252 (348) */
/* 98 */ NdrFcLong( 0x24 ), /* 36 */
/* 102 */ NdrFcShort( 0x2f4 ), /* Offset= 756 (858) */
/* 104 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 108 */ NdrFcShort( 0x2ee ), /* Offset= 750 (858) */
/* 110 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 114 */ NdrFcShort( 0x2ec ), /* Offset= 748 (862) */
/* 116 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 120 */ NdrFcShort( 0x2ea ), /* Offset= 746 (866) */
/* 122 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 126 */ NdrFcShort( 0x2e8 ), /* Offset= 744 (870) */

```

```

/* 128 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 132 */ NdrFcShort( 0x2e6 ), /* Offset= 742 (874) */
/* 134 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 138 */ NdrFcShort( 0x2e4 ), /* Offset= 740 (878) */
/* 140 */ NdrFcLong( 0x400b ), /* 16395 */
/* 144 */ NdrFcShort( 0x2d2 ), /* Offset= 722 (866) */
/* 146 */ NdrFcLong( 0x400a ), /* 16394 */
/* 150 */ NdrFcShort( 0x2d0 ), /* Offset= 720 (870) */
/* 152 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 156 */ NdrFcShort( 0x2d6 ), /* Offset= 726 (882) */
/* 158 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 162 */ NdrFcShort( 0x2cc ), /* Offset= 716 (878) */
/* 164 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 168 */ NdrFcShort( 0x2ce ), /* Offset= 718 (886) */
/* 170 */ NdrFcLong( 0x400d ), /* 16397 */
/* 174 */ NdrFcShort( 0x2cc ), /* Offset= 716 (890) */
/* 176 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 180 */ NdrFcShort( 0x2ca ), /* Offset= 714 (894) */
/* 182 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 186 */ NdrFcShort( 0x2c8 ), /* Offset= 712 (898) */
/* 188 */ NdrFcLong( 0x400c ), /* 16396 */
/* 192 */ NdrFcShort( 0x2c6 ), /* Offset= 710 (902) */
/* 194 */ NdrFcLong( 0x10 ), /* 16 */
/* 198 */ NdrFcShort( 0x8002 ), /* Simple arm type: FC_CHAR */
/* 200 */ NdrFcLong( 0x12 ), /* 18 */
/* 204 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 206 */ NdrFcLong( 0x13 ), /* 19 */
/* 210 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 212 */ NdrFcLong( 0x16 ), /* 22 */
/* 216 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 218 */ NdrFcLong( 0x17 ), /* 23 */
/* 222 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 224 */ NdrFcLong( 0xe ), /* 14 */
/* 228 */ NdrFcShort( 0x2aa ), /* Offset= 682 (910) */
/* 230 */ NdrFcLong( 0x400e ), /* 16398 */
/* 234 */ NdrFcShort( 0x2b0 ), /* Offset= 688 (922) */
/* 236 */ NdrFcLong( 0x4010 ), /* 16400 */
/* 240 */ NdrFcShort( 0x2ae ), /* Offset= 686 (926) */
/* 242 */ NdrFcLong( 0x4012 ), /* 16402 */
/* 246 */ NdrFcShort( 0x26c ), /* Offset= 620 (866) */
/* 248 */ NdrFcLong( 0x4013 ), /* 16403 */
/* 252 */ NdrFcShort( 0x26a ), /* Offset= 618 (870) */
/* 254 */ NdrFcLong( 0x4016 ), /* 16406 */
/* 258 */ NdrFcShort( 0x264 ), /* Offset= 612 (870) */
/* 260 */ NdrFcLong( 0x4017 ), /* 16407 */
/* 264 */ NdrFcShort( 0x25e ), /* Offset= 606 (870) */
/* 266 */ NdrFcLong( 0x0 ), /* 0 */
/* 270 */ NdrFcShort( 0x0 ), /* Offset= 0 (270) */
/* 272 */ NdrFcLong( 0x1 ), /* 1 */
/* 276 */ NdrFcShort( 0x0 ), /* Offset= 0 (276) */
/* 278 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (277) */
/* 280 */
/* 282 */ NdrFcShort( 0x8 ), /* 8 */
/* 284 */ 0xb, /* FC_HYPER */
/* 286 */
/* 288 */ NdrFcShort( 0xe ), /* Offset= 14 (302) */
/* 290 */
/* 292 */ 0x1b, /* FC_CARRAY */
/* 294 */ 0x1, /* 1 */
/* 296 */ 0x15, /* FC_STRUCT */
/* 298 */ 0x7, /* 7 */
/* 300 */ 0x12, 0x0, /* FC_UP */
/* 302 */ /* Offset= 14 (302) */
/* 304 */ 0x1b, /* FC_CARRAY */
/* 306 */ 0x1, /* 1 */
/* 308 */ 0x292, /* FC_END */
/* 310 */
/* 312 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 314 */ /* Offset= 2 (352) */
/* 316 */
/* 318 */ 0x12, 0x0, /* FC_UP */
/* 320 */ /* Offset= 486 (840) */
/* 322 */
/* 324 */ 0x2a, /* FC_ENCAPSULATED_UNION */
/* 326 */ 0x89, /* 137 */
/* 328 */ 0x20, /* 32 */
/* 330 */ 0xa, /* 10 */
/* 332 */ 0x8, /* 8 */
/* 334 */ 0x50, /* Offset= 80 (446) */
/* 336 */ 0xd, /* 13 */
/* 338 */ 0x70, /* Offset= 112 (484) */
/* 340 */ 0x9, /* 9 */
/* 342 */ 0x90, /* Offset= 144 (522) */
/* 344 */ 0xc, /* 12 */
/* 346 */ 0xb0, /* Offset= 176 (560) */

```

```

/* 292 */ NdrFcShort( 0x2 ), /* 2 */
/* 294 */ 0x9, /* Corr desc: FC_ULONG */
/* 296 */ 0x0, /* -4 */
/* 298 */ NdrFcShort( 0xfffc ), /* -4 */
/* 300 */ 0x6, /* Corr flags: early, */
/* 302 */ 0x5b, /* FC_SHORT */
/* 304 */ 0x17, /* FC_END */
/* 306 */ 0x3, /* FC_CSTRUCT */
/* 308 */ 0x8, /* 8 */
/* 310 */ 0x5c, /* FC_LONG */
/* 312 */ 0x5b, /* FC_PAD */
/* 314 */ 0x2f, /* FC_IP */
/* 316 */ 0x5a, /* FC_CONSTANT_IID */
/* 318 */ 0x0, /* 0 */
/* 320 */ 0x0, /* 0 */
/* 322 */ 0xc0, /* 192 */
/* 324 */ 0x0, /* 0 */
/* 326 */ 0x0, /* 0 */
/* 328 */ 0x0, /* 0 */
/* 330 */ 0x46, /* 70 */
/* 332 */ 0x2f, /* FC_IP */
/* 334 */ 0x5a, /* FC_CONSTANT_IID */
/* 336 */ 0x20400, /* 132096 */
/* 338 */ 0x0, /* 0 */
/* 340 */ 0xc0, /* 192 */
/* 342 */ 0x0, /* 0 */
/* 344 */ 0x0, /* 0 */
/* 346 */ 0x0, /* 0 */
/* 348 */ 0x46, /* 70 */
/* 350 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 352 */ /* Offset= 2 (352) */
/* 354 */ 0x12, 0x0, /* FC_UP */
/* 356 */ /* Offset= 486 (840) */
/* 358 */ 0x2a, /* FC_ENCAPSULATED_UNION */
/* 360 */ 0x89, /* 137 */
/* 362 */ 0x20, /* 32 */
/* 364 */ 0xa, /* 10 */
/* 366 */ 0x8, /* 8 */
/* 368 */ 0x50, /* Offset= 80 (446) */
/* 370 */ 0xd, /* 13 */
/* 372 */ 0x70, /* Offset= 112 (484) */
/* 374 */ 0x9, /* 9 */
/* 376 */ 0x90, /* Offset= 144 (522) */
/* 378 */ 0xc, /* 12 */
/* 380 */ 0xb0, /* Offset= 176 (560) */

```

```

/* 386 */ NdrFcLong( 0x24 ), /* 36 */
/* 390 */ NdrFcShort( 0x104 ), /* Offset= 260 (650) */
/* 392 */ NdrFcLong( 0x800d ), /* 32781 */
/* 396 */ NdrFcShort( 0x120 ), /* Offset= 288 (684) */
/* 398 */ NdrFcLong( 0x10 ), /* 16 */
/* 402 */ NdrFcShort( 0x13a ), /* Offset= 314 (716) */
/* 404 */ NdrFcLong( 0x2 ), /* 2 */
/* 408 */ NdrFcShort( 0x150 ), /* Offset= 336 (744) */
/* 410 */ NdrFcLong( 0x3 ), /* 3 */
/* 414 */ NdrFcShort( 0x166 ), /* Offset= 358 (772) */
/* 416 */ NdrFcLong( 0x14 ), /* 20 */
/* 420 */ NdrFcShort( 0x17c ), /* Offset= 380 (800) */
/* 422 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (421) */
/* 424 */

                                0x21, /* FC_BOGUS_ARRAY */
                                0x3, /* 3 */
/* 426 */ NdrFcShort( 0x0 ), /* 0 */
/* 428 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                                0x0, /* */
/* 430 */ NdrFcShort( 0x0 ), /* 0 */
/* 432 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 434 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 438 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 440 */

                                0x12, 0x0, /* FC_UP */
/* 442 */ NdrFcShort( 0xffffffff74 ), /* Offset= -140 (302) */
/* 444 */ 0x5c, /* FC_PAD */
/* 446 */ 0x5b, /* FC_END */

                                0x1a, /* FC_BOGUS_STRUCT */
                                0x3, /* 3 */
/* 448 */ NdrFcShort( 0x10 ), /* 16 */
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */ NdrFcShort( 0x6 ), /* Offset= 6 (458) */
/* 454 */ 0x8, /* FC_LONG */
/* 456 */ 0x36, /* FC_POINTER */
                                0x39, /* FC_ALIGNM8 */
                                0x5b, /* FC_END */
/* 458 */

                                0x11, 0x0, /* FC_RP */
/* 460 */ NdrFcShort( 0xffffffffdc ), /* Offset= -36 (424) */
/* 462 */

                                0x21, /* FC_BOGUS_ARRAY */
                                0x3, /* 3 */
/* 464 */ NdrFcShort( 0x0 ), /* 0 */
/* 466 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                                0x0, /* */
/* 468 */ NdrFcShort( 0x0 ), /* 0 */
/* 470 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 472 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 476 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 478 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
                                0x0, /* 0 */
/* 480 */ NdrFcShort( 0xffffffff58 ), /* Offset= -168 (312) */
/* 482 */ 0x5c, /* FC_PAD */
/* 484 */ 0x5b, /* FC_END */

                                0x1a, /* FC_BOGUS_STRUCT */
                                0x3, /* 3 */
/* 486 */ NdrFcShort( 0x10 ), /* 16 */
/* 488 */ NdrFcShort( 0x0 ), /* 0 */
/* 490 */ NdrFcShort( 0x6 ), /* Offset= 6 (496) */
/* 492 */ 0x8, /* FC_LONG */

```

```

                                0x39, /* FC_ALIGNM8 */
/* 494 */ 0x36, /* FC_POINTER */
                                0x5b, /* FC_END */
/* 496 */

                                0x11, 0x0, /* FC_RP */
/* 498 */ NdrFcShort( 0xffffffffdc ), /* Offset= -36 (462) */
/* 500 */

                                0x21, /* FC_BOGUS_ARRAY */
                                0x3, /* 3 */
/* 502 */ NdrFcShort( 0x0 ), /* 0 */
/* 504 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                                0x0, /* */
/* 506 */ NdrFcShort( 0x0 ), /* 0 */
/* 508 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 510 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 514 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 516 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
                                0x0, /* 0 */
/* 518 */ NdrFcShort( 0xffffffff44 ), /* Offset= -188 (330) */
/* 520 */ 0x5c, /* FC_PAD */
/* 522 */ 0x5b, /* FC_END */

                                0x1a, /* FC_BOGUS_STRUCT */
                                0x3, /* 3 */
/* 524 */ NdrFcShort( 0x10 ), /* 16 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 530 */ 0x8, /* FC_LONG */
/* 532 */ 0x36, /* FC_POINTER */
                                0x39, /* FC_ALIGNM8 */
                                0x5b, /* FC_END */
/* 534 */

                                0x11, 0x0, /* FC_RP */
/* 536 */ NdrFcShort( 0xffffffffdc ), /* Offset= -36 (500) */
/* 538 */

                                0x21, /* FC_BOGUS_ARRAY */
                                0x3, /* 3 */
/* 540 */ NdrFcShort( 0x0 ), /* 0 */
/* 542 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                                0x0, /* */
/* 544 */ NdrFcShort( 0x0 ), /* 0 */
/* 546 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 548 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 552 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 554 */

                                0x12, 0x0, /* FC_UP */
/* 556 */ NdrFcShort( 0x176 ), /* Offset= 374 (930) */
/* 558 */ 0x5c, /* FC_PAD */
/* 560 */ 0x5b, /* FC_END */

                                0x1a, /* FC_BOGUS_STRUCT */
                                0x3, /* 3 */
/* 562 */ NdrFcShort( 0x10 ), /* 16 */
/* 564 */ NdrFcShort( 0x0 ), /* 0 */
/* 566 */ NdrFcShort( 0x6 ), /* Offset= 6 (572) */
/* 568 */ 0x8, /* FC_LONG */
/* 570 */ 0x36, /* FC_POINTER */
                                0x39, /* FC_ALIGNM8 */
                                0x5b, /* FC_END */
/* 572 */

                                0x11, 0x0, /* FC_RP */
/* 574 */ NdrFcShort( 0xffffffffdc ), /* Offset= -36 (538) */
/* 576 */

```

```

0x2f,          /* FC_IP */
0x5a,          /* FC_CONSTANT_IID */
/* 578 */ NdrFcLong( 0x2f ), /* 47 */
/* 582 */ NdrFcShort( 0x0 ), /* 0 */
/* 584 */ NdrFcShort( 0x0 ), /* 0 */
/* 586 */ 0xc0, /* 192 */
/* 588 */ 0x0, /* 0 */
/* 590 */ 0x0, /* 0 */
/* 592 */ 0x0, /* 0 */
/* 594 */ 0x46, /* 70 */
/* 596 */ 0x1b, /* FC_CARRAY */
0x0,          /* 0 */
/* 598 */ NdrFcShort( 0x1 ), /* 1 */
/* 598 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0,          /* */
/* 600 */ NdrFcShort( 0x4 ), /* 4 */
/* 602 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 604 */ 0x1, /* FC_BYTE */
0x5b,          /* FC_END */
/* 606 */ 0x1a, /* FC_BOGUS_STRUCT */
0x3,          /* 3 */
/* 608 */ NdrFcShort( 0x18 ), /* 24 */
/* 610 */ NdrFcShort( 0x0 ), /* 0 */
/* 612 */ NdrFcShort( 0xc ), /* Offset= 12 (624) */
/* 614 */ 0x8, /* FC_LONG */
0x8,          /* FC_LONG */
/* 616 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0,          /* 0 */
/* 618 */ NdrFcShort( 0xfffffd6 ), /* Offset= -42 (576) */
/* 620 */ 0x39, /* FC_ALIGNM8 */
0x36,          /* FC_POINTER */
/* 622 */ 0x5c, /* FC_PAD */
0x5b,          /* FC_END */
/* 624 */ 0x12, 0x0, /* FC_UP */
/* 626 */ NdrFcShort( 0xfffffe0 ), /* Offset= -32 (594) */
/* 628 */ 0x21, /* FC_BOGUS_ARRAY */
0x3,          /* 3 */
/* 630 */ NdrFcShort( 0x0 ), /* 0 */
/* 632 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0,          /* */
/* 634 */ NdrFcShort( 0x0 ), /* 0 */
/* 636 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 638 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 642 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 644 */ 0x12, 0x0, /* FC_UP */
/* 646 */ NdrFcShort( 0xfffffd8 ), /* Offset= -40 (606) */
/* 648 */ 0x5c, /* FC_PAD */
0x5b,          /* FC_END */
/* 650 */ 0x1a, /* FC_BOGUS_STRUCT */
0x3,          /* 3 */
/* 652 */ NdrFcShort( 0x10 ), /* 16 */
/* 654 */ NdrFcShort( 0x0 ), /* 0 */
/* 656 */ NdrFcShort( 0x6 ), /* Offset= 6 (662) */
/* 658 */ 0x8, /* FC_LONG */

```

```

0x39,          /* FC_ALIGNM8 */
/* 660 */ 0x36, /* FC_POINTER */
0x5b,          /* FC_END */
/* 662 */ 0x11, 0x0, /* FC_RP */
/* 664 */ NdrFcShort( 0xfffffddc ), /* Offset= -36 (628) */
/* 666 */ 0x1d, /* FC_SMFARRAY */
0x0,          /* 0 */
/* 668 */ NdrFcShort( 0x8 ), /* 8 */
/* 670 */ 0x2, /* FC_CHAR */
0x5b,          /* FC_END */
/* 672 */ 0x15, /* FC_STRUCT */
0x3,          /* 3 */
/* 674 */ NdrFcShort( 0x10 ), /* 16 */
/* 676 */ 0x8, /* FC_LONG */
0x6,          /* FC_SHORT */
/* 678 */ 0x6, /* FC_SHORT */
0x4c,          /* FC_EMBEDDED_COMPLEX */
/* 680 */ 0x0, /* 0 */
NdrFcShort( 0xfffffff1 ), /* Offset= -15 (666) */
0x5b,          /* FC_END */
/* 684 */ 0x1a, /* FC_BOGUS_STRUCT */
0x3,          /* 3 */
/* 686 */ NdrFcShort( 0x20 ), /* 32 */
/* 688 */ NdrFcShort( 0x0 ), /* 0 */
/* 690 */ NdrFcShort( 0xa ), /* Offset= 10 (700) */
/* 692 */ 0x8, /* FC_LONG */
0x39,          /* FC_ALIGNM8 */
/* 694 */ 0x36, /* FC_POINTER */
0x4c,          /* FC_EMBEDDED_COMPLEX */
/* 696 */ 0x0, /* 0 */
NdrFcShort( 0xffffffe7 ), /* Offset= -25 (672) */
0x5b,          /* FC_END */
/* 700 */ 0x11, 0x0, /* FC_RP */
/* 702 */ NdrFcShort( 0xffffff10 ), /* Offset= -240 (462) */
/* 704 */ 0x1b, /* FC_CARRAY */
0x0,          /* 0 */
/* 706 */ NdrFcShort( 0x1 ), /* 1 */
/* 708 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0,          /* */
/* 710 */ NdrFcShort( 0x0 ), /* 0 */
/* 712 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 714 */ 0x1, /* FC_BYTE */
0x5b,          /* FC_END */
/* 716 */ 0x1a, /* FC_BOGUS_STRUCT */
0x3,          /* 3 */
/* 718 */ NdrFcShort( 0x10 ), /* 16 */
/* 720 */ NdrFcShort( 0x0 ), /* 0 */
/* 722 */ NdrFcShort( 0x6 ), /* Offset= 6 (728) */
/* 724 */ 0x8, /* FC_LONG */
0x39,          /* FC_ALIGNM8 */
/* 726 */ 0x36, /* FC_POINTER */
0x5b,          /* FC_END */
/* 728 */ 0x12, 0x0, /* FC_UP */
/* 730 */ NdrFcShort( 0xffffffe6 ), /* Offset= -26 (704) */
/* 732 */

```

```

0x1b,          /* FC_CARRAY */
0x1,          /* 1 */
/* 734 */ NdrFcShort( 0x2 ), /* 2 */
/* 736 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0,          /* */
/* 738 */ NdrFcShort( 0x0 ), /* 0 */
/* 740 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 742 */ 0x6, /* FC_SHORT */
0x5b,        /* FC_END */
/* 744 */
0x1a,        /* FC_BOGUS_STRUCT */
0x3,        /* 3 */
/* 746 */ NdrFcShort( 0x10 ), /* 16 */
/* 748 */ NdrFcShort( 0x0 ), /* 0 */
/* 750 */ NdrFcShort( 0x6 ), /* Offset= 6 (756) */
/* 752 */ 0x8, /* FC_LONG */
0x39,        /* FC_ALIGNM8 */
/* 754 */ 0x36, /* FC_POINTER */
0x5b,        /* FC_END */
/* 756 */
0x12, 0x0,   /* FC_UP */
/* 758 */ NdrFcShort( 0xffffffe6 ), /* Offset= -26 (732) */
/* 760 */
0x1b,        /* FC_CARRAY */
0x3,        /* 3 */
/* 762 */ NdrFcShort( 0x4 ), /* 4 */
/* 764 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0,          /* */
/* 766 */ NdrFcShort( 0x0 ), /* 0 */
/* 768 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 770 */ 0x8, /* FC_LONG */
0x5b,        /* FC_END */
/* 772 */
0x1a,        /* FC_BOGUS_STRUCT */
0x3,        /* 3 */
/* 774 */ NdrFcShort( 0x10 ), /* 16 */
/* 776 */ NdrFcShort( 0x0 ), /* 0 */
/* 778 */ NdrFcShort( 0x6 ), /* Offset= 6 (784) */
/* 780 */ 0x8, /* FC_LONG */
0x39,        /* FC_ALIGNM8 */
/* 782 */ 0x36, /* FC_POINTER */
0x5b,        /* FC_END */
/* 784 */
0x12, 0x0,   /* FC_UP */
/* 786 */ NdrFcShort( 0xffffffe6 ), /* Offset= -26 (760) */
/* 788 */
0x1b,        /* FC_CARRAY */
0x7,        /* 7 */
/* 790 */ NdrFcShort( 0x8 ), /* 8 */
/* 792 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0,          /* */
/* 794 */ NdrFcShort( 0x0 ), /* 0 */
/* 796 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 798 */ 0xb, /* FC_HYPER */
0x5b,        /* FC_END */
/* 800 */
0x1a,        /* FC_BOGUS_STRUCT */
0x3,        /* 3 */
/* 802 */ NdrFcShort( 0x10 ), /* 16 */
/* 804 */ NdrFcShort( 0x0 ), /* 0 */
/* 806 */ NdrFcShort( 0x6 ), /* Offset= 6 (812) */
/* 808 */ 0x8, /* FC_LONG */
0x39,        /* FC_ALIGNM8 */

```

```

/* 810 */ 0x36, /* FC_POINTER */
0x5b,        /* FC_END */
/* 812 */
0x12, 0x0,   /* FC_UP */
/* 814 */ NdrFcShort( 0xffffffe6 ), /* Offset= -26 (788) */
/* 816 */
0x15,        /* FC_STRUCT */
0x3,        /* 3 */
/* 818 */ NdrFcShort( 0x8 ), /* 8 */
/* 820 */ 0x8, /* FC_LONG */
0x8,        /* FC_LONG */
/* 822 */ 0x5c, /* FC_PAD */
0x5b,        /* FC_END */
/* 824 */
0x1b,        /* FC_CARRAY */
0x3,        /* 3 */
/* 826 */ NdrFcShort( 0x8 ), /* 8 */
/* 828 */ 0x7, /* Corr desc: FC_USHORT */
0x0,          /* */
/* 830 */ NdrFcShort( 0xffc8 ), /* -56 */
/* 832 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 834 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0,          /* 0 */
/* 836 */ NdrFcShort( 0xfffffec ), /* Offset= -20 (816) */
/* 838 */ 0x5c, /* FC_PAD */
0x5b,        /* FC_END */
/* 840 */
0x1a,        /* FC_BOGUS_STRUCT */
0x3,        /* 3 */
/* 842 */ NdrFcShort( 0x38 ), /* 56 */
/* 844 */ NdrFcShort( 0xfffffec ), /* Offset= -20 (824) */
/* 846 */ NdrFcShort( 0x0 ), /* Offset= 0 (846) */
/* 848 */ 0x6, /* FC_SHORT */
0x6,        /* FC_SHORT */
/* 850 */ 0x38, /* FC_ALIGNM4 */
0x8,        /* FC_LONG */
/* 852 */ 0x8, /* FC_LONG */
0x4c,        /* FC_EMBEDDED_COMPLEX */
/* 854 */ 0x4, /* 4 */
NdrFcShort( 0xfffffe0d ), /* Offset= -499 (356) */
0x5b,        /* FC_END */
/* 858 */
0x12, 0x0,   /* FC_UP */
/* 860 */ NdrFcShort( 0xffffff02 ), /* Offset= -254 (606) */
/* 862 */
0x12, 0x8,   /* FC_UP [simple_pointer] */
/* 864 */ 0x1, /* FC_BYTE */
0x5c,        /* FC_PAD */
/* 866 */
0x12, 0x8,   /* FC_UP [simple_pointer] */
/* 868 */ 0x6, /* FC_SHORT */
0x5c,        /* FC_PAD */
/* 870 */
0x12, 0x8,   /* FC_UP [simple_pointer] */
/* 872 */ 0x8, /* FC_LONG */
0x5c,        /* FC_PAD */
/* 874 */
0x12, 0x8,   /* FC_UP [simple_pointer] */
/* 876 */ 0xa, /* FC_FLOAT */
0x5c,        /* FC_PAD */
/* 878 */
0x12, 0x8,   /* FC_UP [simple_pointer] */
/* 880 */ 0xc, /* FC_DOUBLE */

```



```

                                0x5c,          /* FC_PAD */
/* 882 */
                                0x12, 0x0,      /* FC_UP */
/* 884 */ NdrFcShort( 0xffffda4 ), /* Offset= -604(280) */
/* 886 */
                                0x12, 0x10,     /* FC_UP [pointer_deref] */
/* 888 */ NdrFcShort( 0xffffda6 ), /* Offset= -602(286) */
/* 890 */
                                0x12, 0x10,     /* FC_UP [pointer_deref] */
/* 892 */ NdrFcShort( 0xffffdbc ), /* Offset= -580(312) */
/* 894 */
                                0x12, 0x10,     /* FC_UP [pointer_deref] */
/* 896 */ NdrFcShort( 0xffffdca ), /* Offset= -566(330) */
/* 898 */
                                0x12, 0x10,     /* FC_UP [pointer_deref] */
/* 900 */ NdrFcShort( 0xffffdd8 ), /* Offset= -552(348) */
/* 902 */
                                0x12, 0x10,     /* FC_UP [pointer_deref] */
/* 904 */ NdrFcShort( 0x2 ), /* Offset= 2 (906) */
/* 906 */
                                0x12, 0x0,      /* FC_UP */
/* 908 */ NdrFcShort( 0x16 ), /* Offset= 22 (930) */
/* 910 */
                                0x15,          /* FC_STRUCT */
                                0x7,          /* 7 */
/* 912 */ NdrFcShort( 0x10 ), /* 16 */
/* 914 */ 0x6,                /* FC_SHORT */
                                0x1,          /* FC_BYTE */
/* 916 */ 0x1,                /* FC_ALIGNM4 */
                                0x3B,         /* FC_LONG */
/* 918 */ 0x8,                /* FC_ALIGNM8 */
                                0x39,         /* FC_HYPER */
/* 920 */ 0xb,                /* FC_END */
                                0x5b,         /* FC_UP */
/* 922 */
                                0x12, 0x0,     /* FC_UP */
/* 924 */ NdrFcShort( 0xffffff2 ), /* Offset= -14(910) */
/* 926 */
                                0x12, 0x8,     /* FC_UP [simple_pointer] */
/* 928 */ 0x2,                /* FC_CHAR */
                                0x5c,         /* FC_PAD */
/* 930 */
                                0x1a,          /* FC_BOGUS_STRUCT */
                                0x7,          /* 7 */
/* 932 */ NdrFcShort( 0x20 ), /* 32 */
/* 934 */ NdrFcShort( 0x0 ), /* 0 */
/* 936 */ NdrFcShort( 0x0 ), /* Offset= 0 (936) */
/* 938 */ 0x8,                /* FC_LONG */
                                0x8,          /* FC_LONG */
/* 940 */ 0x6,                /* FC_SHORT */
                                0x6,          /* FC_SHORT */
/* 942 */ 0x6,                /* FC_SHORT */
                                0x6,          /* FC_SHORT */
/* 944 */ 0x4c,               /* FC_EMBEDDED_COMPLEX */
                                0x0,          /* 0 */
/* 946 */ NdrFcShort( 0xfffffc54 ), /* Offset= -940(6) */
/* 948 */ 0x5c,               /* FC_PAD */
                                0x5b,         /* FC_END */
/* 950 */ 0xb4,               /* FC_USER_MARSHAL */
                                0x83,         /* 131 */
/* 952 */ NdrFcShort( 0x0 ), /* 0 */
/* 954 */ NdrFcShort( 0x18 ), /* 24 */
/* 956 */ NdrFcShort( 0x0 ), /* 0 */

```

```

/* 958 */ NdrFcShort( 0xfffffc44 ), /* Offset= -956(2) */
/* 960 */
                                0x11, 0x4,     /* FC_RP [allocated_on_stack] */
/* 962 */ NdrFcShort( 0x6 ), /* Offset= 6 (968) */
/* 964 */
                                0x13, 0x0,     /* FC_OP */
/* 966 */ NdrFcShort( 0xfffffcdc ), /* Offset= -36(930) */
/* 968 */ 0xb4,               /* FC_USER_MARSHAL */
                                0x83,         /* 131 */
/* 970 */ NdrFcShort( 0x0 ), /* 0 */
/* 972 */ NdrFcShort( 0x18 ), /* 24 */
/* 974 */ NdrFcShort( 0x0 ), /* 0 */
/* 976 */ NdrFcShort( 0xfffffff4 ), /* Offset= -12(964) */
                                0x0
    }
};

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

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

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

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

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{
    if(!_tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }

    return 0;
}

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

```

```

    0, /* Filler2 */
    0 /* Filler3 */
};

#endif /* defined(_M_IA64) || defined(_M_AXP64)*/

```

tpcc_com_sl.rgs

```

HKCR
{
    TPCC.StockLevel.1 = s 'StockLevel Class'
    {
        CLSID = s '{2668369E-A50D-11D2-BA4E-00C04FBFE08B}'
    }
    TPCC.StockLevel = s 'StockLevel Class'
    {
        CurVer = s 'TPCC.StockLevel.1'
    }
    NoRemove CLSID
    {
        ForceRemove {2668369E-A50D-11D2-BA4E-00C04FBFE08B} = s
'StockLevel Class'
        {
            ProgID = s 'TPCC.StockLevel.1'
            VersionIndependentProgID = s 'TPCC.StockLevel'
            InprocServer32 = s '%MODULE%'
            {
                val ThreadingModel = s 'Both'
            }
        }
    }
}

```

tpcc_dblib.cpp

```

/* FILE: TPCC_DBLIB.CPP
 * Microsoft TPC-C Kit Ver. 4.20.000
 * Copyright Microsoft, 1999
 * All Rights Reserved
 *
 * Version 4.10.000 audited by Richard Gimarc,
Performance Metrics, 3/17/99
 *
 * PURPOSE: Implements dblib calls for TPC-C txns.
 * Contact: Charles Levine (clevine@microsoft.com)
 *
 * Change history:
 * 4.20.000 - updated rev number to match kit
 * 4.10.001 - not deleting error class in catch handler on deadlock
retry;
 * not a functional bug, but a memory leak
 * - had to tweak some declarations to compile
with latest SDK; no functional change
*/

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

```

```

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

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

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

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

#define DEFCLPACKSIZE 4096

// version string; must match return value from tpcc_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 WINAPIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:
            DisableThreadLibraryCalls(hModule);
            dbinit(); // initialize dblib
            break;

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

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

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

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

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

```

```

}

/* FUNCTION: int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int
severity, char *msgtext)
*
* PURPOSE:      This function handles DB-Library SQL Server error messages
*
* ARGUMENTS:    DBPROCESS      *dbproc      DBPROCESS id
pointer
*              message number      DBINT      msgno
*
*              int
*              message state
*              int
*              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) (DBPROCESS, DBINT, INT, INT, LPCSTR,
LPCSTR, LPCSTR, DBUSMALLINT);

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

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

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

    return 0;
}

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

```

```

* COMMENTS:     Unlike strncpy this function ensures that the result string is
*              always null terminated.
*
*/

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

    return;
}

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
*
*/

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

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

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

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

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

```

```

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

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

    m_MaxRetries = 10;      // how many retries on deadlock

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

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

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

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

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

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

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

    m_dbproc = dbopen(login, szServer);

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

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

    // save address of class instance so that the message and error handler

```

```

// can get to data.
dbsetuserdata(m_dbproc, (LPVOID)this);

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

// set connection properties to match those used by ODBC
dbcmd(m_dbproc, "set ANSI_DEFAULTS ON ");
dbcmd(m_dbproc, "set CURSOR_CLOSE_ON_COMMIT OFF ");
dbcmd(m_dbproc, "set IMPLICIT_TRANSACTIONS OFF ");
dbcmd(m_dbproc, "set NOCOUNT ON "); // do not
return row counts
on abort dbcmd(m_dbproc, "set XACT_ABORT ON "); // rollback transaction

// for coyote
dbcmd(m_dbproc, "set ansi_warnings on "); //
dbcmd(m_dbproc, "set ansi_nulls on "); //

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

// This value must match the number of commands above.
DiscardNextResults(2);
DiscardNextResults(5); // coyote

// 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
    DLib error.
    if (m_SqlErr != NULL)
    {
        CSQLERR *pSqlErr;
        pSqlErr = m_SqlErr;
        m_SqlErr = NULL; // clear our pointer to instance; catch
        handler will delete
        throw pSqlErr;
    }

    CDBLIBERR *pDbLibErr;
    if (m_DbLibErr == NULL)
        // this case isn't expected to happen, since it means that an
        error was returned
        // but the error handlers were not called.

```

```

        pDbLibErr = new CDBLIBERR(eAction);
    else
    {
        pDbLibErr = m_DbLibErr;
        pDbLibErr->m_eAction = eAction;
        m_DbLibErr = NULL; // clear our pointer to instance;
        catch handler will delete
    }

    throw pDbLibErr;
}

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

    while (TRUE)
    {
        rc = dbnextrow(m_dbproc);
        if (rc == NO_MORE_ROWS)
            break;
        if (rc == FAIL)
        {
            if (iExpectedCount >= 0)
                ThrowError(CDBLIBERR::eDbNextRow);
            else
                break;
        }
        iRowsRead++;
    }

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

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

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

```

```

        if (iExpectedCount >= 0)
            ThrowError(CDBLIBERR::eDbResults);
        else
            break;
    }

    DiscardNextRows(-1);
    iResultsRead++;
}

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

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

    ResetError();

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

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

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

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

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

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

            DiscardNextRows(0);
            DiscardNextResults(0);

            m_txn.StockLevel.exec_status_code = eOK;
            return;
        }
        catch (CSQLERR *e)
        {
            if ((e->m_msgno == 1205 ||
                (e->m_msgno == iErrOleDbProvider &&
                 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;
    DBDATERECD         daterec;

    int                iTryCount = 0;
    const BYTE        *pData;

    ResetError();

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

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

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

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

```

```

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

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

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

                if (dbnumcols(m_dbproc) != 5)

                    ThrowError(CDBLIBERR::eWrongNumCols);

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

                if (pData=dbdata(m_dbproc, 1))

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

                if (pData=dbdata(m_dbproc, 3))

                    UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_generic, pData,
                    dbdatlen(m_dbproc, 3));
                if (pData=dbdata(m_dbproc, 4))
                    dbconvert(m_dbproc, SQLNUMERIC,
                    (LPCBYTE)pData, dbdatlen(m_dbproc, 4),
                    SQLFLT8, (BYTE
                    *)&m_txn.NewOrder.OL[i].ol_i_price, 8);
                if (pData=dbdata(m_dbproc, 5))
                    dbconvert(m_dbproc, SQLNUMERIC,
                    (LPCBYTE)pData, dbdatlen(m_dbproc, 5),
                    SQLFLT8, (BYTE
                    *)&m_txn.NewOrder.OL[i].ol_amount, 8);

                m_txn.NewOrder.total_amount =
                m_txn.NewOrder.total_amount + m_txn.NewOrder.OL[i].ol_amount;

                DiscardNextRows(0);
            }

            // get remaining values for w_tax, d_tax, o_id,
            c_last, c_discount, c_credit, o_entry_d, commit_flag
            if (dbresults(m_dbproc) != SUCCEED)
                ThrowError(CDBLIBERR::eDbResults);

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

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

```

```

            if (pData=dbdata(m_dbproc, 1))

                dbconvert(m_dbproc, SQLNUMERIC,
                (LPCBYTE)pData, dbdatlen(m_dbproc, 1), SQLFLT8, (BYTE *)&m_txn.NewOrder.w_tax, 8);
            if (pData=dbdata(m_dbproc, 2))

                dbconvert(m_dbproc, SQLNUMERIC,
                (LPCBYTE)pData, dbdatlen(m_dbproc, 2), SQLFLT8, (BYTE *)&m_txn.NewOrder.d_tax, 8);
            if (pData=dbdata(m_dbproc, 3))
                m_txn.NewOrder.o_id = (*(DBINT *) pData);
            if (pData=dbdata(m_dbproc, 4))
                UtilStrCpy(m_txn.NewOrder.c_last, pData,
                dbdatlen(m_dbproc, 4));
            if (pData=dbdata(m_dbproc, 5))
                dbconvert(m_dbproc, SQLNUMERIC,
                (LPCBYTE)pData, dbdatlen(m_dbproc, 5), SQLFLT8, (BYTE *)&m_txn.NewOrder.c_discount,
                8);
            if (pData=dbdata(m_dbproc, 6))
                UtilStrCpy(m_txn.NewOrder.c_credit, pData,
                dbdatlen(m_dbproc, 6));
            if (pData=dbdata(m_dbproc, 7))
            {
                datetime = (*(DBDATETIME *) pData);
                dbdatecrack(m_dbproc, &daterec, &datetime);
                m_txn.NewOrder.o_entry_d.year =
                m_txn.NewOrder.o_entry_d.month =
                m_txn.NewOrder.o_entry_d.day =
                m_txn.NewOrder.o_entry_d.hour =
                m_txn.NewOrder.o_entry_d.minute =
                m_txn.NewOrder.o_entry_d.second =
                daterec.year;
                daterec.month;
                daterec.day;
                daterec.hour;
                daterec.minute;
                daterec.second;
            }
            if (pData=dbdata(m_dbproc, 8))
                commit_flag = (*(DBTINYINT *) pData);

            DiscardNextRows(0);
            DiscardNextResults(0);

            if (commit_flag == 1)
            {
                m_txn.NewOrder.total_amount *= ((1 +
                m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 - m_txn.NewOrder.c_discount));
                m_txn.NewOrder.exec_status_code = eOK;
            }
            else
                m_txn.NewOrder.exec_status_code =
                eInvalidItem;

            return;
        }
        catch (CSQLERR *e)
        {
            if ((e->m_msgno == 1205 ||
                (e->m_msgno == iErrOleDbProvider &&

```

```

        strstr(e->m_msgtext, sErrTimeoutExpired) !=
NULL)) &&
        {
            ++iTryCount <= iMaxRetries)
            // hit deadlock; backoff for increasingly
            longer period
            delete e;
            Sleep(10 * iTryCount);
        }
        else
            throw;
    } // while (TRUE)
// if (iTryCount)
// throw new CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

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

    int              iTryCount = 0;
    const BYTE      *pData;

    ResetError();

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

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

            // 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 =
    m_txn.Payment.h_date.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
            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;
    DBDATERECD daterec;

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

    ResetError();

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

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

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

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

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

```

```

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

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

    if(pData=dbdata(m_dbproc, 1))

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

        if(pData=dbdata(m_dbproc, 3))

            m_txn.OrderStatus.OL[i].ol_quantity = (*DBSMALLINT *) pData);
            if(pData=dbdata(m_dbproc, 4))
                dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,4),
                                SQLFLT8,
(BYTE *) &m_txn.OrderStatus.OL[i].ol_amount, 8);
                if(pData=dbdata(m_dbproc, 5))
                {
                    datetime = *((DBDATETIME *)
                                dbdatecrack(m_dbproc, &daterec,
                                &datetime);

                    m_txn.OrderStatus.OL[i].ol_delivery_d.year = daterec.year;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.month = daterec.month;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.day = daterec.day;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.hour = daterec.hour;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.minute = daterec.minute;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.second = daterec.second;
                }
                i++;
            m_txn.OrderStatus.o_ol_cnt = i;

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

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

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

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

```

```

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

        DiscardNextRows(0);
        DiscardNextResults(0);

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

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

```

```

                delete e;
                Sleep(10 * iTryCount);
            }
            else
                throw;
        }
        // while (TRUE)
    }
    // if (iTryCount)
    // throw new CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
    iTryCount);
}

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

    ResetError();

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

            dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE
*) &m_txn.Delivery.w_id);

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

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

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

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

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

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

            DiscardNextRows(0);
            DiscardNextResults(0);

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

```

```

                strstr(e->m_msgtext, sErrTimeoutExpired) !=
NULL)) &&
                (++iTryCount <= iMaxRetries))
            {
                // hit deadlock; backoff for increasingly
                longer period
                delete e;
                Sleep(10 * iTryCount);
            }
            else
                throw;
        }
        // while (TRUE)
    }
    // if (iTryCount)
    // throw new CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
    iTryCount);
}

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

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

    return;
}

```

tpcc_dblib.h

```

/* FILE: TPC_C_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
dbsqlxec
            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
dbrpcxec
            eDbSetMaxProcs, // error from
dbsetmaxprocs
            eDbProcHandler // error from either
dbprocerrhandle or dbprocmsghandle
        };
        CDBLIBERR(ACTION eAction, int severity = 0, int dberror = 0, int
oserr = 0)

```

```

{
        m_eAction = eAction;
        m_severity = severity;
        m_dberror = dberror;
        m_oserr = oserr;

        m_dberrstr = NULL;
        m_oserrstr = NULL;
};

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

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

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

class CTPCC_DBLIB_ERR : public CBaseErr
{
    public:
        enum CTPCC_DBLIB_ERRS
        {
            ERR_WRONG_SP_VERSION = 1, // "Wrong version of
stored procs on database server"
            ERR_INVALID_CUST, // "Invalid
Customer id,name."
            ERR_NO_SUCH_ORDER, // "No orders
found for customer."
            ERR_RETRIED_TRANS, // "Retries
before transaction succeeded."
        };
        CTPCC_DBLIB_ERR( int iErr ) { m_errno = iErr; m_iTryCount = 0;
};
        CTPCC_DBLIB_ERR( int iErr, int iTryCount ) { m_errno = iErr;
m_iTryCount = iTryCount; };
        int          m_errno;
        int          m_iTryCount;

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

        char *ErrorText();
};

class DllDecl CTPCC_DBLIB : public CTPCC_BASE
{
    private:
        // declare variables and private functions here...

```

```

        PDBPROCESS          m_dbproc;
        CDBLIBERR *m_DbLibErr;          // not allocated until
needed (maybe never)
        CSQLERR             *m_SqlErr;          //
not allocated until needed (maybe never)
        int                 m_MaxRetries;      //
retry count on deadlock

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

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

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

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

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

        // these are public because they must be called from the dblib
err_handler and msg_handler
        // 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);

```

tpcc_odbc.cpp

```

/*      FILE:          TPC_C_ODBC.CPP
*      Microsoft TPC-C Kit Ver. 4.20.000
*      Copyright Microsoft, 1999
*
*      All Rights Reserved
*
*      Version 4.10.000 audited by Richard Gimarc,
Performance Metrics, 3/17/99
*
*      PURPOSE:  Implements ODBC calls for TPC-C txns.
*      Contact:  Charles Levine (clevine@microsoft.com)
*
*      Change history:
*      4.20.000 - updated rev number to match kit
*      4.10.001 - not deleting error class in catch handler on deadlock
retry;
*      not a functional bug, but a memory leak
*/

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

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

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

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

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

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

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

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

static SQLHENV henv = SQL_NULL_HENV;          // ODBC
environment handle

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

```

```

                break;
        case DLL_PROCESS_DETACH:
            if (henv != NULL)
                SQLFreeEnv(henv);
            break;
        default:
            /* nothing */;
    }
    return TRUE;
}

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

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

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

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

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

CTPCC_ODBC::CTPCC_ODBC (
    LPCSTR szServer,          // name of SQL server

```

```

    LPCSTR szUser,           // user name
for login
    LPCSTR szPassword,     // password for login
    LPCSTR szHost,         // not used
    LPCSTR szDatabase      // name of database to
use
    )
{
    RETCODE rc;

    // initialization
    m_hdbc = SQL_NULL_HDBC;
    m_hstmt = SQL_NULL_HSTMT;

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

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

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

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

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

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

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

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

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

    {
        char buffer[128];

        // set some options affecting connection behavior
        strcpy(buffer, "set nocount on ");
        strcat(buffer, "set XACT_ABORT ON ");

        // for coyote
        strcat(buffer, "set ansi_warnings on ");
        strcat(buffer, "set ansi_nulls on ");
    }
}

```

```

rc = SQLExecDirect(m_hstmt, (unsigned char *)buffer, SQL_NTS);
if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
    ThrowError(CODBCERR::eExecDirect);

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

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

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

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

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

    pODBCerr = new CODBCERR();

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

    szTmp[0] = 0;
    while (TRUE)

```

```

{
    rc = SQLError(henv, m_hdbc, m_hstmt, (BYTE *)&szState,
&lNativeError,
        (BYTE *)&szMsg, sizeof(szMsg),
    NULL);

    if (rc == SQL_NO_DATA)
        break;

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

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

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

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

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

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

SQLFreeStmt(m_hstmt, SQL_CLOSE);
throw pODBCerr;
}

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

    m_hstmt = m_hstmtStockLevel;

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

```

```

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

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

    m_hstmt = m_hstmtStockLevel;

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

            if ( SQLFetch(m_hstmt) == SQL_ERROR )
                ThrowError(CODBCERR::eFetch);

            SQLFreeStmt(m_hstmt, SQL_CLOSE);

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

            // hit deadlock; backoff for increasingly longer
            period

            delete e;
            Sleep(10 * iTryCount);
        }
    }

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

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

    m_hstmt = m_hstmtNewOrder;

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

```

```

        int i = 0;
        if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.NewOrder.w_id, 0, NULL) != SQL_SUCCESS
            || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.NewOrder.d_id, 0, NULL) != SQL_SUCCESS
            || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_txn.NewOrder.c_id, 0, NULL) != SQL_SUCCESS
            || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.NewOrder.o_ol_cnt, 0, NULL) != SQL_SUCCESS
            || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.NewOrder.o_all_local, 0, NULL) !=
SQL_SUCCESS
        )
            ThrowError(CODBCERR::eBindParam);

        for (int j=0; j<MAX_OL_NEW_ORDER_ITEMS; j++)
        {
            if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SLONG, SQL_INTEGER, 0, 0, &m_txn.NewOrder.OL[j].ol_i_id, 0, NULL) !=
SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SSHORT, SQL_SMALLINT, 0, 0, &m_txn.NewOrder.OL[j].ol_supply_w_id, 0, NULL) !=
SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SSHORT, SQL_SMALLINT, 0, 0, &m_txn.NewOrder.OL[j].ol_quantity, 0, NULL) !=
SQL_SUCCESS
            )
                ThrowError(CODBCERR::eBindParam);
        }

#ifdef new_order_strstr
        // 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);
#endif

        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);
    #else
        // prototype to eliminate patindex in server; shift work to client
        i = 0;
        if ( SQLBindCol(m_hstmt, ++i, SQL_C_CHAR, &m_ol_i_name,
sizeof(m_ol_i_name), NULL) != SQL_SUCCESS
            || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT, &m_ol_stock, 0, NULL)
!= SQL_SUCCESS
            || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR, &m_i_data,
sizeof(m_i_data), NULL) != SQL_SUCCESS
            || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR, &m_s_data,
sizeof(m_s_data), NULL) != SQL_SUCCESS
        )

```



```

        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE, &m_ol_i_price, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE, &m_ol_amount, 0, NULL)
!= SQL_SUCCESS
        )
        ThrowError(CODBCERR::eBindCol);
#endif

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

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

void CTPCC_ODBC::NewOrder()
{
    int          RETCODE          rc;          i;
    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;
    wcsncpy( &szSqlTemplate[i], L"");

```

```

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

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

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

            if ( SQLFetch(m_hstmt) == SQL_ERROR)
                ThrowError(CODBCERR::eFetch);
#else
            if ( SQLFetch(m_hstmt) == SQL_ERROR)
                ThrowError(CODBCERR::eFetch);

            strcpy( m_txn.NewOrder.OL[i].ol_i_name,
m_ol_i_name );

            if ( strstr(m_i_data, "ORIGINAL") != NULL &&
strstr(m_s_data, "ORIGINAL") != NULL )
                m_txn.NewOrder.OL[i].ol_brand_generic[0] = 'B';
            else
                m_txn.NewOrder.OL[i].ol_brand_generic[0] = 'G';
            m_txn.NewOrder.OL[i].ol_brand_generic[1] =
0;

            m_txn.NewOrder.OL[i].ol_stock
= m_ol_stock;
            m_txn.NewOrder.OL[i].ol_i_price
= m_ol_i_price;
            m_txn.NewOrder.OL[i].ol_amount
= m_ol_amount;
#endif

            // move to the next resultset
            if ( SQLMoreResults(m_hstmt) == SQL_ERROR )
                ThrowError(CODBCERR::eMoreResults);

```

```

        m_txn.NewOrder.total_amount +=
m_txn.NewOrder.OL[i].ol_amount;
    }

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

    if ( SQLFetch(m_hstmt) == SQL_ERROR )
        ThrowError(CODBCERR::eFetch);

    SQLFreeStmt(m_hstmt, SQL_CLOSE);

    if (m_no_commit_flag == 1)
    {
        m_txn.NewOrder.total_amount *= ((1 +
m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 - m_txn.NewOrder.c_discount));
        m_txn.NewOrder.exec_status_code = eOK;
    }
    else
        m_txn.NewOrder.exec_status_code =

eInvalidItem;

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

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

}

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

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

    m_hstmt = m_hstmtPayment;

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Payment.w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Payment.c_w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_DOUBLE,
SQL_NUMERIC, 6, 2, &m_txn.Payment.h_amount, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.Payment.d_id, 0, NULL) != SQL_SUCCESS

```

```

        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.Payment.c_d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_txn.Payment.c_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_CHAR,
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,
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.Payment.c_since,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_credit,
sizeof(m_txn.Payment.c_credit), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.Payment.c_credit_lim,0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.Payment.c_discount,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.Payment.c_balance,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_data,
sizeof(m_txn.Payment.c_data), NULL) !=
SQL_SUCCESS
    )
    ThrowError(CODBCERR::eBindCol);
}

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

    m_hstmt = m_hstmtPayment;

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

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

            if ( SQLFetch(m_hstmt) == SQL_ERROR)
                ThrowError(CODBCERR::eFetch);

            SQLFreeStmt(m_hstmt, SQL_CLOSE);

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

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

            // hit deadlock; backoff for increasingly longer
            period

            delete e;
            Sleep(10 * iTryCount);
        }
    }
}

```

```

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

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

    m_hstmt = m_hstmtOrderStatus;

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

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.OrderStatus.w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.OrderStatus.d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_txn.OrderStatus.c_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, sizeof(m_txn.OrderStatus.c_last), 0, &m_txn.OrderStatus.c_last,
sizeof(m_txn.OrderStatus.c_last), NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindParam);

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

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

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

```

```

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

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

    m_hstmt = m_hstmtOrderStatus;

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

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

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

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

            // configure block cursor
            if ( SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)MAX_OL_ORDER_STATUS_ITEMS, 0) != SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);

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

            m_txn.OrderStatus.o_ol_cnt = (short)m_RowsFetched;
        }
    }
}

```

```

        if (m_txn.OrderStatus.o_ol_cnt != 0)
        {
            if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols2, SQL_IS_POINTER ) != SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);

            if ( SQLMoreResults(m_hstmt) == SQL_ERROR )
                ThrowError(CODBCERR::eMoreResults);

            if ( (rc = SQLFetch(m_hstmt)) == SQL_ERROR )
                ThrowError(CODBCERR::eFetch);
        }

        SQLFreeStmt(m_hstmt, SQL_CLOSE);

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

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

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

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

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

    m_hstmt = m_hstmtDelivery;

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Delivery.w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Delivery.o_carrier_id, 0, NULL) != SQL_SUCCESS
        )
        ThrowError(CODBCERR::eBindParam);

    for (i=0;i<10;i++)

```

```

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

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

    m_hstmt = m_hstmtDelivery;

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

            if ( SQLFetch(m_hstmt) == SQL_ERROR )
                ThrowError(CODBCERR::eFetch);

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

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

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

```

tpcc_odbc.h

```

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

```

```

#pragma once

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

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

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

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

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

    int ErrorType() {return ERR_TYPE_ODBC;};
};

```

```

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

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

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

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

    int m_errno;
    int m_iTryCount;

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

    char *ErrorText();
};

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

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

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

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

    // new-order specific fields
    SQLINTEGER m_BindOffset;
    SQLINTEGER m_RowsFetched;
    int m_no_commit_flag;
};

```

```

#ifdef new_order_strstr
    // for new-order txn;
    // output params
    char m_ol_i_name[I_NAME_LEN+1];
    double m_ol_i_price;
    double m_ol_amount;
    short m_ol_stock;
    // used locally, but not returned to caller
    char m_i_data[I_DATA_LEN];
    char m_s_data[S_DATA_LEN];
#endif

void ThrowError( CODBCERR::ACTION eAction );

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

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

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

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

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

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

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

```

trans.h

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

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

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

```
    unsigned short /* SQLSMALLINT */ second;
    unsigned long /* SQLINTEGER */ fraction;
} TIMESTAMP_STRUCT;
#endif

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

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

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

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

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

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

    // output params
};
```

```

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

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

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

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

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

    // output params

```

```

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

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

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

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

```

txn_base.h

```

/* FILE: TXN_BASE.H
 * Microsoft TPC-C Kit Ver. 4.20.000
 * Copyright Microsoft, 1999
 * All Rights Reserved
 * Version 4.10.000 audited by Richard Gimarc,
 * Performance Metrics, 3/17/99
 * PURPOSE: Header file for TPC-C txn class implementation.
 * Change history:
 * 4.20.000 - updated rev number to match kit
 */

#pragma once

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifdef DllDecl
#define DllDecl __declspec( dllimport )
#endif

class DllDecl CTPCC_BASE
{
public:
    CTPCC_BASE(void) {};
    virtual ~CTPCC_BASE(void) {};

    virtual PNEW_ORDER_DATA BuffAddr_NewOrder()
    = 0;
    virtual PPAYMENT_DATA BuffAddr_Payment()
    = 0;

```



```

        virtual PDELIVERY_DATA          BuffAddr_Delivery()
= 0;
        virtual PSTOCK_LEVEL_DATA      BuffAddr_StockLevel() = 0;
        virtual PORDER_STATUS_DATA     BuffAddr_OrderStatus() = 0;

        virtual void NewOrder          () = 0;
        virtual void Payment            () = 0;
        virtual void Delivery           () = 0;
        virtual void StockLevel         () = 0;
        virtual void OrderStatus        () = 0;
};

```

txnlog.h

```

/* FILE: TXNLOG.H Microsoft TPC-C Kit Ver. 4.10.000
 * not yet audited
 *
 * PURPOSE: Header file for txn log class
 * Copyright Microsoft, 1999
 * All Rights Reserved
 */
#pragma once

typedef struct _TXN_NEWORDER
{
    BYTE    OL_Count;          //range 0 to 31
    BYTE    OL_Remote_Count;   //range 0 to 31
    WORD    c_id;
    int     o_id;
} TXN_NEWORDER;

typedef struct _TXN_PAYMENT
{
    BYTE    CustByName;
    BYTE    IsRemote;
} TXN_PAYMENT;

typedef struct _TXN_ORDERSTATUS
{
    BYTE    CustByName;
} TXN_ORDERSTATUS;

typedef union _TXN_DETAILS
{
    TXN_NEWORDER    NewOrder;
    TXN_PAYMENT     Payment;
    TXN_ORDERSTATUS OrderStatus;
} TXN_DETAILS;

// Common header for all records in txn log. The TxnType field is
// a switch which identifies the particular variant.
#define TXN_REC_TYPE_CONTROL 1 //
#define TXN_REC_TYPE_TPCC 2 // replaces
TRANSACTION_TYPE_TPCC
#define TXN_REC_TYPE_TPCC_DELIV_DEF 3

typedef struct _TXN_RECORD_HEADER
{

```

```

        JULIAN_TIME          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
WORD     w_id;           // warehouse ID
BYTE     d_id;          // assigned district ID
for this thread
BYTE     d_id_ThisTxn;   // district ID chosen for this
particular
BYTE     TxnStatus;      // completion status for
txn to indicate errors
BYTE     reserved;       // for word alignment
        TXN_DETAILS     TxnDetails;    //
    } TXN_RECORD_TPCC, *PTXN_RECORD_TPCC;
//
// TPC-C Deferred Delivery Txn Record Layout:
//
//Incorporating delivery transaction information into the above
//structure would increase the size of TXN_DETAILS from 8 to 42 bytes.
//Hence, we store delivery transaction details in a separate structure.
//
typedef struct _TXN_RECORD_TPCC_DELIV_DEF
{
    // common header; must exactly match TXN_RECORD_HEADER
    JULIAN_TIME     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)
    WORD     w_id;           // warehouse ID
    BYTE     TxnStatus;       // completion status for
txn to indicate errors
    BYTE     reserved;       // for word alignment
    short    o_carrier_id;    // carrier id
    long     o_id[10];        // returned delivery transaction
ids
} TXN_RECORD_TPCC_DELIV_DEF, *PTXN_RECORD_TPCC_DELIV_DEF;

#define TXN_LOG_VERSION 1
#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
// the record map provides a fast way to get close to a
particular timestamp in a sorted log file.
//
// {
//     JULIAN_TIME     TS;
//     int      iPos;
//     int      iPos;
//     int      iPos;
// }
// byte position in file
// RecMap[RecMapSize];
// #define RecMapSize 200
} TXN_LOG_HEADER, *PTXN_LOG_HEADER;

#define READ_BUFFER_SIZE 64*1024
#define WRITE_BUFFER_SIZE 8*1024

#define NUM_READ_BUFFERS 1
#define NUM_WRITE_BUFFERS 2
#define MAX_NUM_BUFFERS 2

// flags passed in to the constructor
#define TXN_LOG_WRITE 0x01
#define TXN_LOG_READ 0x02
#define TXN_LOG_SORTED 0x04

#define TXN_LOG_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.
    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

```



```
/Ep"webcnct\obj\webcnct.pch" /YX /Fo"webcnct\obj/" /c
"D:\wctest\webcnct\src\webcnct.c"
pp"
ctest\tpccstub\src\tpccidl_i.c"
src\dlldata.c"
"D:\wctest\tpccstub\src\tpccidl_p.c"
"D:\wctest\tpccstub\src\tpccidl_i.c"
```

```
Control-Break          ?Datatype Misalignment          Control-C
Violation              ?Access
In Page Error          No Memory
Instruction
Disposition            ? Float Denormal
Operand                ? Float Divide by Zero      ? Float Inexact
Result                 ? Float Invalid Operation
```

Float Overflow[]
Underflow[]

Stack Overflow[]
Exception[]

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;
        };
    };

};

////////////////////////////////////
// CTPCC
class CTPCC :
public CTPCC_Common,
public CComCoClass<CTPCC, &CLSID_TPCCC>
{
public:

```



```

DECLARE_REGISTRY_RESOURCEID(IDR_TPCC)

BEGIN_COM_MAP(CTPCC)
    COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
    COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

};

////////////////////////////////////
// CNewOrder
class CNewOrder :
    public CTPCC_Common,
    public CComCoClass<CNewOrder, &CLSID_NewOrder>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_NEWORDER)

BEGIN_COM_MAP(CNewOrder)
    COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
    COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
// HRESULT __stdcall NewOrder(          VARIANT txn_in, VARIANT* txn_out)
{return E_NOTIMPL;}
// HRESULT __stdcall Payment(          VARIANT txn_in, VARIANT* txn_out)
{return E_NOTIMPL;}
// HRESULT __stdcall StockLevel( VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
// HRESULT __stdcall OrderStatus(      VARIANT txn_in, VARIANT* txn_out)
{return E_NOTIMPL;}
};

////////////////////////////////////
// COrderStatus
class COrderStatus :
    public CTPCC_Common,
    public CComCoClass<COrderStatus, &CLSID_OrderStatus>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_ORDERSTATUS)

BEGIN_COM_MAP(COrderStatus)
    COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
    COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
// HRESULT __stdcall NewOrder(          VARIANT txn_in, VARIANT* txn_out)
{return E_NOTIMPL;}
// HRESULT __stdcall Payment(          VARIANT txn_in, VARIANT* txn_out)
{return E_NOTIMPL;}
// HRESULT __stdcall StockLevel( VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
// HRESULT __stdcall OrderStatus(      VARIANT txn_in, VARIANT* txn_out)
{return E_NOTIMPL;}
};

```

```

////////////////////////////////////
// CPayment
class CPayment :
    public CTPCC_Common,
    public CComCoClass<CPayment, &CLSID_Payment>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_PAYMENT)

BEGIN_COM_MAP(CPayment)
    COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
    COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
// HRESULT __stdcall NewOrder(          VARIANT txn_in, VARIANT* txn_out)
{return E_NOTIMPL;}
// HRESULT __stdcall Payment(          VARIANT txn_in, VARIANT* txn_out)
{return E_NOTIMPL;}
// HRESULT __stdcall StockLevel( VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
// HRESULT __stdcall OrderStatus(      VARIANT txn_in, VARIANT* txn_out)
{return E_NOTIMPL;}
};

////////////////////////////////////
// CStockLevel
class CStockLevel :
    public CTPCC_Common,
    public CComCoClass<CStockLevel, &CLSID_StockLevel>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_STOCKLEVEL)

BEGIN_COM_MAP(CStockLevel)
    COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
    COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
// HRESULT __stdcall NewOrder(          VARIANT txn_in, VARIANT* txn_out)
{return E_NOTIMPL;}
// HRESULT __stdcall Payment(          VARIANT txn_in, VARIANT* txn_out)
{return E_NOTIMPL;}
// HRESULT __stdcall StockLevel( VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
// HRESULT __stdcall OrderStatus(      VARIANT txn_in, VARIANT* txn_out)
{return E_NOTIMPL;}
};

```

```

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 ( !strcmp(szTmp, szDBNames[ODBC]) )
            pReg->eDB_Protocol = ODBC;
        else if ( !strcmp(szTmp, szDBNames[DBLIB]) )
            pReg->eDB_Protocol = DBLIB;
    }

    pReg->eTxnMon = None;
    // determine txn monitor to use; may be either TUXEDO, or blank
    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey, "TxnMonitor", 0, &type, (BYTE *)szTmp, &size)
== ERROR_SUCCESS )
    {
        if ( !strcmp(szTmp, szTxnMonNames[TUXEDO]) )
            pReg->eTxnMon = TUXEDO;
        else if ( !strcmp(szTmp, szTxnMonNames[ENCINA]) )
            pReg->eTxnMon = ENCINA;
        else if ( !strcmp(szTmp, szTxnMonNames[COM]) )
            pReg->eTxnMon = COM;
    }

    pReg->bCOM_SinglePool = FALSE;
    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey, "COM_SinglePool", 0, &type, (BYTE *)szTmp,
&size) == ERROR_SUCCESS )
    {
        if ( !strcmp(szTmp, "YES") )

```

```

        pReg->bCOM_SinglePool = TRUE;
    }

    pReg->dwMaxConnections = 0;
    size = sizeof(dwTmp);
    if ( ( RegQueryValueEx(hKey, "MaxConnections", 0, &type, (LPBYTE)&dwTmp,
&size) == ERROR_SUCCESS )
        && (type == REG_DWORD) )
        pReg->dwMaxConnections = dwTmp;

    pReg->dwMaxPendingDeliveries = 0;
    size = sizeof(dwTmp);
    if ( ( RegQueryValueEx(hKey, "MaxPendingDeliveries", 0, &type,
(LPBYTE)&dwTmp, &size) == ERROR_SUCCESS )
        && (type == REG_DWORD) )
        pReg->dwMaxPendingDeliveries = dwTmp;

    pReg->dwNumberOfDeliveryThreads = 0;
    size = sizeof(dwTmp);
    if ( ( RegQueryValueEx(hKey, "NumberOfDeliveryThreads", 0, &type,
(LPBYTE)&dwTmp, &size) == ERROR_SUCCESS )
        && (type == REG_DWORD) )
        pReg->dwNumberOfDeliveryThreads = dwTmp;

    size = sizeof( pReg->szPath );
    if ( RegQueryValueEx(hKey, "Path", 0, &type, (BYTE *)pReg->szPath, &size)
!= ERROR_SUCCESS )
        pReg->szPath[0] = 0;

    size = sizeof( pReg->szDbServer );
    if ( RegQueryValueEx(hKey, "DbServer", 0, &type, (BYTE *)pReg-
>szDbServer, &size) != ERROR_SUCCESS )
        pReg->szDbServer[0] = 0;

    size = sizeof( pReg->szDbName );
    if ( RegQueryValueEx(hKey, "DbName", 0, &type, (BYTE *)pReg->szDbName,
&size) != ERROR_SUCCESS )
        pReg->szDbName[0] = 0;

    size = sizeof( pReg->szDbUser );
    if ( RegQueryValueEx(hKey, "DbUser", 0, &type, (BYTE *)pReg->szDbUser,
&size) != ERROR_SUCCESS )
        pReg->szDbUser[0] = 0;

    size = sizeof( pReg->szDbPassword );
    if ( RegQueryValueEx(hKey, "DbPassword", 0, &type, (BYTE *)pReg-
>szDbPassword, &size) != ERROR_SUCCESS )
        pReg->szDbPassword[0] = 0;

    RegCloseKey(hKey);

    return FALSE;
}

```

ReadRegistry.h

```

/*   FILE:           ReadRegistry.h
*
*   Microsoft TPC-C Kit Ver. 4.20.000
*   Copyright Microsoft, 1999
*
*   All Rights Reserved
*
*   not audited

```

```

*
*      PURPOSE: Header for registry related code.
*
* Change history:
*      4.20.000 - first version
*/

enum DBPROTOCOL { Unspecified, ODBC, DBLIB };
const char *szDBNames[] = { "Unspecified", "ODBC", "DBLIB" };

enum TXNMON { None, TUXEDO, ENCINA, COM };
const char *szTxnMonNames[] = { "NONE", "TUXEDO", "ENCINA", "COM" };

//This structure defines the data necessary to keep distinct for each terminal or
client connection.
typedef struct _TPCCREGISTRYDATA
{
    enum DBPROTOCOL eDB_Protocol;
    enum TXNMON eTxnMon;
    BOOL bCOM_SinglePool;
    DWORD dwMaxConnections;
    DWORD dwMaxPendingDeliveries;
    DWORD dwNumberOfDeliveryThreads;
    char szPath[128];
    char szDbServer[32];
    char szDbName[32];
    char szDbUser[32];
    char szDbPassword[32];
} TPCCREGISTRYDATA, *PTPCCREGISTRYDATA;

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );

```

WEBCLNT.DSP

```

# Microsoft Developer Studio Project File - Name="webclnt" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 5.00
# ** DO NOT EDIT **

# TARGETTYPE "Win32 (x86) Application" 0x0101

CFG=webclnt - Win32 Release
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "Webclnt.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "Webclnt.mak" CFG="webclnt - Win32 Release"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "webclnt - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "webclnt - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

# Begin Project
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe

```

```

RSC=rc.exe

!IF "$(CFG)" == "webclnt - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir ".\Release"
# PROP BASE Intermediate_Dir ".\Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\Release"
# PROP Intermediate_Dir ".\Release"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:I386

!ELSEIF "$(CFG)" == "webclnt - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir ".\Debug"
# PROP BASE Intermediate_Dir ".\Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\Debug"
# PROP Intermediate_Dir ".\Debug"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/c
# ADD CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD BASE MTL /nologo /D "_DEBUG" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386

!ENDIF

```

```

# Begin Target

# Name "webclnt - Win32 Release"
# Name "webclnt - Win32 Debug"
# End Target
# End Project

```

Webclnt.dsw

Microsoft Developer Studio Workspace File, Format Version 6.00
WARNING: DO NOT EDIT OR DELETE THIS WORKSPACE FILE!

```

#####

Project: "db_dblib_dll"=.\db_dblib_dll\db_dblib_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

#####

Project: "db_odbc_dll"=.\db_odbc_dll\db_odbc_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

#####

Project: "install"=.\install\install.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name isapi_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tuxapp
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_dblib_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_odbc_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_com_dll
    End Project Dependency
}}}

```

```

    Begin Project Dependency
    Project_Dep_Name tm_tuxedo_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tpcc_com_all
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tpcc_com_ps
    End Project Dependency
}}}

#####

Project: "isapi_dll"=.\isapi_dll\isapi_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name db_dblib_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_odbc_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_tuxedo_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_com_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_encina_dll
    End Project Dependency
}}}

#####

Project: "tm_com_dll"=.\tm_com_dll\tm_com_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name tpcc_com_ps
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tpcc_com_all
    End Project Dependency
}}}

#####

Project: "tm_encina_dll"=.\tm_encina_dll\tm_encina_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

```

```

Package=<4>
{{{
}}}

#####

Project: "tm_tuxedo_dll"=.\tm_tuxedo_dll\tm_tuxedo_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

#####

Project: "tpcc_com_all"=.\tpcc_com_all\tpcc_com_all.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name tpcc_com_ps
    End Project Dependency
}}}

#####

Project: "tpcc_com_ps"=.\tpcc_com_ps\tpcc_com_ps.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

#####

Project: "tuxapp"=.\tuxapp\tuxapp.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name db_dblib_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_odbc_dll
    End Project Dependency
}}}

#####

```

```

Global:

Package=<5>
{{{
}}}

Package=<3>
{{{
}}}

#####

```

com_all_resource.h

```

//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by tpcc_com_all.rc
//
#define IDS_PROJNAME                100
#define IDR_TPCC                    101
#define IDR_NEWORDER                102
#define IDR_ORDERSTATUS            103
#define IDR_PAYMENT                 104
#define IDR_STOCKLEVEL             105

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifdef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE    202
#define _APS_NEXT_COMMAND_VALUE    32768
#define _APS_NEXT_CONTROL_VALUE    201
#define _APS_NEXT_SYMED_VALUE      106
#endif
#endif

```

db_dblib_dll.dsp

```

# Microsoft Developer Studio Project File - Name="db_dblib_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGETTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=db_dblib_dll - Win32 IceCAP
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "db_dblib_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "db_dblib_dll.mak" CFG="db_dblib_dll - Win32 IceCAP"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "db_dblib_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link
Library")

```

```

!MESSAGE "db_dblib_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_dblib_dll - Win32 IceCAP" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF "$(CFG)" == "db_dblib_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /1 0x409 /d "NDEBUG"
# ADD RSC /1 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ntdbllib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /machine:I386 /out:".bin\tpcc_dblib.dll"

!ELSEIF "$(CFG)" == "db_dblib_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD BASE MTL /nologo /D "DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /1 0x409 /d "_DEBUG"

```

```

# ADD RSC /1 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ntdbllib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_dblib.dll"
/pdbtype:sept

!ELSEIF "$(CFG)" == "db_dblib_dll - Win32 IceCAP"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "db_dblib"
# PROP BASE Intermediate_Dir "db_dblib"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /Gm /GX /ZI /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"ICECAP" /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /1 0x409 /d "_DEBUG"
# ADD RSC /1 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 ntdbllib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_dblib.dll"
/pdbtype:sept
# ADD LINK32 icap.lib ntdbllib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_dblib.dll"
/pdbtype:sept

!ENDIF

# Begin Target

# Name "db_dblib_dll - Win32 Release"
# Name "db_dblib_dll - Win32 Debug"
# Name "db_dblib_dll - Win32 IceCAP"
# Begin Group "Source"

# PROP Default_Filter "*.cpp"
# Begin Source File

SOURCE=.src\tpcc_dblib.cpp
# End Source File
# End Group

```

```

# Begin Group "Header"

# PROP Default_Filter "*.h"
# Begin Source File

SOURCE=..\common\src\error.h
# End Source File
# Begin Source File

SOURCE=..\src\tpcc_dblib.h
# End Source File
# Begin Source File

SOURCE=..\common\src\trans.h
# End Source File
# Begin Source File

SOURCE=..\common\src\txn_base.h
# End Source File
# End Group
# End Target
# End Project

```

db_odbc_dll.dsp

```

# Microsoft Developer Studio Project File - Name="db_odbc_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGETTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=db_odbc_dll - Win32 IceCAP
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "db_odbc_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "db_odbc_dll.mak" CFG="db_odbc_dll - Win32 IceCAP"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "db_odbc_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_odbc_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_odbc_dll - Win32 IceCAP" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF "$ (CFG)" == "db_odbc_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"

```

```

# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386 /out:".bin\tpcc_odbc.dll"

!ELSEIF "$ (CFG)" == "db_odbc_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /Mdd /W3 /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_odbc.dll"
/pdbtype:sept

!ELSEIF "$ (CFG)" == "db_odbc_dll - Win32 IceCAP"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1

```

```

# PROP BASE Output_Dir "db_odbc_"
# PROP BASE Intermediate_Dir "db_odbc_"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /Gm /GX /Zi /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"ICECAP" /YX /FD /Gh /c
# ADD BASE MTL /nologo /D " _DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D " _DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_odbc.dll"
/pdbtype:sept
# ADD LINK32 icap.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_odbc.dll"
/pdbtype:sept

!ENDIF

# Begin Target

# Name "db_odbc_dll - Win32 Release"
# Name "db_odbc_dll - Win32 Debug"
# Name "db_odbc_dll - Win32 IceCAP"
# Begin Group "Source"

# PROP Default_Filter "*.cpp"
# Begin Source File

SOURCE=.\src\tpcc_odbc.cpp
# End Source File
# End Group
# Begin Group "Header"

# PROP Default_Filter "*.h"
# Begin Source File

SOURCE=.\common\src\error.h
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_odbc.h
# End Source File
# Begin Source File

SOURCE=.\common\src\trans.h
# End Source File
# Begin Source File

```

```

SOURCE=.\common\src\txn_base.h
# End Source File
# End Group
# End Target
# End Project

```

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 */

```

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_INS_MEMORY     "Insufficient Memory to continue."
#define ERR_UNKNOWN       "Unknown error."
#define ERR_MSG_BUF_SIZE  512
#define INV_ERROR_CODE    -1

class CBaseErr
{
public:
    CBaseErr(LPCTSTR szLoc = NULL)
    {
        m_idMsg = INV_ERROR_CODE;
        if (szLoc)
        {
            m_szLoc = new char[m_szLoc_size];
            strcpy(m_szLoc, szLoc);
        }
        else
            m_szLoc = NULL;

        m_szApp = new char[m_szApp_size];
        GetModuleFileName(GetModuleHandle(NULL), m_szApp, m_szApp_size);
    }

    CBaseErr(int idMsg, LPCTSTR szLoc = NULL)
    {
        m_idMsg = idMsg;
        if (szLoc)
        {
            m_szLoc = new char[m_szLoc_size];
            strcpy(m_szLoc, szLoc);
        }
        else
            m_szLoc = NULL;

        m_szApp = new char[m_szApp_size];
        GetModuleFileName(GetModuleHandle(NULL), m_szApp, m_szApp_size);
    }

    virtual ~CBaseErr(void)
    {
        if (m_szApp)
            delete [] m_szApp;
    }
}

```

```

        if (m_szLoc)
            delete [] m_szLoc;
    };

    virtual void Draw(HWND hwnd, LPCTSTR szStr = NULL)
    {
        int j = 0;
        char szTmp[512];

        if (szStr)
            j = wsprintf(szTmp, "%s\n", szStr);
        if (ErrorNum() != INV_ERROR_CODE)
            j += wsprintf(szTmp+j, "Error = %d\n", ErrorNum());
        if (m_szLoc)
            j += wsprintf(szTmp+j, "Location = %s\n",
                GetLocation());

        j += wsprintf(szTmp+j, "%s\n", ErrorText());

        ::MessageBox(hwnd, szTmp, m_szApp, MB_OK);
    }

    char *GetApp(void) { return m_szApp; }
    char *GetLocation(void) { return m_szLoc; }
    virtual int ErrorNum() { return m_idMsg; }
    virtual int ErrorType() = 0; // a value which distinguishes the kind of
    error that occurred
    virtual char *ErrorText() = 0; // a string (i.e., human readable)
    representation of the error

protected:
    char *m_szApp;
    char *m_szLoc; // code location where the error occurred
    int m_idMsg;
};

class CSocketErr : public CBaseErr
{
public:
    enum Action
    {
        eNone,
        eSend,
        eSocket,
        eBind,
        eConnect,
        eListen,
        eHost,
        eRecv,
    };

    CSocketErr(Action eAction, LPCTSTR szLocation = NULL);
    Action m_eAction;

    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,
    };

    CSystemErr(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;};
};

```

install.c

```

/* FILE: INSTALL.C
 * Microsoft TPC-C Kit Ver. 4.20.000
 * Copyright Microsoft, 1999
 * All Rights Reserved
 *
 * not audited
 *
 * PURPOSE: Automated installation application for TPC-C Web Kit
 * Contact: Charles Levine (clevine@microsoft.com)
 *
 * Change history:

```

```

*           4.20.000 - added COM installation steps
*/

#include <windows.h>
#include <direct.h>
#include <io.h>
#include <stdlib.h>
#include <stdio.h>
#include <commctrl.h>
#include "..\..\common\src\ReadRegistry.h"

#include "resource.h"

#define WM_INITTEXT WM_USER+100

HICON hIcon;
HINSTANCE hInst;

DWORD versionExeMS;
DWORD versionExeLS;
DWORD versionExeMM;
DWORD versionDllMS;
DWORD versionDllLS;

// TPC-C registry settings
TPCPCREGISTRYDATA Reg;

static int iPoolThreadLimit;
static int iThreadTimeout;
static int iListenBackLog;
static int iAcceptExOutstanding;

static int iMaxPhysicalMemory; //max physical memory in
MB
static char szLastFileName[64]; // last file we worked on (for
error reporting)

BOOL CALLBACK LicenseDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM
lParam);
BOOL CALLBACK UpdatedDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM
lParam);
BOOL CALLBACK MainDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam);
BOOL CALLBACK CopyDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam);
static void ProcessOK(HWND hwnd, char *szDllPath);
static void ReadRegistrySettings(void);
static void WriteRegistrySettings(char *szDllPath);
static BOOL RegisterDLL(char *szFileName);
static int CopyFiles(HWND hDlg, char *szDllPath);
static BOOL GetInstallPath(char *szDllPath);
static void GetVersionInfo(char *szDLLPath, char *szExePath);
static BOOL CheckWWWWebService(void);
static BOOL StartWWWWebService(void);
static BOOL StopWWWWebService(void);
static void UpdateDialog(HWND hDlg);

BOOL install_com(char *szDllPath);

#include "..\..\common\src\ReadRegistry.cpp"

int WINAPI WinMain( HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine,
int nCmdShow )
{

```

```

int iRc;

hInst = hInstance;

InitCommonControls();

hIcon = LoadIcon(hInstance, MAKEINTRESOURCE(IDI_ICON1));

iRc = DialogBox(hInstance, MAKEINTRESOURCE(IDD_DIALOG4),
GetDesktopWindow(), LicenseDlgProc);
if ( iRc )
{
    iRc = DialogBox(hInstance, MAKEINTRESOURCE(IDD_DIALOG1),
GetDesktopWindow(), MainDlgProc);
    if ( iRc )
    {
        DialogBoxParam(hInstance,
MAKEINTRESOURCE(IDD_DIALOG2), GetDesktopWindow(), UpdatedDlgProc, (LPARAM)iRc);
    }

    DestroyIcon(hIcon);
    return 0;
}

BOOL CALLBACK LicenseDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    HGLOBAL hRes;
    HRSRC hResInfo;
    BYTE *pSrc, *pDst;
    DWORD dwSize;
    static HFONT hFont;

    switch(uMsg)
    {
        case WM_INITDIALOG:
            hFont = CreateFont(-12, 0, 0, 0, 400, 0, 0, 0, 0, 0,
0, 0, 0, "Arial");
            SendMessage( GetDlgItem(hwnd, IDR_LICENSE1),
WM_SETFONT, (WPARAM)hFont, MAKELPARAM(0, 0) );
            PostMessage(hwnd, WM_INITTEXT, (WPARAM)0, (LPARAM)0);
            return TRUE;
        case WM_INITTEXT:
            hResInfo = FindResource(hInst,
MAKEINTRESOURCE(IDR_LICENSE1), "LICENSE");
            dwSize = SizeofResource(hInst, hResInfo);
            hRes = LoadResource(hInst, hResInfo );
            pSrc = (BYTE *)LockResource(hRes);
            pDst = (unsigned char *)malloc(dwSize+1);
            if ( pDst )
            {
                memcpy(pDst, pSrc, dwSize);
                pDst[dwSize] = 0;
                SetDlgItemText(hwnd, IDC_LICENSE, (const
char *)pDst);

                free(pDst);
            }
            else
                SetDlgItemText(hwnd, IDC_LICENSE, (const
char *)pSrc);

            return TRUE;
        case WM_DESTROY:
            DeleteObject(hFont);
    }
}

```

```

        return TRUE;
    case WM_COMMAND:
        if ( wParam == IDOK )
            EndDialog(hwnd, TRUE);
        if ( wParam == IDCANCEL )
            EndDialog(hwnd, FALSE);
    default:
        break;
    }
    return FALSE;
}

BOOL CALLBACK UpdatedDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    switch(uMsg)
    {
        case WM_INITDIALOG:
            switch(lParam)
            {
                case 1:
                case 2:
                    SetDlgItemText(hwnd, IDC_RESULTS,
"TPC-C Web Client Installed");
                    break;
            }
            return TRUE;
    case WM_COMMAND:
        if ( wParam == IDOK )
            EndDialog(hwnd, TRUE);
        break;
    default:
        break;
    }
    return FALSE;
}

BOOL CALLBACK MainDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    PAINTSTRUCT ps;
    MEMORYSTATUS memoryStatus;
    OSVERSIONINFO VI;
    char szTmp[256];
    static char szDllPath[256];
    static char szExePath[256];

    switch(uMsg)
    {
        case WM_INITDIALOG:
            GlobalMemoryStatus(&memoryStatus);
            iMaxPhysicalMemory = (memoryStatus.dwTotalPhys/
1048576);

            if ( GetInstallPath(szDllPath) )
            {
                MessageBox(hwnd, "Error internet service
inetsrv is not installed.", NULL, MB_ICONSTOP | MB_OK);
                EndDialog(hwnd, FALSE);
                return TRUE;
            }

            // set default values
            ZeroMemory( &Reg, sizeof(Reg) );
            Reg.dwNumberOfDeliveryThreads = 4;

```

```

            Reg.dwMaxConnections = 100;
            Reg.dwMaxPendingDeliveries = 100;
            Reg.eDB_Protocol = DBLIB;
            Reg.eTxnMon = None;
            strcpy(Reg.szDbServer, "");
            strcpy(Reg.szDbName, "tpcc");
            strcpy(Reg.szDbUser, "sa");
            strcpy(Reg.szDbPassword, "");

            iPoolThreadLimit = iMaxPhysicalMemory * 2;
            iThreadTimeout = 86400;
            iListenBackLog = 15;
            iAcceptExOutstanding = 40;

            ReadTPCCRegistrySettings( &Reg );
            ReadRegistrySettings();

            GetModuleFileName(hInst, szExePath,
sizeof(szExePath));
            GetVersionInfo(szDllPath, szExePath);

            wsprintf(szTmp, "Version %d.%2.2d.%3.3d",
versionExeMS, versionExeMM, versionExeLS);
            SetDlgItemText(hwnd, IDC_VERSION, szTmp);

            SetDlgItemText(hwnd, IDC_PATH, szDllPath);

            SetDlgItemText(hwnd, ED_DB_SERVER, Reg.szDbServer);
            SetDlgItemText(hwnd, ED_DB_USER_ID, Reg.szDbUser);
            SetDlgItemText(hwnd, ED_DB_PASSWORD,
Reg.szDbPassword);
            SetDlgItemText(hwnd, ED_DB_NAME, Reg.szDbName);

            SetDlgItemInt(hwnd, ED_THREADS,
Reg.dwNumberOfDeliveryThreads, FALSE);
            SetDlgItemInt(hwnd, ED_MAXCONNECTION,
Reg.dwMaxConnections, FALSE);
            SetDlgItemInt(hwnd, ED_MAXDELIVERIES,
Reg.dwMaxPendingDeliveries, FALSE);
            SetDlgItemInt(hwnd, ED_IIS_MAX_THREAD_POOL_LIMIT,
iPoolThreadLimit, FALSE);
            SetDlgItemInt(hwnd, ED_IIS_THREAD_TIMEOUT,
iThreadTimeout, FALSE);
            SetDlgItemInt(hwnd, ED_IIS_LISTEN_BACKLOG,
iListenBackLog, FALSE);
            SetDlgItemInt(hwnd, ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,
iAcceptExOutstanding, FALSE);

            CheckDlgButton(hwnd, IDC_DBLIB, 0);
            CheckDlgButton(hwnd, IDC_ODBC, 0);
            if ( Reg.eDB_Protocol == DBLIB )
                CheckDlgButton(hwnd, IDC_DBLIB, 1);
            else
                CheckDlgButton(hwnd, IDC_ODBC, 1);

            // check OS version level for COM. Must be at least
Windows 2000
            VI.dwOSVersionInfoSize = sizeof(VI);
            GetVersionEx( &VI );
            if (VI.dwMajorVersion < 5)
            {
                HWND hDlg = GetDlgItem( hwnd, IDC_TM_MTS );

```

```

        EnableWindow( hDlg, 0 );    // disable COM
option
        if (Reg.eTxnMon == COM)
            Reg.eTxnMon = None;
    }

    CheckDlgButton(hwnd, IDC_TM_NONE, 0);
    CheckDlgButton(hwnd, IDC_TM_TUXEDO, 0);
    CheckDlgButton(hwnd, IDC_TM_MTS, 0);
    CheckDlgButton(hwnd, IDC_TM_ENCINA, 0);
    switch (Reg.eTxnMon)
    {
    case None:
        CheckDlgButton(hwnd, IDC_TM_NONE, 1);
        break;
    case TUXEDO:
        CheckDlgButton(hwnd, IDC_TM_TUXEDO, 1);
        break;
    case ENCINA:
        CheckDlgButton(hwnd, IDC_TM_ENCINA, 1);
        break;
    case COM:
        CheckDlgButton(hwnd, IDC_TM_MTS, 1);
        break;
    }

    return TRUE;
case WM_PAINT:
    if ( IsIconic(hwnd) )
    {
        BeginPaint(hwnd, &ps);
        DrawIcon(ps.hdc, 0, 0, hIcon);
        EndPaint(hwnd, &ps);
        return TRUE;
    }
    break;
case WM_COMMAND:
    if ( HIWORD(wParam) == BN_CLICKED )
    {
        switch( LOWORD(wParam) )
        {
            case IDC_DBLIB:
                return TRUE;
            case IDC_ODBC:
                return TRUE;
            case IDOK:
                ProcessOK(hwnd,
                    szDllPath);
                return TRUE;
            case IDCANCEL:
                EndDialog(hwnd, FALSE);
                return TRUE;
            default:
                return FALSE;
        }
    }
    break;
default:
    break;
}
return FALSE;
}

```

```

static void ProcessOK(HWND hwnd, char *szDllPath)
{
    int         d;
    HWND       hDlg;
    int         rc;

    char        szFullName[256];
    char        szErrTxt[128];

    // read settings from dialog
    Reg.dwNumberOfDeliveryThreads = GetDlgItemInt(hwnd, ED_THREADS, &d,
FALSE);
    Reg.dwMaxConnections = GetDlgItemInt(hwnd, ED_MAXCONNECTION, &d, FALSE);
    Reg.dwMaxPendingDeliveries = GetDlgItemInt(hwnd, ED_MAXDELIVERIES, &d,
FALSE);

    GetDlgItemText(hwnd, ED_DB_SERVER, Reg.szDbServer,
sizeof(Reg.szDbServer));
    GetDlgItemText(hwnd, ED_DB_USER_ID, Reg.szDbUser, sizeof(Reg.szDbUser));
    GetDlgItemText(hwnd, ED_DB_PASSWORD, Reg.szDbPassword,
sizeof(Reg.szDbPassword));
    GetDlgItemText(hwnd, ED_DB_NAME, Reg.szDbName, sizeof(Reg.szDbName));

    if ( IsDlgButtonChecked(hwnd, IDC_DBLIB) )
    {
        Reg.eDB_Protocol = DBLIB;
        rc = 1;
    }
    else if ( IsDlgButtonChecked(hwnd, IDC_ODBC) )
    {
        Reg.eDB_Protocol = ODBC;
        rc = 2;
    }

    if ( IsDlgButtonChecked(hwnd, IDC_TM_NONE) )
        Reg.eTxnMon = None;
    else if ( IsDlgButtonChecked(hwnd, IDC_TM_TUXEDO) )
        Reg.eTxnMon = TUXEDO;
    else if ( IsDlgButtonChecked(hwnd, IDC_TM_MTS) )
        Reg.eTxnMon = COM;
    else if ( IsDlgButtonChecked(hwnd, IDC_TM_ENCINA) )
        Reg.eTxnMon = ENCINA;

    iPoolThreadLimit = GetDlgItemInt(hwnd, ED_IIS_MAX_THREAD_POOL_LIMIT, &d,
FALSE);
    iThreadTimeout = GetDlgItemInt(hwnd, ED_IIS_THREAD_TIMEOUT, &d, FALSE);
    iListenBackLog = GetDlgItemInt(hwnd, ED_IIS_LISTEN_BACKLOG, &d, FALSE);
    iAcceptExOutstanding = GetDlgItemInt(hwnd,
ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE, &d, FALSE);

    ShowWindow(hwnd, SW_HIDE);
    hDlg = CreateDialog(hInst, MAKEINTRESOURCE(IDD_DIALOG3), hwnd,
CopyDlgProc);
    ShowWindow(hDlg, SW_SHOWNA);
    UpdateDialog(hDlg);

    // write binaries to inetpub\wwwroot
    rc = CopyFiles(hDlg, szDllPath);
    if ( !rc )
    {
        ShowWindow(hwnd, SW_SHOWNA);
    }
}

```

```

DestroyWindow(hDlg);
strcpy( szErrTxt, "Error(s) ocured when creating " );
strcat( szErrTxt, szLastFileName );
MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP | MB_OK);
EndDialog(hwnd, 0);
return;
}

// update registry
SetDlgItemText(hDlg, IDC_STATUS, "Updating Registry.");
SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
UpdateDialog(hDlg);
WriteRegistrySettings(szDllPath);

// register com proxy stub
strcpy(szFullName, szDllPath);
strcat(szFullName, "tpcc_com_ps.dll");
if (!RegisterDLL(szFullName))
{
    ShowWindow(hwnd, SW_SHOWNA);
    DestroyWindow(hDlg);
    strcpy( szErrTxt, "Error ocured when registering " );
    strcat( szErrTxt, szFullName );
    MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP | MB_OK);
    EndDialog(hwnd, 0);
    return;
}

// if using COM
if (Reg.eTnxMon == 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 ocured when configuring COM
settings." );
        MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP | MB_OK);
        EndDialog(hwnd, 0);
        return;
    }
}

Sleep(100);

ShowWindow(hwnd, SW_SHOWNA);
DestroyWindow(hDlg);

EndDialog(hwnd, rc);
return;
}

static void ReadRegistrySettings(void)
{
    HKEY    hKey;
    DWORD   size;
    DWORD   type;

```

```

        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Parameters", 0, KEY_READ, &hKey) ==
ERROR_SUCCESS )
        {
            size = sizeof(iPoolThreadLimit);
            if ( RegQueryValueEx(hKey, "PoolThreadLimit", 0, &type, (char
*)&iPoolThreadLimit, &size) == ERROR_SUCCESS )
                if ( !iPoolThreadLimit )
                    iPoolThreadLimit = iMaxPhysicalMemory * 2;

            size = sizeof(iThreadTimeout);
            if ( RegQueryValueEx(hKey, "ThreadTimeout", 0, &type, (char
*)&iThreadTimeout, &size) == ERROR_SUCCESS )
                if ( !iThreadTimeout )
                    iThreadTimeout = 86400;

            size = sizeof(iListenBackLog);
            if ( RegQueryValueEx(hKey, "ListenBackLog", 0, &type, (char
*)&iListenBackLog, &size) == ERROR_SUCCESS )
                if ( !iListenBackLog )
                    iListenBackLog = 15;

            RegCloseKey(hKey);
        }

        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters", 0, KEY_READ, &hKey) ==
ERROR_SUCCESS )
        {
            size = sizeof(iAcceptExOutstanding);
            if ( RegQueryValueEx(hKey, "AcceptExOutstanding", 0, &type,
(char *)&iAcceptExOutstanding, &size) == ERROR_SUCCESS )
                if ( !iAcceptExOutstanding )
                    iAcceptExOutstanding = 40;

            RegCloseKey(hKey);
        }
    }

static void WriteRegistrySettings(char *szDllPath)
{
    HKEY    hKey;
    DWORD   dwDisposition;
    char    szTmp[256];
    char    *ptr;
    int     iRc;

    if ( RegCreateKeyEx(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\TPCC", 0,
NULL, REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition) ==
ERROR_SUCCESS )
    {
        strcpy(szTmp, szDllPath);
        ptr = strstr(szTmp, "tpcc");
        if ( ptr )
            *ptr = 0;

        RegSetValueEx(hKey, "Path", 0, REG_SZ, szTmp, strlen(szTmp)+1);

        RegSetValueEx(hKey, "NumberOfDeliveryThreads", 0, REG_DWORD,
(char *)&Reg.dwNumberOfDeliveryThreads, sizeof(Reg.dwNumberOfDeliveryThreads));
        RegSetValueEx(hKey, "MaxConnections", 0, REG_DWORD, (char
*)&Reg.dwMaxConnections, sizeof(Reg.dwMaxConnections));
    }
}

```

```

        RegSetValueEx(hKey, "MaxPendingDeliveries", 0, REG_DWORD, (char
*)&Reg.dwMaxPendingDeliveries, sizeof(Reg.dwMaxPendingDeliveries));

        RegSetValueEx(hKey, "DB_Protocol", 0, REG_SZ,
szDBNames[Reg.eDB_Protocol], strlen(szDBNames[Reg.eDB_Protocol])+1);
        RegSetValueEx(hKey, "TxnMonitor", 0, REG_SZ,
szTxnMonNames[Reg.eTxnMon], strlen(szTxnMonNames[Reg.eTxnMon])+1);

        RegSetValueEx(hKey, "DbServer", 0, REG_SZ, Reg.szDbServer,
strlen(Reg.szDbServer)+1);
        RegSetValueEx(hKey, "DbName", 0, REG_SZ, Reg.szDbName,
strlen(Reg.szDbName)+1);
        RegSetValueEx(hKey, "DbUser", 0, REG_SZ, Reg.szDbUser,
strlen(Reg.szDbUser)+1);
        RegSetValueEx(hKey, "DbPassword", 0, REG_SZ, Reg.szDbPassword,
strlen(Reg.szDbPassword)+1);

        strcpy(szTmp, "YES");
        RegSetValueEx(hKey, "COM_SinglePool", 0, REG_SZ, szTmp,
strlen(szTmp)+1);

        RegFlushKey(hKey);
        RegCloseKey(hKey);
    }

    if ( (iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Parameters", 0, NULL,
REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition)) ==
ERROR_SUCCESS )
    {
        RegSetValueEx(hKey, "PoolThreadLimit", 0, REG_DWORD, (char
*)&iPoolThreadLimit, sizeof(iPoolThreadLimit));
        RegSetValueEx(hKey, "ThreadTimeout", 0, REG_DWORD, (char
*)&iThreadTimeout, sizeof(iThreadTimeout));
        RegSetValueEx(hKey, "ListenBackLog", 0, REG_DWORD, (char
*)&iListenBackLog, sizeof(iListenBackLog));

        RegFlushKey(hKey);
        RegCloseKey(hKey);
    }

    if ( (iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters", 0, NULL,
REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition)) ==
ERROR_SUCCESS )
    {
        RegSetValueEx(hKey, "AcceptExOutstanding", 0, REG_DWORD, (char
*)&iAcceptExOutstanding, sizeof(iAcceptExOutstanding));

        RegFlushKey(hKey);
        RegCloseKey(hKey);
    }

    return;
}

BOOL CALLBACK CopyDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    if ( uMsg == WM_INITDIALOG )
    {
        SendDlgItemMessage(hwnd, IDC_PROGRESS1, PBM_SETRANGE, 0,
MAKELPARAM(0, 15));
    }
}

```

```

        SendDlgItemMessage(hwnd, IDC_PROGRESS1, PBM_SETSTEP, (LPARAM)1,
0);
        return TRUE;
    }
    return FALSE;
}

BOOL RegisterDLL(char *szFileName)
{
    HINSTANCE hLib;
    FARPROC lpDllEntryPoint;

    hLib = LoadLibrary(szFileName);
    if ( hLib == NULL )
        return FALSE;
    // Find the entry point.
    lpDllEntryPoint = GetProcAddress(hLib, "DllRegisterServer");
    if (lpDllEntryPoint != NULL)
    {
        return ((*lpDllEntryPoint)()) == S_OK;
    }
    else
        return FALSE; //unable to locate entry point
}

BOOL FileFromResource( char *szResourceName, int iResourceId, char *szDllPath, char
*szFileName )
{
    HGLOBAL hDLL;
    HRSRC hResInfo;
    HANDLE hFile;
    DWORD dwSize;
    BYTE *pSrc;
    DWORD d;
    char szFullName[256];

    hResInfo = FindResource(hInst, MAKEINTRESOURCE(iResourceId),
szResourceName);

    strcpy(szFullName, szDllPath);
    strcat(szFullName, szFileName);

    dwSize = SizeofResource(hInst, hResInfo);
    hDLL = LoadResource(hInst, hResInfo );
    pSrc = (BYTE *)LockResource(hDLL);
    remove(szFullName);

    if ( !hFile = CreateFile(szFullName, GENERIC_WRITE, 0, NULL,
CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL) )
        return FALSE;

    if ( !WriteFile(hFile, pSrc, dwSize, &d, NULL) )
        return FALSE;

    CloseHandle(hFile);

    UnlockResource(hDLL);
    FreeResource(hDLL);
    return TRUE;
}

static int CopyFiles(HWND hDlg, char *szDllPath)
{
}

```

```

BOOL                bSvcRunning;

bSvcRunning = CheckWWWebService();
if ( bSvcRunning )
{
    SetDlgItemText(hDlg, IDC_STATUS, "Stopping Web Service.");
    SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);

    StopWWWebService();
    SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);
}

SetDlgItemText(hDlg, IDC_STATUS, "Copying Files...");
SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
UpdateDialog(hDlg);

// install TPCC.DLL
strcpy( szLastFileName, "tpcc.dll" );
if (!FileFromResource( "TPCCDLL", IDR_TPCCDLL, szDllPath, szLastFileName
))
    return 0;
SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
UpdateDialog(hDlg);

// install tpcc_dblib.dll
strcpy( szLastFileName, "tpcc_dblib.dll" );
if (!FileFromResource( "DBLIB_DLL", IDR_DBLIB_DLL, szDllPath,
szLastFileName ))
    return 0;
SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
UpdateDialog(hDlg);

// install tpcc_odbc.dll
strcpy( szLastFileName, "tpcc_odbc.dll" );
if (!FileFromResource( "ODBC_DLL", IDR_ODBC_DLL, szDllPath, szLastFileName
))
    return 0;
SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
UpdateDialog(hDlg);

// install tuxapp.exe
strcpy( szLastFileName, "tuxapp.exe" );
if (!FileFromResource( "TUXEDO_APP", IDR_TUXEDO_APP, szDllPath,
szLastFileName ))
    return 0;
SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
UpdateDialog(hDlg);

// install tpcc_tuxedo.dll
strcpy( szLastFileName, "tpcc_tuxedo.dll" );
if (!FileFromResource( "TUXEDO_DLL", IDR_TUXEDO_DLL, szDllPath,
szLastFileName ))
    return 0;
SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
UpdateDialog(hDlg);

// install tpcc_com.dll
strcpy( szLastFileName, "tpcc_com.dll" );
if (!FileFromResource( "COM_DLL", IDR_COM_DLL, szDllPath, szLastFileName
))
    return 0;

```

```

SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
UpdateDialog(hDlg);

// install tpcc_com_ps.dll
strcpy( szLastFileName, "tpcc_com_ps.dll" );
if (!FileFromResource( "COM_PS_DLL", IDR_COMPS_DLL, szDllPath,
szLastFileName ))
    return 0;
SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
UpdateDialog(hDlg);

// install tpcc_com_all.dll
strcpy( szLastFileName, "tpcc_com_all.dll" );
if (!FileFromResource( "COM_ALL_DLL", IDR_COMALL_DLL, szDllPath,
szLastFileName ))
    return 0;
SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
UpdateDialog(hDlg);

//if we stopped service restart it.
if ( bSvcRunning )
{
    SetDlgItemText(hDlg, IDC_STATUS, "Starting Web Service.");
    SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);
    StartWWWebService();
}

SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
UpdateDialog(hDlg);

return 1;
}

static BOOL GetInstallPath(char *szDllPath)
{
    HKEY hKey;
    BYTE szData[256];
    DWORD sv;
    BOOL bRc;
    int len;
    char *ptr;
    int iRc;

    szDllPath[0] = 0;
    bRc = TRUE;
    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\Virtual Roots", 0,
KEY_ALL_ACCESS, &hKey) == ERROR_SUCCESS )
    {
        sv = sizeof(szData);
        iRc = RegQueryValueEx( hKey, "/", NULL, NULL, szData, &sv );
// used by IIS 3.0
        if (iRc == ERROR_FILE_NOT_FOUND)
            iRc = RegQueryValueEx( hKey, "/", NULL, NULL, szData,
&sv ); // used by IIS 4.0
        if (iRc == ERROR_SUCCESS)
        {
            bRc = FALSE;
            strcpy(szDllPath, szData);
            if ( (ptr = strchr(szDllPath, ',')) )
                *ptr = 0;
        }
    }
}

```



```

        len = strlen(szDllPath);
        if ( szDllPath[len-1] != '\\')
        {
            szDllPath[len] = '\\';
            szDllPath[len+1] = 0;
        }
    }
    RegCloseKey(hKey);
}

return bRc;
}

static void GetVersionInfo(char *szDLLPath, char *szExePath)
{
    DWORD          d;
    DWORD          dwSize;
    DWORD          dwBytes;
    char           *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 CheckWWWWebService(void)
{
    SC_HANDLE      schSCManager;
    SC_HANDLE      schService;
    SERVICE_STATUS ssStatus;

    schSCManager = OpenSCManager(NULL, NULL, SC_MANAGER_ALL_ACCESS);
    schService = OpenService(schSCManager, TEXT("W3SVC"), SERVICE_ALL_ACCESS);

```

```

    if (schService == NULL)
        return FALSE;

    if (! QueryServiceStatus(schService, &ssStatus) )
        goto ServiceNotRunning;

    if ( !ControlService(schService, SERVICE_CONTROL_STOP, &ssStatus) )
        goto ServiceNotRunning;
    //start Service pending, Check the status until the service is running.
    if (! QueryServiceStatus(schService, &ssStatus) )
        goto ServiceNotRunning;

    CloseServiceHandle(schService);
    return TRUE;
}

ServiceNotRunning:

    CloseServiceHandle(schService);
    return FALSE;
}

static BOOL StartWWWWebService(void)
{
    SC_HANDLE      schSCManager;
    SC_HANDLE      schService;
    SERVICE_STATUS ssStatus;
    DWORD          dwOldCheckPoint;

    schSCManager = OpenSCManager(NULL, NULL, SC_MANAGER_ALL_ACCESS);
    schService = OpenService(schSCManager, TEXT("W3SVC"), SERVICE_ALL_ACCESS);
    if (schService == NULL)
        return FALSE;

    if (! StartService(schService, 0, NULL) )
        goto StartWWWWebErr;
    //start Service pending, Check the status until the service is running.
    if (! QueryServiceStatus(schService, &ssStatus) )
        goto StartWWWWebErr;
    while( ssStatus.dwCurrentState != SERVICE_RUNNING)
    {
        dwOldCheckPoint = ssStatus.dwCheckPoint;
        //Save the current checkpoint.
        Sleep(ssStatus.dwWaitHint);
        //Wait for the specified interval.
        if ( !QueryServiceStatus(schService, &ssStatus) ) //Check the
status again.
            break;
        if (dwOldCheckPoint >= ssStatus.dwCheckPoint)
            //Break if the checkpoint has not been incremented.
            break;
    }

    if (ssStatus.dwCurrentState == SERVICE_RUNNING)
        goto StartWWWWebErr;

    CloseServiceHandle(schService);
    return TRUE;
}

StartWWWWebErr:
    CloseServiceHandle(schService);
    return FALSE;
}

```

```

static BOOL StopWWWebService(void)
{
    SC_HANDLE      schSCManager;
    SC_HANDLE      schService;
    SERVICE_STATUS ssStatus;
    DWORD          dwOldCheckPoint;

    schSCManager = OpenSCManager(NULL, NULL, SC_MANAGER_ALL_ACCESS);
    schService = OpenService(schSCManager, TEXT("W3SVC"), SERVICE_ALL_ACCESS);
    if (schService == NULL)
        return FALSE;

    if (! QueryServiceStatus(schService, &ssStatus) )
        goto StopWWWebErr;

    if ( !ControlService(schService, SERVICE_CONTROL_STOP, &ssStatus) )
        goto StopWWWebErr;
    //start Service pending, Check the status until the service is running.
    if (! QueryServiceStatus(schService, &ssStatus) )
        goto StopWWWebErr;
    while( ssStatus.dwCurrentState == SERVICE_RUNNING)
    {
        dwOldCheckPoint = ssStatus.dwCheckPoint;
        //Save the current checkpoint.
        Sleep(ssStatus.dwWaitHint);
        //Wait for the specified interval.
        if ( !QueryServiceStatus(schService, &ssStatus) ) //Check the
status again.
            break;
        if (dwOldCheckPoint >= ssStatus.dwCheckPoint)
            //Break if the checkpoint has not been incremented.
            break;
    }

    if (ssStatus.dwCurrentState == SERVICE_RUNNING)
        goto StopWWWebErr;

    CloseServiceHandle(schService);
    return TRUE;

StopWWWebErr:
    CloseServiceHandle(schService);
    return FALSE;
}

static void UpdateDialog(HWND hDlg)
{
    MSG msg;

    UpdateWindow(hDlg);
    while( PeekMessage(&msg, hDlg, 0, 0, PM_REMOVE) )
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    Sleep(250);
    return;
}

```

install.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_THREAD_POOL_LIMIT 1015
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE 1017
#define ED_IIS_THREAD_TIMEOUT 1018
#define ED_IIS_LISTEN_BACKLOG 1019
#define IDC_DBLIB 1021
#define IDC_ODBC 1022
#define IDC_CONNECT_POOL 1024
#define ED_USER_CONNECT_DELAY_TIME 1024

// Next default values for new objects
//

```

install.rc

```

//Microsoft Developer Studio generated resource script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 2 resource.
//
#include "afxres.h"
////////////////////////////////////
#undef APSTUDIO_READONLY_SYMBOLS

////////////////////////////////////
// English (U.S.) resources

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US

```

```

#pragma code_page(1252)
#endif // WIN32

////////////////////////////////////
//
// Dialog
//

IDD_DIALOG1 DIALOGEX 0, 0, 219, 351
STYLE DS_MODALFRAME | DS_CENTER | WS_MINIMIZEBOX | WS_POPUP | WS_CAPTION |
WS_SYSMENU
CAPTION "TPC-C Web Client Installation Utility"
FONT 8, "MS Sans Serif"
BEGIN
EDITTEXT      ED_THREADS,164,45,34,12,ES_RIGHT | ES_NUMBER,
               WS_EX_RTLREADING
EDITTEXT      ED_MAXDELIVERIES,164,59,34,12,ES_RIGHT | ES_NUMBER,
               WS_EX_RTLREADING
EDITTEXT      ED_MAXCONNECTION,164,73,34,12,ES_RIGHT | ES_NUMBER,
               WS_EX_RTLREADING
CONTROL       "None",IDC_TM_NONE,"Button",BS_AUTORADIOBUTTON |
               WS_GROUP | WS_TABSTOP,43,100,33,10
CONTROL       "COM",IDC_TM_MTS,"Button",BS_AUTORADIOBUTTON |
               WS_TABSTOP,43,113,32,10
CONTROL       "TUXEDO",IDC_TM_TUXEDO,"Button",BS_AUTORADIOBUTTON |
               WS_TABSTOP,106,100,46,10
CONTROL       "ENCINA",IDC_TM_ENCINA,"Button",BS_AUTORADIOBUTTON |
               WS_DISABLED | WS_TABSTOP,106,113,43,10
EDITTEXT      ED_DB_SERVER,131,152,67,12,ES_AUTOHSCROLL
EDITTEXT      ED_DB_USER_ID,131,165,67,12,ES_AUTOHSCROLL
EDITTEXT      ED_DB_PASSWORD,131,178,67,12,ES_AUTOHSCROLL
EDITTEXT      ED_DB_NAME,131,191,67,12,ES_AUTOHSCROLL
CONTROL       "DBLIB",IDC_DBLIB,"Button",BS_AUTORADIOBUTTON | WS_GROUP |
               WS_TABSTOP,45,219,39,12
CONTROL       "ODBC",IDC_ODBC,"Button",BS_AUTORADIOBUTTON | WS_TABSTOP,
               91,219,39,12
EDITTEXT      ED_IIS_MAX_THREAD_POOL_LIMIT,164,263,34,12,ES_RIGHT |
               ES_NUMBER,WS_EX_RTLREADING
EDITTEXT      ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,164,277,34,12,ES_RIGHT |
               ES_NUMBER,WS_EX_RTLREADING
EDITTEXT      ED_IIS_THREAD_TIMEOUT,164,291,34,12,ES_RIGHT | ES_NUMBER,
               WS_EX_RTLREADING
EDITTEXT      ED_IIS_LISTEN_BACKLOG,164,305,34,12,ES_RIGHT | ES_NUMBER,
               WS_EX_RTLREADING
DEFPUSHBUTTON "OK",IDOK,53,331,50,14
PUSHBUTTON    "Cancel",IDCANCEL,119,331,50,14
EDITTEXT      IDC_PATH,106,26,91,13,ES_AUTOHSCROLL | ES_READONLY
LTEXT         "Number of Delivery Threads:",IDC_STATIC,35,45,115,12
LTEXT         "Max Number of Connections:",IDC_STATIC,35,73,115,12
RTEXT         "Version 4.11",IDC_VERSION,120,4,89,9
LTEXT         "IIS Max Thread Pool Limit:",IDC_STATIC,36,263,115,12
LTEXT         "Web Service Backlog Queue Size:",IDC_STATIC,36,277,115,
               12
LTEXT         "IIS Thread Timeout (seconds):",IDC_STATIC,36,291,115,12
LTEXT         "IIS Listen Backlog:",IDC_STATIC,36,307,115,10
GROUPBOX     "Database Interface",IDC_STATIC,35,208,163,27,WS_GROUP
LTEXT         "Installation directory:",IDC_STATIC,35,29,71,10
GROUPBOX     "Transaction Monitor",IDC_STATIC,33,90,165,37
LTEXT         "Server Name:",IDC_STATIC,35,155,56,8
LTEXT         "User ID:",IDC_STATIC,35,168,60,8
LTEXT         "User Password:",IDC_STATIC,35,181,83,8
LTEXT         "Database Name:",IDC_STATIC,35,194,54,8
GROUPBOX     "SQL Server Connection Properties",IDC_STATIC,22,139,187,

```

```

102
GROUPBOX     "Web Client Properties",IDC_STATIC,22,15,187,118
GROUPBOX     "IIS Settings",IDC_STATIC,22,247,187,79
LTEXT         "Max Pending Deliveries:",IDC_STATIC,35,59,115,12
END

IDD_DIALOG2 DIALOGEX 0, 0, 117, 62
STYLE DS_SETFOREGROUND | DS_3DLOOK | DS_CENTER | WS_POPUP | WS_BORDER
EXSTYLE WS_EX_STATICEDGE
FONT 12, "MS Sans Serif", 0, 0, 0x1
BEGIN
DEFPUSHBUTTON "OK",IDOK,33,45,50,9
CTEXT         "HTML TPC-C Installation Successful",IDC_RESULTS,7,22,
               102,18,0,WS_EX_CLIENTEDGE
ICON          IDI_ICON2,IDC_STATIC,50,7,18,20,SS_REALSIZEIMAGE,
               WS_EX_TRANSPARENT
END

IDD_DIALOG3 DIALOG DISCARDABLE 0, 0, 91, 40
STYLE DS_SYSMODAL | DS_MODALFRAME | DS_3DLOOK | DS_CENTER | WS_CAPTION
CAPTION "Installing TPC-C Web Client"
FONT 12, "Arial Black"
BEGIN
CONTROL       "Progress1",IDC_PROGRESS1,"msctls_progress32",WS_BORDER,
               7,20,77,13
CTEXT         "Static",IDC_STATUS,7,7,77,12,SS_SUNKEN
END

IDD_DIALOG4 DIALOG DISCARDABLE 0, 0, 291, 202
STYLE DS_MODALFRAME | DS_CENTER | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Client End User License"
FONT 8, "MS Sans Serif"
BEGIN
EDITTEXT      IDC_LICENSE,7,7,271,167,ES_MULTILINE | ES_AUTOVSCROLL |
               ES_AUTOHSCROLL | ES_READONLY | WS_VSCROLL | WS_HSCROLL
DEFPUSHBUTTON "I &Agree",IDOK,87,181,50,14
PUSHBUTTON    "&Cancel",IDCANCEL,153,181,50,14
END

////////////////////////////////////
//
// DESIGNINFO
//

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
IDD_DIALOG1, DIALOG
BEGIN
LEFTMARGIN, 22
RIGHTMARGIN, 209
VERTGUIDE, 35
VERTGUIDE, 198
TOPMARGIN, 4
BOTTOMMARGIN, 345
END

IDD_DIALOG2, DIALOG
BEGIN
LEFTMARGIN, 7
RIGHTMARGIN, 109
TOPMARGIN, 7

```

```

        BOTTOMMARGIN, 54
    END

    IDD_DIALOG3, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 84
        TOPMARGIN, 7
        BOTTOMMARGIN, 33
    END

    IDD_DIALOG4, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 278
        TOPMARGIN, 7
        BOTTOMMARGIN, 195
    END
END
#endif // APSTUDIO_INVOKED

#ifdef APSTUDIO_INVOKED
////////////////////////////////////
//
// TEXTINCLUDE
//
1 TEXTINCLUDE DISCARDABLE
BEGIN
    "resource.h\0"
END

2 TEXTINCLUDE DISCARDABLE
BEGIN
    "#include \"afxres.h\"\r\n"
    "\0"
END

3 TEXTINCLUDE DISCARDABLE
BEGIN
    "\r\n"
    "\0"
END

#endif // APSTUDIO_INVOKED

////////////////////////////////////
//
// Icon
//
// Icon with lowest ID value placed first to ensure application icon
// remains consistent on all systems.
IDI_ICON1      ICON      DISCARDABLE      "icon1.ico"
IDI_ICON2      ICON      DISCARDABLE      "icon2.ico"

////////////////////////////////////
//
// TPCCDLL
//

```

```

IDR_TPCCDLL      TPCCDLL DISCARDABLE      "..\..\isapi_dll\bin\tpcc.dll"

#ifdef _MAC
////////////////////////////////////
//
// Version
//

VS_VERSION_INFO VERSIONINFO
FILEVERSION 0,4,20,0
PRODUCTVERSION 0,4,20,0
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x40004L
FILETYPE 0x1L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904b0"
        BEGIN
            VALUE "Comments", "TPC-C Web Client Installer\0"
            VALUE "CompanyName", "Microsoft\0"
            VALUE "FileDescription", "install\0"
            VALUE "FileVersion", "0, 4, 20, 0\0"
            VALUE "InternalName", "install\0"
            VALUE "LegalCopyright", "Copyright © 1999\0"
            VALUE "OriginalFilename", "install.exe\0"
            VALUE "ProductName", "Microsoft install\0"
            VALUE "ProductVersion", "0, 4, 20, 0\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END
#endif // !_MAC

////////////////////////////////////
//
// LICENSE
//

IDR_LICENSE1      LICENSE DISCARDABLE      "license.txt"

////////////////////////////////////
//
// DBLIB_DLL
//

IDR_DBLIB_DLL      DBLIB_DLL DISCARDABLE
"..\..\db_dblib_dll\bin\tpcc_dblib.dll"

////////////////////////////////////
//
// ODBC_DLL
//

```

```

//
IDR_ODBC_DLL          ODBC_DLL DISCARDABLE
"..\\..\\db_odbc_dll\\bin\\tpcc_odbc.dll"

////////////////////////////////////
//
// TUXEDO_APP
//

IDR_TUXEDO_APP       TUXEDO_APP DISCARDABLE  "..\\..\\tuxapp\\bin\\tuxapp.exe"

////////////////////////////////////
//
// TUXEDO_DLL
//

IDR_TUXEDO_DLL       TUXEDO_DLL DISCARDABLE
"..\\..\\tm_tuxedo_dll\\bin\\tpcc_tuxedo.dll"

////////////////////////////////////
//
// COM_DLL
//

IDR_COM_DLL          COM_DLL DISCARDABLE
"..\\..\\tm_com_dll\\bin\\tpcc_com.dll"

////////////////////////////////////
//
// COM_PS_DLL
//

IDR_COMPS_DLL        COM_PS_DLL DISCARDABLE
"..\\..\\tpcc_com_ps\\bin\\tpcc_com_ps.dll"

////////////////////////////////////
//
// COM_ALL_DLL
//

IDR_COMALL_DLL       COM_ALL_DLL DISCARDABLE
"..\\..\\tpcc_com_all\\bin\\tpcc_com_all.dll"
#endif // English (U.S.) resources

////////////////////////////////////
//
//
//

#ifndef APSTUDIO_INVOKED
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//

////////////////////////////////////
#endif // not APSTUDIO_INVOKED



---


install_com.cpp


---


/* FILE: INSTALL_COM.CPP

```

```

*
* Microsoft TPC-C Kit Ver. 4.20.000
* Copyright Microsoft, 1999
*
* All Rights Reserved
*
* not audited
*
* PURPOSE: installation code for COM application for TPC-C Web Kit
* Contact: Charles Levine (clevine@microsoft.com)
*
* Change history:
* 4.20.000 - first version
*/

#define _WIN32_WINNT 0x0500

#include <comdef.h>
#include <comadmin.h>
#include <stdio.h>
#include <tchar.h>

extern "C"
{
    BOOL install_com(char *szDllPath);
}

BOOL install_com(char *szDllPath)
{
    ICOMAdminCatalog* pCOMAdminCat = NULL;
    ICatalogCollection* pCatalogCollectionApp = NULL;
    ICatalogCollection* pCatalogCollectionCo = NULL;
    ICatalogCollection* pCatalogCollectionItf = NULL;
    ICatalogCollection* pCatalogCollectionMethod = NULL;

    ICatalogObject* pCatalogObjectApp = NULL;
    ICatalogObject* pCatalogObjectCo = NULL;
    ICatalogObject* pCatalogObjectItf = NULL;
    ICatalogObject* pCatalogObjectMethod = NULL;

    _bstr_t bstrTemp, bstrTemp2, bstrTemp3,
bstrTemp4;
    _bstr_t bstrDllPath = szDllPath;
    _variant_t vTmp, vKey;
    long lActProp, lCount, lCountCo,
lCountItf, lCountMethod;
    bool bTmp;

    CoInitializeEx(NULL, COINIT_MULTITHREADED);

    HRESULT hr = CoCreateInstance(CLSID_COMAdminCatalog,
NULL,
CLSCTX_INPROC_SERVER,
IID_ICOMAdminCatalog,
(void**) &pCOMAdminCat);

    if (!SUCCEEDED(hr)) goto Error;

    bstrTemp = "Applications";

```

```

// Attempt to connect to "Applications" in the Catalog
hr = pCOMAdminCat->GetCollection(bstrTemp,

(IDispatch**) &pCatalogCollectionApp);
if (!SUCCEEDED(hr)) goto Error;

// Attempt to load the "Applications" collection
hr = pCatalogCollectionApp->Populate();
if (!SUCCEEDED(hr)) goto Error;

hr = pCatalogCollectionApp->get_Count(&lCount);
if (!SUCCEEDED(hr)) goto Error;

// iterate through applications to delete existing "TPC-C" application (if
any)
while (lCount > 0)
{
    hr = pCatalogCollectionApp->get_Item(lCount - 1, (IDispatch**)
&pCatalogObjectApp);
    if (!SUCCEEDED(hr)) goto Error;

    hr = pCatalogObjectApp->get_Name(&vTmp);
    if (!SUCCEEDED(hr)) goto Error;

    if (wcsncmp(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";
bstrTemp2 = bstrDllPath + "tpcc_com_all.dll"; // app name
DLL //

bstrTemp3 = "";
// type library (TLB)
bstrTemp4 = bstrDllPath + "tpcc_com_ps.dll"; //
proxy/stub dll

hr = pCOMAdminCat->InstallComponent(bstrTemp,

bstrTemp2,

bstrTemp3,

bstrTemp4);
if (!SUCCEEDED(hr)) goto Error;

bstrTemp = "Components";
hr = pCatalogCollectionApp->GetCollection(bstrTemp, vKey, (IDispatch**)
&pCatalogCollectionCo);
if (!SUCCEEDED(hr)) goto Error;

hr = pCatalogCollectionCo->Populate();
if (!SUCCEEDED(hr)) goto Error;

hr = pCatalogCollectionCo->get_Count(&lCountCo);
if (!SUCCEEDED(hr)) goto Error;

// iterate through components in application and set the properties
while (lCountCo > 0)
{
    hr = pCatalogCollectionCo->get_Item(lCountCo - 1, (IDispatch**)
&pCatalogObjectCo);
    if (!SUCCEEDED(hr)) goto Error;

    // used for debugging (view the name)
    hr = pCatalogObjectCo->get_Name(&vTmp);
    if (!SUCCEEDED(hr)) goto Error;

    bstrTemp = "ConstructionEnabled";
    bTmp = TRUE;
    vTmp = bTmp;
    hr = pCatalogObjectCo->put_Value(bstrTemp, vTmp);
    if (!SUCCEEDED(hr)) goto Error;

    bstrTemp = "ConstructorString";
    bstrTemp2 = "dummy string (do not remove)";
    vTmp = bstrTemp2;
    hr = pCatalogObjectCo->put_Value(bstrTemp, vTmp);
    if (!SUCCEEDED(hr)) goto Error;
}

```

```

        bstrTemp = "JustInTimeActivation";
        bTmp = TRUE;
        vTmp = bTmp;
        hr = pCatalogObjectCo->put_Value(bstrTemp, vTmp);
        if (!SUCCEEDED(hr)) goto Error;

        bstrTemp = "MaxPoolSize";
        vTmp.Clear(); // clear variant so it isn't stored as a
bool (_variant_t feature)
        vTmp = (long)30;
        hr = pCatalogObjectCo->put_Value(bstrTemp, vTmp);
        if (!SUCCEEDED(hr)) goto Error;

        bstrTemp = "ObjectPoolingEnabled";
        bTmp = TRUE;
        vTmp = bTmp;
        hr = pCatalogObjectCo->put_Value(bstrTemp, vTmp);
        if (!SUCCEEDED(hr)) goto Error;

        // save key to get the InterfacesForComponent collection
        hr = pCatalogObjectCo->get_Key(&vKey);
        if (!SUCCEEDED(hr)) goto Error;

        bstrTemp = "InterfacesForComponent";
        hr = pCatalogCollectionCo->GetCollection(bstrTemp, vKey,
(IDispatch**) &pCatalogCollectionItf);
        if (!SUCCEEDED(hr)) goto Error;

        hr = pCatalogCollectionItf->Populate();
        if (!SUCCEEDED(hr)) goto Error;

        hr = pCatalogCollectionItf->get_Count(&lCountItf);
        if (!SUCCEEDED(hr)) goto Error;

        // iterate through interfaces in component
        while (lCountItf > 0)
        {
            hr = pCatalogCollectionItf->get_Item(lCountItf - 1,
(IDispatch**) &pCatalogObjectItf);
            if (!SUCCEEDED(hr)) goto Error;

            // save key to get the MethodsForInterface collection
            hr = pCatalogObjectItf->get_Key(&vKey);
            if (!SUCCEEDED(hr)) goto Error;

            bstrTemp = "MethodsForInterface";
            hr = pCatalogCollectionItf->GetCollection(bstrTemp,
vKey, (IDispatch**) &pCatalogCollectionMethod);
            if (!SUCCEEDED(hr)) goto Error;

            hr = pCatalogCollectionMethod->Populate();
            if (!SUCCEEDED(hr)) goto Error;

            hr = pCatalogCollectionMethod-
>get_Count(&lCountMethod);
            if (!SUCCEEDED(hr)) goto Error;

            // iterate through methods of interface
            while (lCountMethod > 0)
            {

```

```

                hr = pCatalogCollectionMethod-
>get_Item(lCountMethod - 1, (IDispatch**) &pCatalogObjectMethod);
                if (!SUCCEEDED(hr)) goto Error;

                bstrTemp = "AutoComplete";
                bTmp = TRUE;
                vTmp = bTmp;
                hr = pCatalogObjectMethod-
>put_Value(bstrTemp, vTmp);
                if (!SUCCEEDED(hr)) goto Error;

                pCatalogObjectMethod->Release();
                pCatalogObjectMethod = NULL;

                lCountMethod--;
            }

            // save changes
            hr = pCatalogCollectionMethod->SaveChanges(&lActProp);
            if (!SUCCEEDED(hr)) goto Error;

            pCatalogObjectItf->Release();
            pCatalogObjectItf = NULL;

            lCountItf--;
        }

        pCatalogObjectCo->Release();
        pCatalogObjectCo = NULL;

        lCountCo--;
    }

    // save changes
    hr = pCatalogCollectionCo->SaveChanges(&lActProp);
    if (!SUCCEEDED(hr)) goto Error;

    pCatalogCollectionApp->Release();
    pCatalogCollectionApp = NULL;

    pCatalogCollectionCo->Release();
    pCatalogCollectionCo = NULL;

    pCatalogCollectionItf->Release();
    pCatalogCollectionItf = NULL;

    pCatalogCollectionMethod->Release();
    pCatalogCollectionMethod = NULL;

Error:
    CoUninitialize();

    if (!SUCCEEDED(hr))
    {
        LPTSTR lpBuf;
        DWORD dwRes = FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER |
FORMAT_MESSAGE_FROM_SYSTEM,
        NULL,

```

```

    hr,

    MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),

    (LPTSTR) &lpBuf,

    0,

    NULL);
//      _tprintf(_T("Error adding components. HRESULT: 0x%x\n%s"), hr,
lpBuf);
        return TRUE;
    }
    else
        return FALSE;
}

```

install_resource.h

```

//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by install.rc
//
#define IDD_DIALOG1          101
#define IDI_ICON1           102
#define IDR_TPCCDLL         103
#define IDD_DIALOG2        105
#define IDI_ICON2           106
#define IDR_DELIVERY        107
#define IDD_DIALOG3        108
#define IDR_LICENSE1       112
#define IDD_DIALOG4        113
#define IDR_TPCCOBJ1       117
#define IDR_TPCCSTUB1      118
#define IDR_DBLIB_DLL      122
#define IDR_ODBC_DLL       123
#define IDR_TUXEDO_APP     124
#define IDR_TUXEDO_DLL     125
#define IDR_COM_DLL        126
#define IDR_COMPS_DLL      127
#define IDR_COMALL_DLL     128
#define BN_LOG              1001
#define ED_KEEP             1002
#define ED_THREADS         1003
#define ED_THREADS2        1004
#define IDC_PATH           1007
#define IDC_VERSION        1009
#define IDC_RESULTS        1010
#define IDC_PROGRESS1      1011
#define IDC_STATUS         1012
#define IDC_BUTTON1        1013
#define ED_MAXCONNECTION   1014
#define ED_IIS_MAX_THREAD_POOL_LIMIT 1015
#define ED_MAXDELIVERIES   1016
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE 1017
#define ED_IIS_THREAD_TIMEOUT 1018
#define ED_IIS_LISTEN_BACKLOG 1019
#define IDC_DBLIB          1021
#define IDC_LICENSE        1022
#define IDC_ODBC           1022
#define IDC_CONNECT_POOL   1023

```

```

#define ED_DB_SERVER        1023
#define ED_USER_CONNECT_DELAY_TIME 1024
#define ED_DB_USER_ID      1024
#define IDC_MTS             1025
#define IDC_TM_MTS         1025
#define IDC_TM_TUXEDO      1026
#define IDC_TM_NONE        1027
#define ED_DB_PASSWORD     1028
#define ED_DB_NAME         1029
#define IDC_TM_ENCINA      1030

```

```

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifdef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        129
#define _APS_NEXT_COMMAND_VALUE        40001
#define _APS_NEXT_CONTROL_VALUE        1024
#define _APS_NEXT_SYMED_VALUE         101
#endif
#endif

```

isapi_dll.dsp

```

# Microsoft Developer Studio Project File - Name="isapi_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGETTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=isapi_dll - Win32 IceCAP
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "isapi_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "isapi_dll.mak" CFG="isapi_dll - Win32 IceCAP"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "isapi_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "isapi_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "isapi_dll - Win32 IceCAP" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF "$ (CFG)" == "isapi_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"

```



```

# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "NDEBUG" /D "WIN32" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /1 0x409 /d "NDEBUG"
# ADD RSC /1 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ..\common\txnlog\lib\release\rtetime.lib
..\common\txnlog\lib\release\spinlock.lib ..\common\txnlog\lib\release\error.lib
..\common\txnlog\lib\release\txnlog.lib wsock32.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbcc32.lib odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
/nodfaultlib:"LIBCMT" /out:".bin\tpcc.dll"
# SUBTRACT LINK32 /nodfaultlib

!ELSEIF "$(CFG)" == "isapi_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /GX /ZI /Od /D "_DEBUG" /D "WIN32" /D "_WINDOWS" /FR /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D " _DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /1 0x409 /d " _DEBUG"
# ADD RSC /1 0x409 /d " _DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdptype:sept
# ADD LINK32 ..\common\txnlog\lib\debug\rtetime.lib
..\common\txnlog\lib\debug\spinlock.lib ..\common\txnlog\lib\debug\error.lib
..\common\txnlog\lib\debug\txnlog.lib wsock32.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib
odbcc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386
/nodfaultlib:"LIBCMTD" /out:".bin\tpcc.dll" /pdptype:sept

```

```

# SUBTRACT LINK32 /profile /pdb:none /nodfaultlib

!ELSEIF "$(CFG)" == "isapi_dll - Win32 IceCAP"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "isapi_dll"
# PROP BASE Intermediate_Dir "isapi_dll"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /GX /ZI /Od /D "_DEBUG" /D "WIN32" /D "_WINDOWS" /FR
/YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /GX /ZI /Od /D "NDEBUG" /D "ICECAP" /D "WIN32" /D
"_WINDOWS" /FR /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D " _DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /1 0x409 /d " _DEBUG"
# ADD RSC /1 0x409 /d " _DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc.dll"
/pdptype:sept
# SUBTRACT BASE LINK32 /profile /pdb:none
# ADD LINK32 icap.lib ..\common\txnlog\lib\release\rtetime.lib
..\common\txnlog\lib\release\spinlock.lib ..\common\txnlog\lib\release\error.lib
..\common\txnlog\lib\release\txnlog.lib wsock32.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbcc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug
/machine:I386 /out:".bin\tpcc.dll" /pdptype:sept
# SUBTRACT LINK32 /profile /pdb:none /map

!ENDIF

# Begin Target

# Name "isapi_dll - Win32 Release"
# Name "isapi_dll - Win32 Debug"
# Name "isapi_dll - Win32 IceCAP"
# Begin Group "Source"

# PROP Default_Filter "*.cpp, *.def, *.rc"
# Begin Source File

SOURCE=.\src\tpcc.cpp
# End Source File
# Begin Source File

SOURCE=.\src\tpcc.def
# End Source File
# Begin Source File

SOURCE=.\src\tpcc.rc
# End Source File

```

```

# End Group
# Begin Group "Header Files"

# PROP Default_Filter "*.h, *.hpp"
# Begin Source File

SOURCE=..\common\src\error.h
# End Source File
# Begin Source File

SOURCE=..\common\src\ReadRegistry.h
# End Source File
# Begin Source File

SOURCE=..\src\tpcc.h
# End Source File
# Begin Source File

SOURCE=..\db_dblib_dll\src\tpcc_dblib.h
# End Source File
# Begin Source File

SOURCE=..\db_odbc_dll\src\tpcc_odbc.h
# End Source File
# Begin Source File

SOURCE=..\tm_tuxedo_dll\src\tpcc_tux.h
# End Source File
# Begin Source File

SOURCE=..\common\src\trans.h
# End Source File
# Begin Source File

SOURCE=..\common\src\txn_base.h
# End Source File
# End Group
# End Target
# End Project

```

isapi_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
#ifdef 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

```

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);
}

```

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.
*/

#ifdef _INC_Spinlock

const LONG LockClosed = 1;
const LONG LockOpen = 0;

/*****
* Spinlock and Semaphore locking.
*
* This class provides a very conservative locking scheme.
* The assumption behind the code is that locks will be
* held for a very short time. When a lock is taken a memory
* location is exchanged. All other threads that want this
* lock wait by spinning and sometimes sleeping on a semaphore
*****/

```

```

* until it becomes free again. The only other choice is not
* to wait at all and move on to do something else. This
* module should normally be used in conjunction with cache
* aligned memory in minimize cache line misses.
*
*****/
class Spinlock
{
    // Private data.
    HANDLE                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

```

tm_com_dll.dsp

```

# Microsoft Developer Studio Project File - Name="tm_com_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGETTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=tm_com_dll - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tm_com_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tm_com_dll.mak" CFG="tm_com_dll - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tm_com_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "tm_com_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""

```

```

# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF "$(CFG)" == "tm_com_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_ExporT_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386 /out:".bin\tpcc_com.dll"

!ELSEIF "$(CFG)" == "tm_com_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_ExporT_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/ED /c
# ADD BASE MTL /nologo /D "DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe

```

```

# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".bin\tpcc_com.dll"
/pdbtype:sept

!ENDIF

# Begin Target

# Name "tm_com_dll - Win32 Release"
# Name "tm_com_dll - Win32 Debug"
# Begin Source File

SOURCE=.\src\tpcc_com.cpp
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com.h
# End Source File
# End Target
# End Project

```

tpcc.cpp

```

/* FILE: TPCC.C
 *
 * Microsoft TPC-C Kit Ver. 4.20.000
 * Copyright Microsoft, 1999
 *
 * All Rights Reserved
 *
 * Version 4.10.000 audited by Richard Gimarc,
 * Performance Metrics, 3/17/99
 *
 * PURPOSE: Main module for TPCC.DLL which is an ISAPI service dll.
 * Contact: Charles Levine (clevine@microsoft.com)
 *
 * Change history:
 * 4.20.000 - reworked error handling; added options for COM and
 * Encina txn monitors
 */

#include <windows.h>
#include <process.h>
#include <tchar.h>
#include <stdio.h>
#include <stdarg.h>
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <io.h>
#include <assert.h>

#include <sqltypes.h>

#ifdef ICECAP
#include <icapexp.h>
#endif

```

```

#include "..\..\common\src\trans.h" //tpckit transaction header
contains definitions of structures specific to TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\ReadRegistry.h"

#include "..\..\common\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 ocured in initialization
* TRUE
DLL successfully initialized
*/

BOOL WINAPI DllMain(HANDLE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
    DWORD i;
    char szEvent[LEN_ERR_STRING] = "\0";
    char szLogFile[128];
    char szDllName[128];

    try
    {
        switch( ul_reason_for_call )
        {
            case DLL_PROCESS_ATTACH:
                {

```

```

DWORD dwSize =
MAX_COMPUTERNAME_LENGTH+1;
GetComputerName(szMyComputerName,
&dwSize);
    szMyComputerName[dwSize] = 0;
}
DisableThreadLibraryCalls((HMODULE)hModule);
InitializeCriticalSection(&TermCriticalSection);
    if ( ReadTPCCRegistrySettings( &Reg )
        throw new CWEBCLNT_ERR(
ERR_MISSING_REGISTRY_ENTRIES );
        dwDelBuffSize = min(
Reg.dwMaxPendingDeliveries, 10000 ); // min with 10000 as a sanity constraint
        dwNumDeliveryThreads = min(
Reg.dwNumberOfDeliveryThreads, 100 ); // min with 100 as a sanity constraint
TermInit();
// load DLL for txn monitor
if (Reg.eTxnMon == TUXEDO)
{
    strcpy( szDllName, Reg.szPath );
    strcat( szDllName,
"tpcc_tuxedo.dll");
    szDllName );
    hLibInstanceTm = LoadLibrary(
    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");
        szDllName );
        hLibInstanceTm = LoadLibrary(
        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");
    szDllName );
    hLibInstanceTm = LoadLibrary(
    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");
        LoadLibrary( szDllName );
        hLibInstanceDb =
        if (hLibInstanceDb ==
NULL)
            throw new
CWEBCLNT_ERR( ERR_LOADDLL_FAILED, szDllName, GetLastError() );
// get function pointer
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");
        LoadLibrary( szDllName );
        hLibInstanceDb =
        if (hLibInstanceDb ==
NULL)
            throw new
CWEBCLNT_ERR( ERR_LOADDLL_FAILED, szDllName, GetLastError() );
// get function pointer
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
                                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
                                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);
    lstrcpy(pVer->lpszExtensionDesc, "TPC-C Server.",
HSE_MAX_EXT_DLL_NAME_LEN);

    // TODO: why do we need this here instead of in the DLL attach?
    if (Reg.eTxnMon == ENCINA)
        pCTPCC_ENCINA_post_init();

    return TRUE;
}

/* FUNCTION: TerminateExtension
*
* PURPOSE:        This function is called by the inet service when the DLL is
about to be unloaded.                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 TPC DLL. The
internet service                    calls this function passing in the http string.
*
* ARGUMENTS:      EXTENSION_CONTROL_BLOCK  *pECB    structure pointer to
passed in internet
*
* 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
    int iRetryCnt = 0;
    static int iMaxRetries = 10;
    assert(txnDeliRec != NULL);
Reconnect:
    try
    {
        if (Reg.eDB_Protocol == ODBC)
            pTxn = pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );
        else if (Reg.eDB_Protocol == DBLIB)
            pTxn = pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );
        pDeliveryData = pTxn->BuffAddr_Delivery();
    }
    catch (CBaseErr *e)
    {
        char szTmp[1024];
        wsprintf( szTmp, "Error in Delivery Txn thread. Could not
connect to database. "
"%s. Server=%s, User=%s, Password=%s,
Database=%s",

```

```

        e->ErrorText(), Reg.szDbServer,
Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
        WriteMessageToEventLog( szTmp );
        delete e;
    }
    // will retry connection up to ten times
    if (iRetryCnt++ < iMaxRetries)
    {
        Sleep(5000); // delay for 5 seconds
        goto Reconnect;
    }
    wsprintf( szTmp, "Delivery Txn thread terminating after %d
retries.", iMaxRetries );
    WriteMessageToEventLog( szTmp );
    goto ErrorExit;
}
catch (...)
{
    WriteMessageToEventLog(TEXT("Unhandled exception caught in
DeliveryWorkerThread. Delivery Txn thread terminating."));
    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(short w_id, short o_carrier_id)
{
    BOOL bError;

    EnterCriticalSection(&DelBuffCriticalSection);
    if (dwDelBuffFreeCount > 0)
    {
        bError = FALSE;
        (pDelBuff+dwDelBuffFreeIndex)->w_id =
w_id;
        (pDelBuff+dwDelBuffFreeIndex)->o_carrier_id =
o_carrier_id;
        GetLocalTime(&(pDelBuff+dwDelBuffFreeIndex)->queue);

        dwDelBuffFreeCount--;
        dwDelBuffFreeIndex++;
        if (dwDelBuffFreeIndex == dwDelBuffSize)
            dwDelBuffFreeIndex = 0; // wrap-around

if at end of buffer
    }
    else
        // No free buffers. Return an error, which indicates that the
delivery buffer is full.
        // Most likely, the number of delivery worker threads needs to
be increased to keep up
        // with the txn rate.
        bError = TRUE;
    LeaveCriticalSection(&DelBuffCriticalSection);

    if (!bError)
        // increment worker semaphore to wake up a worker thread
        ReleaseSemaphore( hWorkerSemaphore, 1, NULL );

    return bError;
}

/* FUNCTION: ProcessQueryString
 *
 * PURPOSE:      This function extracts the relevent information out of the http
command passed in from
                the browser.
 *
 * COMMENTS:     If this is the initial connection i.e. client is at welcome
screen then
                there will not be a terminal id or current
form id. If this is the case
                then the pTermid and pFormid return values
are undefined.
 */

void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd, int *pFormId, int
*pTermId, int *pSyncId)
{
    char *ptr = pECB->lpszQueryString;
    char szBuffer[25];
    int i;

    //allowable client command strings i.e. CMD=command
    static char *szCmds[] =
    {

```

```

        "Process", "..NewOrder..", "..Payment..", "..Delivery..",
"..Order-Status..", "..Stock-Level..",
        );
        "..Exit..", "Submit", "Menu", "Clear", "Stats", ""
    };

    *pCmd      = 0;          // default is the login screen
    *pTermId = 0;

    // if no params (i.e., empty query string), then return login screen
    if (strlen(pECB->lpszQueryString) == 0)
        return;

    // parse FORMID, TERMIID, and SYNCID
    *pFormId = GetIntKeyValue(&ptr, "FORMID", NO_ERR, NO_ERR);
    *pTermId = GetIntKeyValue(&ptr, "TERMIID", NO_ERR, NO_ERR);
    *pSyncId = GetIntKeyValue(&ptr, "SYNCID", NO_ERR, NO_ERR);

    // parse CMD
    GetKeyValue(&ptr, "CMD", szBuffer, sizeof(szBuffer),
ERR_COMMAND_UNDEFINED);

    // see which command it matches
    for(i=0; ; i++)
    {
        if (szCmds[i][0] == 0)
            // no more; no match; return error
            throw new CWEBCLNT_ERR( ERR_COMMAND_UNDEFINED );
        if ( !strcmp(szCmds[i], szBuffer) )
        {
            *pCmd = i+1;
            break;
        }
    }
}

/* FUNCTION: void WelcomeForm
 *
 */

void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer)
{
    char szTmp[1024];

    //welcome to tpc-c html form buffer, this is first form client sees.
    strcpy( szBuffer, "<HTML><HEAD><TITLE>TPC-C Web
Client</TITLE></HEAD><BODY>"

        "<B><BIG>Microsoft TPC-C
Web Client (ver 4.20)</BIG></B> <BR> <BR>"

        "New\ "<PRE>"

        "__TIME__" <BR>"

        ("__TIMESTAMP__") <BR>"

        ACTION="\"tpcc.dll\" METHOD="\"GET\">"

        NAME="\"STATUSID\" VALUE="\"0\">"

        NAME="\"ERROR\" VALUE="\"0\">"

        NAME="\"FORMID\" VALUE="\"1\">"

```

```

        NAME="\"TERMIID\" VALUE="\"0\">"          "<INPUT TYPE="\"hidden\"
        NAME="\"SYNCID\" VALUE="\"0\">"          "<INPUT TYPE="\"hidden\"
        NAME="\"VERSION\" VALUE="\"\" WEBCLIENT_VERSION \"\">"      "<INPUT TYPE="\"hidden\"
        );

        sprintf( szTmp, "Configuration Settings: <BR><font face="\"Courier
New\" color="\"blue\"><PRE>"

            "Txn Monitor          =
            <B>%s</B><BR>"          "Database protocol          =
            <B>%s</B><BR>"          "Max Connections          =
            <B>%d</B><BR>"          "# of Delivery Threads     =
            <B>%d</B><BR>"          "Max Pending Deliveries   =
            <B>%d</B><BR>"

            , szTxnMonNames[Reg.eTxnMon],
            szDBNames[Reg.eDB_Protocol],
            Reg.dwMaxConnections, dwNumDeliveryThreads,
            dwDelBuffSize );
        strcat( szBuffer, szTmp);

        if (Reg.eTxnMon == COM)
        {
            sprintf( szTmp, "COM Single Pool          = <B>%s</B><BR>",
                Reg.bCOM_SinglePool ? "YES" : "NO" );
            strcat( szBuffer, szTmp);
        }
        strcat( szBuffer, "</PRE></font>");

        if (Reg.eTxnMon == None)
            // connection options may be specified when not using a txn
            monitor

            sprintf( szTmp, "Please enter your database options for this
connection:<BR>"

                "<font face="\"Courier
New\" color="\"blue\"><PRE>"

                "DB Server          = <INPUT
NAME="\"db_server\" SIZE=20 VALUE="\"%s\"><BR>"          "DB User ID          = <INPUT
NAME="\"db_user\" SIZE=20 VALUE="\"%s\"><BR>"          "DB Password          = <INPUT
NAME="\"db_passwd\" SIZE=20 VALUE="\"%s\"><BR>"          "DB Name          = <INPUT
NAME="\"db_name\" SIZE=20 VALUE="\"%s\"><BR>"

                "</PRE></font>"

                , Reg.szDbServer, Reg.szDbUser,
                Reg.szDbPassword, Reg.szDbName );
            else
                // if using a txn monitor, connection options are determined
                from registry; can't
                // set per user. show options fyi
                sprintf( szTmp, "Database options which will be used by the
transaction monitor:<BR>"

                    "<font face="\"Courier
New\" color="\"blue\"><PRE>"

                    "DB Server
                    = <B>%s</B><BR>"

```

```

                                "DB User ID
= <B>%s</B><BR>"                                "DB Password
= <B>%s</B><BR>"                                "DB Name
= <B>%s</B><BR>"                                "</PRE></font>"
                                                , Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, Reg.szDbName );
    strcat( szBuffer, szTmp);

    sprintf( szTmp, "Please enter your Warehouse and District for this
session:<BR>"                                "<font face=\"Courier New\""
color=\"blue\"><PRE>" );
    strcat( szBuffer, szTmp);
    strcat( szBuffer, "Warehouse ID = <INPUT NAME=\"w_id\" SIZE=4><BR>"
                                                "District ID = <INPUT
NAME=\"d_id\" SIZE=2><BR>"
                                                "</PRE></font><HR>"
                                                "<INPUT TYPE=\"submit\""
NAME=\"CMD\" VALUE=\"Submit\">"
    "</FORM></BODY></HTML>");
}

/* FUNCTION: SubmitCmd
 *
 * PURPOSE:      This function allocated a new terminal id in the Term structure
array.
 *
 */

void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer)
{
    int            iNewTerm;
    char          *ptr = pECB->lpszQueryString;

    char          szVersion[32]    = { 0 };
    char          szServer[32]     = { 0 };
    char          szUser[32]       = "sa";
    char          szPassword[32]   = { 0 };
    char          szDatabase[32]   = "tpcc";

    // validate version field; the version field ensures that the RTE is
synchronized with the web client
    GetKeyValue(&ptr, "VERSION", szVersion, sizeof(szVersion),
ERR_VERSION_MISMATCH);
    if ( strcmp( szVersion, WEBCLIENT_VERSION ) )
        throw new CWEBCLNT_ERR( ERR_VERSION_MISMATCH );

    if (Reg.eTxnMon == None)
    {
        // parse Server name
        GetKeyValue(&ptr, "db_server", szServer, sizeof(szServer),
ERR_NO_SERVER_SPECIFIED);
        // parse User name
        GetKeyValue(&ptr, "db_user", szUser, sizeof(szUser), NO_ERR);
        // parse Password
        GetKeyValue(&ptr, "db_passwd", szPassword, sizeof(szPassword),
NO_ERR);
        // parse Database name

```

```

                                GetKeyValue(&ptr, "db_name", szDatabase, sizeof(szDatabase),
NO_ERR);
    }
    // parse warehouse ID
    int w_id = GetIntKeyValue(&ptr, "w_id", ERR_HTML_ILL_FORMED,
ERR_W_ID_INVALID);
    if ( w_id < 1 )
        throw new CWEBCLNT_ERR( ERR_W_ID_INVALID );

    // parse district ID
    int d_id = GetIntKeyValue(&ptr, "d_id", ERR_HTML_ILL_FORMED,
ERR_D_ID_INVALID);
    if ( d_id < 1 || d_id > 10 )
        throw new CWEBCLNT_ERR( ERR_D_ID_INVALID );

    iNewTerm = TermAdd();

    Term.pClientData[iNewTerm].w_id = w_id;
    Term.pClientData[iNewTerm].d_id = d_id;

    try
    {
        if (Reg.eTxnMon == TUXEDO)
            Term.pClientData[iNewTerm].pTxn = pCTPCC_TUXEDO_new();
        else if (Reg.eTxnMon == ENCINA)
            Term.pClientData[iNewTerm].pTxn = pCTPCC_ENCINA_new();
        else if (Reg.eTxnMon == COM)
            Term.pClientData[iNewTerm].pTxn = pCTPCC_COM_new(
Reg.bCOM_SinglePool );
        else if (Reg.eDB_Protocol == ODBC)
            Term.pClientData[iNewTerm].pTxn = pCTPCC_ODBC_new(
szServer, szUser, szPassword, szMyComputerName, szDatabase );
        else if (Reg.eDB_Protocol == DBLIB)
            Term.pClientData[iNewTerm].pTxn = pCTPCC_DBLIB_new(
szServer, szUser, szPassword, szMyComputerName, szDatabase );
    }
    catch (...)
    {
        TermDelete(iNewTerm);
        throw; // pass exception upward
    }

    MakeMainMenuForm(iNewTerm, Term.pClientData[iNewTerm].iSyncId, szBuffer);
}

/* FUNCTION: StatsCmd
 *
 * PURPOSE:      This function returns to the browser the total number of active
terminal ids.
 *
 * This routine is for development/debugging purposes.
 *
 */

void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer)
{
    int i;
    int iTotals;

    EnterCriticalSection(&TermCriticalSection);

    iTotals = 0;

```

```

for(i=0; i<Term.iNumEntries; i++)
{
    if (Term.pClientData[i].iNextFree == -1)
        iTot++;
}

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>"
, iTot );

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_TOO_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,
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,
"sync.\" },
        {
            ERR_W_ID_INVALID,
"Invalid Warehouse ID."
        },
        {
            0,
            ""
        }
    }
}

```

```

};
char szTmp[256];
int i = 0;
while (TRUE)
{
    if (errorMsgs[i].szMsg[0] == 0)
    {
        strcpy( szTmp, "Unknown error number." );
        break;
    }
    if (m_Error == errorMsgs[i].iError)
    {
        strcpy( szTmp, errorMsgs[i].szMsg );
        break;
    }
    i++;
}

if (m_szTextDetail)
    strcat( szTmp, m_szTextDetail );
if (m_SystemErr)
    sprintf( szTmp+strlen(szTmp), " Error=%d", m_SystemErr );

m_szErrorText = new char[strlen(szTmp)+1];
strcpy( m_szErrorText, szTmp );
return m_szErrorText;
}

/* FUNCTION: GetKeyValue
*
* PURPOSE:          This function parses a http formatted string for specific key
values.
*
* ARGUMENTS:       char          *pQueryString      http string
from client browser
                  char          *pKey
                  key value to look for
                  char          *pValue
character array into which to place key's value
                  int          iMax
maximum length of key value array.
                  WEBERROR     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".

```



```

never // This is intentional, as the zero entry is used as an anchor and
// 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=\"TERMINID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
        "<BOLD>An Error Occurred</BOLD><BR><BR>"
        "%s"
        "<BR><BR><HR>"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Payment..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Delivery..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-
Status..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Exit..\">"
        "</FORM></BODY></HTML>"
        , iType, iErrorNum, MAIN_MENU_FORM, iTermId, 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=\"TERMINID\" 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=\"TERMINID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
        "<PRE><font face=\"Courier\">"
        "Stock-Level<BR>"
        "Warehouse: %4.4d District: %2.2d<BR> <BR>",
        STOCK_LEVEL_FORM, iTermId, Term.pClientData[iTermId].iSyncId,
        Term.pClientData[iTermId].w_id, Term.pClientData[iTermId].d_id);

    if ( bInput )
    {
        strcpy(szForm+c,
            "Stock Level Threshold: <INPUT NAME=\"TT*\"
SIZE=2><BR> <BR>"
            "low stock: </font><BR> <BR> <BR> <BR> <BR> <BR>"
            "<BR> <BR> <BR> <BR>"
            " <BR> <BR> <BR> <BR> <BR> <BR> <BR></PRE><HR>"
            "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Process\">"
            "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
            "</FORM></HTML>" );
    }
    else
    {
        wsprintf(szForm+c,
            "Stock Level Threshold: %2.2d<BR> <BR>"
            "low stock: %3.3d</font> <BR> <BR> <BR> <BR> <BR> <BR>"
            "<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR></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> <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=\"TERMINID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
        "<PRE><font face=\"Courier\">"
        "New Order<BR>"
        "      , bValid ? 0 : ERR_BAD_ITEM_ID, NEW_ORDER_FORM, iTermId,
        Term.pClientData[iTermId].iSyncId);

    if ( bInput )
    {
        c += wsprintf(szForm+c, "Warehouse: %4.4d ",
        Term.pClientData[iTermId].w_id );

        strcpy( szForm+c,
            "District: <INPUT NAME=\"DID\" SIZE=1>"
            "Customer: <INPUT NAME=\"CID\" SIZE=4> Name:
            Credit: %8.4d<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
            <INPUT NAME=\"Qty00*\"
            NAME=\"IID00*\" SIZE=6>
            SIZE=1><BR>"
            " <INPUT NAME=\"SP01*\" SIZE=4> <INPUT
            <INPUT NAME=\"Qty01*\"
            NAME=\"IID01*\" SIZE=6>
            SIZE=1><BR>"
            " <INPUT NAME=\"SP02*\" SIZE=4> <INPUT
            <INPUT NAME=\"Qty02*\"
            NAME=\"IID02*\" SIZE=6>
            SIZE=1><BR>"
            " <INPUT NAME=\"SP03*\" SIZE=4> <INPUT
            <INPUT NAME=\"Qty03*\"
            NAME=\"IID03*\" SIZE=6>
            SIZE=1><BR>"
            " <INPUT NAME=\"SP04*\" SIZE=4> <INPUT
            <INPUT NAME=\"Qty04*\"
            NAME=\"IID04*\" SIZE=6>
            SIZE=1><BR>"

```

```

        " <INPUT NAME=\"SP05*\" SIZE=4> <INPUT
        <INPUT NAME=\"Qty05*\"
        NAME=\"IID05*\" SIZE=6>
        SIZE=1><BR>"
        " <INPUT NAME=\"SP06*\" SIZE=4> <INPUT
        <INPUT NAME=\"Qty06*\"
        NAME=\"IID06*\" SIZE=6>
        SIZE=1><BR>"
        " <INPUT NAME=\"SP07*\" SIZE=4> <INPUT
        <INPUT NAME=\"Qty07*\"
        NAME=\"IID07*\" SIZE=6>
        SIZE=1><BR>"
        " <INPUT NAME=\"SP08*\" SIZE=4> <INPUT
        <INPUT NAME=\"Qty08*\"
        NAME=\"IID08*\" SIZE=6>
        SIZE=1><BR>"
        " <INPUT NAME=\"SP09*\" SIZE=4> <INPUT
        <INPUT NAME=\"Qty09*\"
        NAME=\"IID09*\" SIZE=6>
        SIZE=1><BR>"
        " <INPUT NAME=\"SP10*\" SIZE=4> <INPUT
        <INPUT NAME=\"Qty10*\"
        NAME=\"IID10*\" SIZE=6>
        SIZE=1><BR>"
        " <INPUT NAME=\"SP11*\" SIZE=4> <INPUT
        <INPUT NAME=\"Qty11*\"
        NAME=\"IID11*\" SIZE=6>
        SIZE=1><BR>"
        " <INPUT NAME=\"SP12*\" SIZE=4> <INPUT
        <INPUT NAME=\"Qty12*\"
        NAME=\"IID12*\" SIZE=6>
        SIZE=1><BR>"
        " <INPUT NAME=\"SP13*\" SIZE=4> <INPUT
        <INPUT NAME=\"Qty13*\"
        NAME=\"IID13*\" SIZE=6>
        SIZE=1><BR>"
        " <INPUT NAME=\"SP14*\" SIZE=4> <INPUT
        <INPUT NAME=\"Qty14*\"
        "Execution Status:
        Total:<BR>"
        "</font></PRE><HR>"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
        VALUE=\"Process\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
        "</FORM></HTML>"
    );
    }
    else
    {
        c += wsprintf(szForm+c, "Warehouse: %4.4d District: %2.2d
        Date: ",
        pNewOrderData->w_id,
        pNewOrderData->d_id);

        if ( bValid )
        {
            c += wsprintf(szForm+c, "%2.2d-%2.2d-%4.4d
            %2.2d:%2.2d:%2.2d",
            pNewOrderData->o_entry_d.day,
            pNewOrderData->o_entry_d.month,
            pNewOrderData->o_entry_d.year,
            pNewOrderData->o_entry_d.hour,
            pNewOrderData->o_entry_d.minute,
            pNewOrderData->o_entry_d.second);
        }

        c += wsprintf(szForm+c, "<BR>Customer: %4.4d Name: %16s
        Credit: %2s ",
        pNewOrderData->c_id, pNewOrderData->c_last,
        pNewOrderData->c_credit);

```

```

        if ( bValid )
        {
            c += sprintf(szForm+c,
                "%%Disc: %5.2f
                "Order Number: %8.8d
                " Supp_W Item_Id Item
                Name Qty Stock B/G Price Amount<BR>",
                100.0*pNewOrderData->c_discount,
                pNewOrderData->o_id,
                pNewOrderData->o_ol_cnt,
                100.0 * pNewOrderData->w_tax,
                100.0 * pNewOrderData->d_tax);

            for(i=0; i<pNewOrderData->o_ol_cnt; i++)
            {
                c += sprintf(szForm+c, " %4.4d %6.6d %-
                24s %2.2d %3.3d %1.1s %$6.2f %$7.2f <BR>",
                pNewOrderData->
                >OL[i].ol_supply_w_id,
                pNewOrderData->OL[i].ol_i_id,
                pNewOrderData->OL[i].ol_i_name,
                pNewOrderData->OL[i].ol_quantity,
                pNewOrderData->OL[i].ol_stock,
                pNewOrderData->OL[i].ol_i_price,
                pNewOrderData->OL[i].ol_amount );
            }
        }
        else
        {
            c += wsprintf(szForm+c,
                "%%Disc:<BR>"
                "Order Number: %8.8d Number of Lines:
                " 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
            committed. Total: %$8.2f ",
            pNewOrderData->total_amount);
        }
        else
        c += wsprintf(szForm+c, "Execution Status: Item number
        is not valid. Total:");

        strcpy(szForm+c,
            " <BR></font></PRE><HR>"
            "<INPUT TYPE=\"submit\" NAME=\"CMD\"
            VALUE=\"..NewOrder..\">"
            "<INPUT TYPE=\"submit\" NAME=\"CMD\"
            VALUE=\"..Payment..\">"
            "<INPUT TYPE=\"submit\" NAME=\"CMD\"
            VALUE=\"..Delivery..\">"

```

```

        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-
        Status..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-
        Level..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
        VALUE=\"..Exit..\">"
        "</FORM></HTML>"
    );
}

/* FUNCTION: MakePaymentForm
 *
 * COMMENTS: The internal client buffer is created when the terminal id is
 * assigned and should not be freed except when the client terminal id
 * is no longer needed.
 */

void MakePaymentForm(int iTermId, PAYMENT_DATA *pPaymentData, BOOL bInput, char
*szForm)
{
    int c;

    c = wsprintf(szForm,
        "<HTML><HEAD><TITLE>TPC-C Payment</TITLE></HEAD><BODY>"
        "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
        "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"
        "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
        "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"TERMINID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
        "<PRE><font face=\"Courier\">"
        Payment<BR>"
        "Date: "
        , PAYMENT_FORM, iTermId, Term.pClientData[iTermId].iSyncId);

    if ( !bInput )
    {
        c += wsprintf(szForm+c, "%2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d",
            pPaymentData->h_date.day,
            pPaymentData->h_date.month,
            pPaymentData->h_date.year,
            pPaymentData->h_date.hour,
            pPaymentData->h_date.minute,
            pPaymentData->h_date.second);
    }

    if ( bInput )
    {
        c += wsprintf(szForm+c,
            "<BR> <BR>Warehouse: %4.4d"
            " District: <INPUT
            NAME=\"DID*\" SIZE=1><BR> <BR> <BR> <BR>"
            "Customer: <INPUT NAME=\"CID*\" SIZE=4>"
            "Cust-Warehouse: <INPUT NAME=\"CWI*\" SIZE=4> "
            "Cust-District: <INPUT NAME=\"CDI*\" SIZE=1><BR>"
            "Name: <INPUT NAME=\"CLT*\"
            SIZE=16>
            Since:<BR>"
            "
            Credit:<BR>"
            "
            Disc:<BR>"

```



```

"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 += sprintf(szForm+c,
        "District: %2.2d<BR>"
        "Customer: %4.4d  Name: %-16s %-2s %-16s<BR>",
        pOrderStatusData->d_id, pOrderStatusData->c_id,
        pOrderStatusData->c_first, pOrderStatusData->c_middle,
        pOrderStatusData->c_last);

    c += sprintf(szForm+c, "Cust-Balance: %9.2f<BR> <BR>",
        pOrderStatusData->c_balance);

    c += sprintf(szForm+c,
        "Order-Number: %8.8d  Entry-Date: %2.2d-%2.2d-%4.4d
%2.2d:%2.2d:%2.2d  Carrier-Number: %2.2d<BR>"
        "Supply-W      Item-Id  Qty      Amount      Delivery-
Date<BR>",
        pOrderStatusData->o_id,
        pOrderStatusData->o_entry_d.day,
        pOrderStatusData->o_entry_d.month,
        pOrderStatusData->o_entry_d.year,
        pOrderStatusData->o_entry_d.hour,
        pOrderStatusData->o_entry_d.minute,
        pOrderStatusData->o_entry_d.second,
        pOrderStatusData->o_carrier_id);

    for(i=0; i< pOrderStatusData->o_ol_cnt; i++)
    {
        c += sprintf(szForm+c, " %4.4d      %6.6d      %2.2d
%8.2f      %2.2d-%2.2d-%4.4d<BR>",
            pOrderStatusData->OL[i].ol_supply_w_id,
            pOrderStatusData->OL[i].ol_i_id,
            pOrderStatusData->OL[i].ol_quantity,
            pOrderStatusData->OL[i].ol_amount,
            pOrderStatusData->OL[i].ol_delivery_d.day,
            pOrderStatusData->OL[i].ol_delivery_d.month,
            pOrderStatusData->OL[i].ol_delivery_d.year);
    }

    strncpy( szForm+c, szBR, (15-i)*5 );
    c += (15-i)*5;

    strcpy(szForm+c,
        "</font></PRE><HR><INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"

```

```

"<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-
Status..\">"
"<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-
Level..\">"
"<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Exit..\">"
" </BODY></FORM></HTML>" );
}
}

/* FUNCTION: MakeDeliveryForm
*
* COMMENTS:      The internal client buffer is created when the terminal id is
*                 assigned and should not
*                 be freed except when the client terminal id
*                 is no longer needed.
*/

void MakeDeliveryForm(int iTermId, DELIVERY_DATA *pDeliveryData, BOOL bInput, char
*szForm)
{
    int      c;

    c = sprintf(szForm,
        "<HTML><HEAD><TITLE>TPC-C Delivery</TITLE></HEAD><BODY>"
        "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
        "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
        "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"TERMINID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"SYCID\" VALUE=\"%d\">"
        "<PRE><font face=\"Courier\">"

    Delivery<BR>"

        "Warehouse: %4.4d<BR> <BR>",
        (!bInput && (pDeliveryData->exec_status_code != eOK)) ?
    ERR_TYPE_DELIVERY_POST : 0,
        DELIVERY_FORM, iTermId, Term.pClientData[iTermId].iSyncId,
    Term.pClientData[iTermId].w_id);

    if ( bInput )
    {
        strcpy( szForm+c,
            "Carrier Number: <INPUT NAME=\"OCD*\" SIZE=1><BR>
<BR>"
            "Execution Status: <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR>"
            " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
</font></PRE><HR>"
            "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Process\">"
            "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
            "</BODY></FORM></HTML>" );
    }
    else
    {
        sprintf( 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..\">"

```

```

VALUE=\\..Payment..\\>"
" <INPUT TYPE=\\submit\\ NAME=\\CMD\\"
" <INPUT TYPE=\\submit\\ NAME=\\CMD\\"
VALUE=\\..Delivery..\\>"
" <INPUT TYPE=\\submit\\ NAME=\\CMD\\" VALUE=\\..Order-
Status..\\>"
" <INPUT TYPE=\\submit\\ NAME=\\CMD\\" VALUE=\\..Stock-
Level..\\>"
" <INPUT TYPE=\\submit\\ NAME=\\CMD\\"
VALUE=\\..Exit..\\>"
" </BODY></FORM></HTML>"
, pDeliveryData->o_carrier_id,
(pDeliveryData->exec_status_code == eOK) ? "Delivery
has been queued." : "Delivery
Post Failed "
);
}
/* FUNCTION: ProcessNewOrderForm
*
* PURPOSE: This function gets and validates the input data from the new
order form
* filling in the required input variables. it then calls
the SQLNewOrder
* transaction, constructs the output form and writes it
back to client
* browser.
*/
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char
*szBuffer)
{
    PNEW_ORDER_DATA pNewOrder;
    pNewOrder = Term.pClientData[iTermId].pTxn->BuffAddr_NewOrder();
    ZeroMemory(pNewOrder, sizeof(NEW_ORDER_DATA));
    pNewOrder->w_id = Term.pClientData[iTermId].w_id;
    GetNewOrderData(pECB->lpszQueryString, pNewOrder);
    Term.pClientData[iTermId].pTxn->NewOrder();
    pNewOrder = Term.pClientData[iTermId].pTxn->BuffAddr_NewOrder();
    MakeNewOrderForm(iTermId, pNewOrder, OUTPUT_FORM, szBuffer);
}
/* FUNCTION: void ProcessPaymentForm
*
* PURPOSE: This function gets and validates the input data from the payment
form
* filling in the required input variables. It then calls
the SQLPayment
* transaction, constructs the output form and writes it
back to client
* browser.
*
* ARGUMENTS: EXTENSION_CONTROL_BLOCK *pECB passed in structure
pointer from inetsrv.
* int
* iTermId client browser terminal id
*/

```

```

void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char *szBuffer)
{
    PPAYMENT_DATA pPayment;
    pPayment = Term.pClientData[iTermId].pTxn->BuffAddr_Payment();
    ZeroMemory(pPayment, sizeof(PAYMENT_DATA));
    pPayment->w_id = Term.pClientData[iTermId].w_id;
    GetPaymentData(pECB->lpszQueryString, pPayment);
    Term.pClientData[iTermId].pTxn->Payment();
    pPayment = Term.pClientData[iTermId].pTxn->BuffAddr_Payment();
    MakePaymentForm(iTermId, pPayment, OUTPUT_FORM, szBuffer);
}
/* FUNCTION: ProcessOrderStatusForm
*
* PURPOSE: This function gets and validates the input data from the Order
Status
* form filling in the required input variables. It then
calls the
* SQLOrderStatus transaction, constructs the output form
and writes it
* back to client browser.
*
* ARGUMENTS: EXTENSION_CONTROL_BLOCK *pECB passed in structure
pointer from inetsrv.
* int
* iTermId client browser terminal id
*/
void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char
*szBuffer)
{
    PORDER_STATUS_DATA pOrderStatus;
    pOrderStatus = Term.pClientData[iTermId].pTxn->BuffAddr_OrderStatus();
    ZeroMemory(pOrderStatus, sizeof(ORDER_STATUS_DATA));
    pOrderStatus->w_id = Term.pClientData[iTermId].w_id;
    GetOrderStatusData(pECB->lpszQueryString, pOrderStatus);
    Term.pClientData[iTermId].pTxn->OrderStatus();
    pOrderStatus = Term.pClientData[iTermId].pTxn->BuffAddr_OrderStatus();
    MakeOrderStatusForm(iTermId, pOrderStatus, OUTPUT_FORM, szBuffer);
}
/* FUNCTION: ProcessDeliveryForm
*
* PURPOSE: This function gets and validates the input data from the
delivery form
* filling in the required input variables. It then calls
the PostDeliveryInfo
* Api, The client is then informed that the transaction
has been posted.
*
* ARGUMENTS: EXTENSION_CONTROL_BLOCK *pECB passed in structure
pointer from inetsrv.
* int
* iTermId client browser terminal id
*/

```

```

*/
void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char *szBuffer)
{
    char *ptr = pECB->lpszQueryString;

    PDELIVERY_DATA pDelivery;

    pDelivery = Term.pClientData[iTermId].pTxn->BuffAddr_Delivery();
    ZeroMemory(pDelivery, sizeof(DELIVERY_DATA));
    pDelivery->w_id = Term.pClientData[iTermId].w_id;

    pDelivery->o_carrier_id = GetIntKeyValue(&ptr, "OCD*",
ERR_DELIVERY_MISSING_OCD_KEY, ERR_DELIVERY_CARRIER_INVALID);
    if ( pDelivery->o_carrier_id > 10 || pDelivery->o_carrier_id < 1 )
        throw new CWEBCLNT_ERR( ERR_DELIVERY_CARRIER_ID_RANGE );

    if (dwNumDeliveryThreads)
    {
        //post delivery info
        if ( PostDeliveryInfo(pDelivery->w_id, pDelivery->o_carrier_id)

            pDelivery->exec_status_code = eDeliveryFailed;
        else
            pDelivery->exec_status_code = eOK;
    }
    else // delivery is done synchronously if no delivery threads configured
        Term.pClientData[iTermId].pTxn->Delivery();

    pDelivery = Term.pClientData[iTermId].pTxn->BuffAddr_Delivery();
    MakeDeliveryForm(iTermId, pDelivery, OUTPUT_FORM, szBuffer);
}

/* FUNCTION: ProcessStockLevelForm
*
* PURPOSE: This function gets and validates the input data from the Stock
Level
* form filling in the required input variables. It then
* calls the
SQLStockLevel transaction, constructs the output form
* and writes it
back to client browser.
*
* ARGUMENTS: EXTENSION_CONTROL_BLOCK *pECB passed in structure
pointer from inetsrv.
* int
* iTermId client browser terminal id
*
*/

void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char
*szBuffer)
{
    char *ptr = pECB->lpszQueryString;

    PSTOCK_LEVEL_DATA pStockLevel;

    pStockLevel = Term.pClientData[iTermId].pTxn->BuffAddr_StockLevel();
    ZeroMemory( pStockLevel, sizeof(STOCK_LEVEL_DATA) );

    pStockLevel->w_id = Term.pClientData[iTermId].w_id;
    pStockLevel->d_id = Term.pClientData[iTermId].d_id;

```

```

pStockLevel->threshold = GetIntKeyValue(&ptr, "TT*",
ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY, ERR_STOCKLEVEL_THRESHOLD_INVALID);
    if ( pStockLevel->threshold >= 100 || pStockLevel->threshold < 0 )
        throw new CWEBCLNT_ERR( ERR_STOCKLEVEL_THRESHOLD_RANGE );

    Term.pClientData[iTermId].pTxn->StockLevel();

    pStockLevel = Term.pClientData[iTermId].pTxn->BuffAddr_StockLevel();
    MakeStockLevelForm(iTermId, pStockLevel, OUTPUT_FORM, szBuffer);
}

/* FUNCTION: GetNewOrderData
*
* PURPOSE: This function extracts and validates the new order form data
from an http command string.
*
* ARGUMENTS: LPSTR lpszQueryString
client browser http command string NEW_ORDER_DATA *pNewOrderData
pointer to new order data structure
*
*/

void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA *pNewOrderData)
{
    char szTmp[26];
    int i;
    short items;
    int ol_i_id, ol_quantity;
    char *ptr = lpszQueryString;

    static char szSP[MAX_OL_NEW_ORDER_ITEMS][6] =
    { "SP00*", "SP01*", "SP02*", "SP03*", "SP04*",
      "SP05*", "SP06*", "SP07*", "SP08*", "SP09*",
      "SP10*", "SP11*", "SP12*", "SP13*", "SP14*" };
    static char szIID[MAX_OL_NEW_ORDER_ITEMS][7] =
    { "IID00*", "IID01*", "IID02*", "IID03*", "IID04*",
      "IID05*", "IID06*", "IID07*", "IID08*", "IID09*",
      "IID10*", "IID11*", "IID12*", "IID13*", "IID14*" };
    static char szQty[MAX_OL_NEW_ORDER_ITEMS][7] =
    { "Qty00*", "Qty01*", "Qty02*", "Qty03*", "Qty04*",
      "Qty05*", "Qty06*", "Qty07*", "Qty08*", "Qty09*",
      "Qty10*", "Qty11*", "Qty12*", "Qty13*", "Qty14*" };

    pNewOrderData->d_id = GetIntKeyValue(&ptr, "DID*",
ERR_NEWORDER_FORM_MISSING_DID, ERR_NEWORDER_DISTRICT_INVALID);
    pNewOrderData->c_id = GetIntKeyValue(&ptr, "CID*",
ERR_NEWORDER_CUSTOMER_KEY, ERR_NEWORDER_CUSTOMER_INVALID);

    for(i=0, items=0; i<MAX_OL_NEW_ORDER_ITEMS; i++)
    {
        GetKeyValue(&ptr, szSP[i], szTmp, sizeof(szTmp),
ERR_NEWORDER_MISSING_SUPPW_KEY);
        if ( szTmp[0] )
        {
            if ( !IsNumeric(szTmp) )
                throw new CWEBCLNT_ERR(
ERR_NEWORDER_SUPPW_INVALID );
            pNewOrderData->OL[items].ol_supply_w_id =
(short)atoi(szTmp);

            ol_i_id = pNewOrderData->OL[items].ol_i_id =

```



```

                GetIntKeyValue(&ptr, szIID[i],
ERR_NEWORDER_MISSING_IID_KEY, ERR_NEWORDER_ITEMID_INVALID);
                if ( ol_i_id > 999999 || ol_i_id < 1 )
                    throw new CWEBCLNT_ERR(
ERR_NEWORDER_ITEMID_RANGE );

                ol_quantity = pNewOrderData->OL[items].ol_quantity =
                GetIntKeyValue(&ptr, szQty[i],
ERR_NEWORDER_MISSING_QTY_KEY, ERR_NEWORDER_QTY_INVALID);
                if ( ol_quantity > 99 || ol_quantity < 1 )
                    throw new CWEBCLNT_ERR(
ERR_NEWORDER_QTY_RANGE );

                items++;
            }
            else
            {
                // nothing entered for supply warehouse, so item id
and qty must also be blank
                GetKeyValue(&ptr, szIID[i], szTmp, sizeof(szTmp),
ERR_NEWORDER_MISSING_IID_KEY);
                if ( szTmp[0] )
                    throw new CWEBCLNT_ERR(
ERR_NEWORDER_ITEMID_WITHOUT_SUPPW );

                GetKeyValue(&ptr, szQty[i], szTmp, sizeof(szTmp),
ERR_NEWORDER_MISSING_QTY_KEY);
                if ( szTmp[0] )
                    throw new CWEBCLNT_ERR(
ERR_NEWORDER_QTY_WITHOUT_SUPPW );
            }
        }
        if ( items == 0 )
            throw new CWEBCLNT_ERR( ERR_NEWORDER_NOITEMS_ENTERED );

        pNewOrderData->o_ol_cnt = items;
}

/* FUNCTION: GetPaymentData
 *
 * PURPOSE:      This function extracts and validates the payment form data from
an http command string.
 *
 * ARGUMENTS:   LPSTR                lpszQueryString
                client browser http command string
                PAYMENT_DATA          *pPaymentData
                pointer to payment data structure
 */

void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA *pPaymentData)
{
    char        szTmp[26];
    char        *ptr = lpszQueryString;
    BOOL        bCustIdBlank;

    pPaymentData->d_id = GetIntKeyValue(&ptr, "DID*",
ERR_PAYMENT_MISSING_DID_KEY, ERR_PAYMENT_DISTRICT_INVALID);

    GetKeyValue(&ptr, "CID*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_CID_KEY);
    if ( szTmp[0] == 0 )
    {
        bCustIdBlank = TRUE;
        pPaymentData->c_id = 0;
    }
}

```

```

    }
    else
    {
        // parse customer id and verify that last name was NOT entered
        bCustIdBlank = FALSE;
        if ( !IsNumeric(szTmp) )
            throw new CWEBCLNT_ERR( ERR_PAYMENT_CUSTOMER_INVALID
);
        pPaymentData->c_id = atoi(szTmp);
    }

    pPaymentData->c_w_id = GetIntKeyValue(&ptr, "CWI*",
ERR_PAYMENT_MISSING_CWI_KEY, ERR_PAYMENT_CWI_INVALID);
    pPaymentData->c_d_id = GetIntKeyValue(&ptr, "CDI*",
ERR_PAYMENT_MISSING_CDI_KEY, ERR_PAYMENT_CDI_INVALID);

    if ( bCustIdBlank )
    {
        // customer id is blank, so last name must be entered
        GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_CLT_KEY);
        if ( szTmp[0] == 0 )
            throw new CWEBCLNT_ERR( ERR_PAYMENT_MISSING_CID_CLT );
        strcpy( 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          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;
}

```

tpcc.def

LIBRARY TPCC.DLL

EXPORTS

```

GetExtensionVersion @1
HttpExtensionProc @2
TerminateExtension @3

```

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;
    //synchronization id
    int iTickCount;
    //time of last access;

    CTPCC_BASE *pTxn;
} CLIENTDATA, *PCLIENTDATA;

//This structure is used to define the operational interface for terminal id support
typedef struct _TERM
{
    int iNumEntries;
    //total allocated terminal array entries
    int iFreeList;
    //next available terminal array element or -1 if none
    int iMasterSyncId;
    //synchronization id

```

```

CLIENTDATA *pClientData;
//pointer to allocated client data
} TERM;

typedef TERM *PTERM;
//pointer to terminal structure type

enum WEBERROR
{
    NO_ERR,
    ERR_COMMAND_UNDEFINED,
    ERR_D_ID_INVALID,
    ERR_DELIVERY_CARRIER_ID_RANGE,
    ERR_DELIVERY_CARRIER_INVALID,
    ERR_DELIVERY_MISSING_OCD_KEY,
    ERR_DELIVERY_THREAD_FAILED,
    ERR_GETPROCADDR_FAILED,
    ERR_HTML_ILLEGAL_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_TOO_LONG,
    ERR_PAYMENT_MISSING_CDI_KEY,
    ERR_PAYMENT_MISSING_CID_CLT,
    ERR_PAYMENT_MISSING_CID_KEY,
    ERR_PAYMENT_MISSING_CLT,
    ERR_PAYMENT_MISSING_CLT_KEY,
    ERR_PAYMENT_MISSING_CWI_KEY,

```

```

ERR_PAYMENT_MISSING_DID_KEY,
ERR_PAYMENT_MISSING_HAM_KEY,
ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
ERR_STOCKLEVEL_THRESHOLD_INVALID,
ERR_STOCKLEVEL_THRESHOLD_RANGE,
ERR_VERSION_MISMATCH,
ERR_W_ID_INVALID
};

class CWEBCLNT_ERR : public CBaseErr
{
public:
    CWEBCLNT_ERR(WEBERROR Err)
    {
        m_Error = Err;
        m_szTextDetail = NULL;
        m_SystemErr = 0;
        m_szErrorText = NULL;
    };

    CWEBCLNT_ERR(WEBERROR Err, char *szTextDetail, DWORD
dwSystemErr)
    {
        m_Error = Err;
        m_szTextDetail = new char[strlen(szTextDetail)+1];
        strcpy(m_szTextDetail, szTextDetail);
        m_SystemErr = dwSystemErr;
        m_szErrorText = NULL;
    };

    ~CWEBCLNT_ERR()
    {
        if (m_szTextDetail != NULL)
            delete [] m_szTextDetail;
        if (m_szErrorText != NULL)
            delete [] m_szErrorText;
    };

    WEBERROR m_Error;
    char *m_szTextDetail; //
    char *m_szErrorText;
    DWORD m_SystemErr;

    int ErrorType() {return ERR_TYPE_WEBDLL;};
    int ErrorNum() {return m_Error;};
    char *ErrorText();
};

//These constants have already been defined in engstut.h, but since we do
//not want to include it in the delisrv executable
#define TXN_EVENT_START 2
#define TXN_EVENT_STOP 4
#define TXN_EVENT_WARNING 6 //used to record a warning into
the log

//function prototypes

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call, LPVOID lpReserved);
void WriteMessageToEventLog(LPTSTR lpszMsg);
void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd, int *pFormId, int
*pTermId, int *pSyncId);

```

```

void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId);
void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId);
void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int iError, int iErrorType, char
*szMsg, int iTermId);
void GetKeyValue(char **pQueryString, char *pKey, char *pValue, int iMax, WEBERROR
err);
int GetIntKeyValue(char **pQueryString, char *pKey, WEBERROR NoKeyErr, WEBERROR
NotIntErr);
void TermInit(void);
void TermDeleteAll(void);
int TermAdd(void);
void TermDelete(int id);
void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int iErrorNum, int iTermId,
int iSyncId, char *szErrorText, char *szBuffer);
void MakeMainMenuForm(int iTermId, int iSyncId, char *szForm);
void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA *pStockLevelData, BOOL bInput,
char *szForm);
void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA *pNewOrderData, BOOL bInput, char
*szForm);
void MakePaymentForm(int iTermId, PAYMENT_DATA *pPaymentData, BOOL bInput, char
*szForm);
void MakeOrderStatusForm(int iTermId, ORDER_STATUS_DATA *pOrderStatusData, BOOL
bInput, char *szForm);
void MakeDeliveryForm(int iTermId, DELIVERY_DATA *pDeliveryData, BOOL bInput, char
*szForm);
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char
*szBuffer);
void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char *szBuffer);
void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char
*szBuffer);
void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char
*szBuffer);
void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char
*szBuffer);
void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA *pNewOrderData);
void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA *pPaymentData);
void GetOrderStatusData(LPSTR lpszQueryString, ORDER_STATUS_DATA *pOrderStatusData);
BOOL PostDeliveryInfo(short w_id, short o_carrier_id);
BOOL IsNumeric(char *ptr);
BOOL IsDecimal(char *ptr);
void DeliveryWorkerThread(void *ptr);

```

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

#ifdef _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
////////////////////////////////////

#ifdef _MAC
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//

////////////////////////////////////
#endif // not APSTUDIO_INVOKED

```

tpcc_com.cpp

```
/* FILE: TPC_COM.CPP
 * Microsoft TPC-C Kit Ver. 4.20.000
 * Copyright Microsoft, 1999
 *
 * All Rights Reserved
 *
 * not yet audited
 *
 * PURPOSE: Source file for TPC-C COM+ class implementation.
 * Contact: Charles Levine (clevine@microsoft.com)
 *
 * Change history:
 * 4.20.000 - first version
 */

// needed for CoInitializeEx
#define WIN32_WINNT 0x0400

#include <windows.h>

// need to declare functions for export
#define DllDecl __declspec( dllexport )

#include "..\..\common\src\trans.h" //tpckit transaction header
contains definitions of structures specific to TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_com.h"

#include "..\..\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "..\..\tpcc_com_all\src\tpcc_com_all_i.c"

// wrapper routine for class constructor
_declspec( dllexport ) CTPCC_COM* CTPCC_COM_new(BOOL bSinglePool)
{
    return new CTPCC_COM(bSinglePool);
}

CTPCC_COM::CTPCC_COM(BOOL bSinglePool)
{
    HRESULT hr = NULL;
    long lRet = 0;
    ULONG ulTmpSize = 0;

    m_pTxn = NULL;
    m_pNewOrder = NULL;
    m_pPayment = NULL;
    m_pStockLevel = NULL;
    m_pOrderStatus = NULL;

    m_bSinglePool = bSinglePool;

    ulTmpSize = (ULONG) sizeof(COM_DATA);
    VariantInit(&m_vTxn);
    m_vTxn.vt = VT_SAFEARRAY;

    m_vTxn.parray = SafeArrayCreateVector(VT_UI1, ulTmpSize, ulTmpSize);
    if (!m_vTxn.parray)
        throw new CCOMERR( E_FAIL );

    memset( (void*)m_vTxn.parray->pvData, 0, ulTmpSize);
}
```

```
m_pTxn = (COM_DATA*)m_vTxn.parray->pvData;

hr = CoInitializeEx(NULL, COINIT_MULTITHREADED);
if (FAILED(hr))
{
    throw new CCOMERR( hr );
}

// create components
if (m_bSinglePool)
{
    hr = CoCreateInstance(CLSID_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);
}
```

```

        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 );
}

```

tpcc_com.h

```

/*      FILE:                TPCCOM.H
 *
 *      Microsoft TPC-C Kit Ver. 4.20.000
 *      Copyright Microsoft, 1999
 *
 *      All Rights Reserved
 *
 *      not yet audited
 *
 *      PURPOSE:  Header file for TPC-C COM+ class implementation.
 *
 *      Change history:
 *      4.20.000 - first version
 */

#pragma once

#include <stdio.h>
#include "..\..\tpcc_com_ps\src\tpcc_com_ps.h"

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl    __declspec( dllimport )
#endif

class CCOMERR : public CBaseErr
{
private:
    char m_szErrorText[64];

public:
    // use this interface for genuine COM errors
    CCOMERR( HRESULT hr )
    {
        m_hr = hr;
        m_iErrorType = 0;
        m_iError = 0;
    }

    // use this interface to impersonate a non-COM error type
    CCOMERR( int iErrorType, int iError )
    {
        m_iErrorType = iErrorType;
        m_iError = iError;
        m_hr = S_OK;
    }
}

```

```

    }

    int         m_hr;
    int         m_iErrorType;
    int         m_iError;

    // A CCOMERR class can impersonate another class, which happens
    // 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;

    public:
        VARIANT m_vTxn;
        CTPCC_COM(BOOL bSinglePool);
        ~CTPCC_COM(void);
};

```

```

        inline PNEW_ORDER_DATA         BuffAddr_NewOrder()
        { return &m_pTxn->u.NewOrder; };
        inline PPAYMENT_DATA           BuffAddr_Payment()
        { return &m_pTxn->u.Payment; };
        inline PDELIVERY_DATA          BuffAddr_Delivery()
        { return &m_pTxn->u.Delivery; };
        inline PSTOCK_LEVEL_DATA       BuffAddr_StockLevel()
        return &m_pTxn->u.StockLevel; };
        inline PORDER_STATUS_DATA      BuffAddr_OrderStatus()
        return &m_pTxn->u.OrderStatus; };

        void NewOrder           ();
        void Payment             ();
        void StockLevel          ();
        void OrderStatus        ();
        void Delivery            () { throw new CCOMERR(E_NOTIMPL);
    } // not supported
};

inline void ReleaseInterface(IUnknown *pUnk)
{
    if (pUnk)
    {
        pUnk->Release();
        pUnk = NULL;
    }
}

// wrapper routine for class constructor
extern "C" __declspec(dllexport) CTPCC_COM* CTPCC_COM_new(BOOL);

typedef CTPCC_COM* (TYPE_CTPCC_COM) (BOOL);

```

tpcc_com_all.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"));

    _sprintf(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. GetProcAddress 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)
        {
            strcat( szTmp, "Unknown error number." );
            break;
        }
        if (m_Error == errorMsgs[i].iError)
        {
            strcat( szTmp, errorMsgs[i].szMsg );
            break;
        }
        i++;
    }

    if (m_szTextDetail)
        strcat( szTmp, m_szTextDetail );
    if (m_SystemErr)
        wsprintf( szTmp+strlen(szTmp), " Error=%d", m_SystemErr );

    m_szErrorText = new char[strlen(szTmp)+1];
    strcpy( m_szErrorText, szTmp );
}

```

```

        return m_szErrorText;
    }

CTPCC_Common::CTPCC_Common()
{
    m_pTxn = NULL;
    m_bCanBePooled = TRUE;
}

CTPCC_Common::~CTPCC_Common()
{
    if (m_pTxn)
        delete m_pTxn;
}

HRESULT CTPCC_Common::CallSetComplete()
{
    IObjectContext* pObjectContext = NULL;

    // get our object context
    HRESULT hr = CoGetObjectContext( IID_IObjectContext, (void
**)&pObjectContext );
    pObjectContext->SetComplete();
    ReleaseInterface(pObjectContext);
    return hr;
}

//
// called by the ctor activator
//
STDMETHODIMP CTPCC_Common::Construct(IDispatch * pUnk)
{
    // Code to access construction string, if needed later...
    // if (!pUnk)
    //     return E_UNEXPECTED;
    // IObjectConstructString * pString = NULL;
    // HRESULT hr = pUnk->QueryInterface(IID_IObjectConstructString,
(void **)&pString);
    // pString->Release();

    try
    {
        if (Reg.eDB_Protocol == ODBC)
            m_pTxn = pCTPCC_ODBC_new( Reg.szDbServer,
Reg.szDbUser, Reg.szDbPassword, szMyComputerName, Reg.szDbName );
        else if (Reg.eDB_Protocol == DBLIB)
            m_pTxn = pCTPCC_DBLIB_new( Reg.szDbServer,
Reg.szDbUser, Reg.szDbPassword, szMyComputerName, Reg.szDbName );
    }
    catch (CBaseErr *e)
    {
        WriteMessageToEventLog(e->ErrorText());
        delete e;
        return E_FAIL;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled exception in object
::Construct"));
        return E_FAIL;
    }
}

```

```

        return S_OK;
    }

HRESULT CTPCC_Common::NewOrder(VARIANT txn_in, VARIANT* txn_out)
{
    PNEW_ORDER_DATA    pNewOrder;
    COM_DATA            *pData;
    try
    {
        pData = (COM_DATA*)txn_in.parray->pvData;
        pNewOrder = m_pTxn->BuffAddr_NewOrder();

        memcpy(pNewOrder, &pData->u.NewOrder, sizeof(NEW_ORDER_DATA));

        m_pTxn->NewOrder();           // do the actual txn

        VariantInit(txn_out);
        txn_out->vt = VT_SAFEARRAY;
        txn_out->parray = SafeArrayCreateVector(VT_UI1,
txn_in.parray->rgsabound->cElements,
txn_in.parray->rgsabound->cElements);
        pData = (COM_DATA*) txn_out->parray->pvData;

        memcpy( &pData->u.NewOrder, pNewOrder, sizeof(NEW_ORDER_DATA));

        pData->retval = ERR_SUCCESS;
        pData->error = 0;
        return S_OK;
    }
    catch (CBaseErr *e)
    {
        // check for lost database connection; if yes, component is
toast
        if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() ==
10005)) ||
        ((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum()
== 10054)) )
            m_bCanBePooled = FALSE;

        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        delete e;
        return E_FAIL;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled exception."));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        m_bCanBePooled = FALSE;
        return E_FAIL;
    }
}

HRESULT CTPCC_Common::Payment(VARIANT txn_in, VARIANT* txn_out)
{
    PPAYMENT_DATA      pPayment;
    COM_DATA            *pData;
    try
    {
        pData = (COM_DATA*)txn_in.parray->pvData;

```

```

        pPayment = m_pTxn->BuffAddr_Payment();

        memcpy(pPayment, &pData->u.Payment, sizeof(PAYMENT_DATA));

        m_pTxn->Payment();          // do the actual txn

        VariantInit(txn_out);
        txn_out->vt = VT_SAFEARRAY;
        txn_out->parray = SafeArrayCreateVector( VT_UI1,
        txn_in.parray-
>rgsabound->cElements,
        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)) ||
== 10054)) )
                ((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum()
                m_bCanBePooled = FALSE;

        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        delete e;
        return E_FAIL;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled exception."));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        m_bCanBePooled = FALSE;
        return E_FAIL;
    }
}

```

tpcc_com_all.def

; tpcc_com_all.def : Declares the module parameters.

```

LIBRARY      "tpcc_com_all.dll"

EXPORTS
DllCanUnloadNow      @1 PRIVATE
DllGetClassObject    @2 PRIVATE
DllRegisterServer    @3 PRIVATE
DllUnregisterServer  @4 PRIVATE

```

tpcc_com_all.dsp

```

# Microsoft Developer Studio Project File - Name="tpcc_com_all" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

```

```

# TARGETTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

```

```

CFG=tpcc_com_all - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_all.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE

```

```

!MESSAGE NMAKE /f "tpcc_com_all.mak" CFG="tpcc_com_all - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpcc_com_all - Win32 Release" (based on "Win32 (x86) Dynamic-Link
Library")
!MESSAGE "tpcc_com_all - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

```

```

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

```

```

!IF "$ (CFG)" == "tpcc_com_all - Win32 Release"

```

```

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /1 0x409 /d "NDEBUG"
# ADD RSC /1 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ..\db\dblib_dll\bin\tpcc_dblib.lib ..\db\odbc_dll\bin\tpcc_odbc.lib
kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows
/dll /machine:I386

```

```

!ELSEIF "$ (CFG)" == "tpcc_com_all - Win32 Debug"

```

```

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""

```

```

# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /1 0x409 /d "_DEBUG"
# ADD RSC /1 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ..\db\dblib_dll\bin\tpcc_dblib.lib ..\db\odbc_dll\bin\tpcc_odbc.lib
kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows
/dll /debug /machine:I386 /pdbtype:sept

!ENDIF

# Begin Target

# Name "tpcc_com_all - Win32 Release"
# Name "tpcc_com_all - Win32 Debug"
# Begin Group "Source"

# PROP Default_Filter "*.cpp, *.c"
# Begin Source File

SOURCE=.\src\tpcc_com_all.cpp
# SUBTRACT CPP /YX
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_all.def
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_all.idl

!IF "$(CFG)" == "tpcc_com_all - Win32 Release"

# PROP Ignore_Default_Tool 1
# Begin Custom Build - Performing MIDL step
InputPath=.\src\tpcc_com_all.idl

BuildCmds= \
midl /Oicf /h "tpcc_com_all.h" /iid "tpcc_com_all_i.c"
"..\src\tpcc_com_all.idl" /out "..\src"

"..\src\tpcc_com_all.tlb" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
$(BuildCmds)

"..\src\tpcc_com_all.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
$(BuildCmds)

"..\src\tpcc_com_all_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
$(BuildCmds)
# End Custom Build

!ELSEIF "$(CFG)" == "tpcc_com_all - Win32 Debug"

```

```

# PROP Ignore_Default_Tool 1
# Begin Custom Build - Performing MIDL step
InputPath=.\src\tpcc_com_all.idl

BuildCmds= \
midl /Oicf /h "tpcc_com_all.h" /iid "tpcc_com_all_i.c"
"..\src\tpcc_com_all.idl" /out "..\src"

"..\src\tpcc_com_all.tlb" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
$(BuildCmds)

"..\src\tpcc_com_all.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
$(BuildCmds)

"..\src\tpcc_com_all_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
$(BuildCmds)
# End Custom Build

!ENDIF

# End Source File
# End Group
# Begin Group "Header"

# PROP Default_Filter "*.h"
# Begin Source File

SOURCE=.\src\Methods.h
# End Source File
# Begin Source File

SOURCE=.\src\resource.h
# End Source File
# End Group
# Begin Source File

SOURCE=.\src\tpcc_com_all.rc
# End Source File
# End Target
# End Project

```

tpcc_com_all.h

```

#pragma warning( disable: 4049 ) /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the definitions for the interfaces */

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jun 12 18:15:19 2000
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

```

```

/* verify that the <rpcndr.h> version is high enough to compile this file*/
#ifndef REQUIRED_RPCNDR_H_VERSION
#define REQUIRED_RPCNDR_H_VERSION 440
#endif

#include "rpc.h"
#include "rpcndr.h"

#ifndef tpcc_com_all_h
#define tpcc_com_all_h

/* Forward Declarations */

#ifndef TPCC_FWD_DEFINED_
#define TPCC_FWD_DEFINED_

#ifdef __cplusplus
typedef Class TPCC TPCC;
#else
typedef struct TPCC TPCC;
#endif /* __cplusplus */

#endif /* TPCC_FWD_DEFINED_ */

#ifndef NewOrder_FWD_DEFINED_
#define NewOrder_FWD_DEFINED_

#ifdef __cplusplus
typedef class NewOrder NewOrder;
#else
typedef struct NewOrder NewOrder;
#endif /* __cplusplus */

#endif /* NewOrder_FWD_DEFINED_ */

#ifndef OrderStatus_FWD_DEFINED_
#define OrderStatus_FWD_DEFINED_

#ifdef __cplusplus
typedef class OrderStatus OrderStatus;
#else
typedef struct OrderStatus OrderStatus;
#endif /* __cplusplus */

#endif /* OrderStatus_FWD_DEFINED_ */

#ifndef Payment_FWD_DEFINED_
#define Payment_FWD_DEFINED_

#ifdef __cplusplus
typedef Class Payment Payment;
#else
typedef struct Payment Payment;
#endif /* __cplusplus */

#endif /* Payment_FWD_DEFINED_ */

#ifndef StockLevel_FWD_DEFINED_

```

```

#define __StockLevel_FWD_DEFINED__

#ifdef __cplusplus
typedef Class StockLevel StockLevel;
#else
typedef struct StockLevel StockLevel;
#endif /* __cplusplus */

#endif /* __StockLevel_FWD_DEFINED_ */

/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"
#include "tpcc_com_ps.h"

#ifdef __cplusplus
extern "C"{
#endif

void __RPC_FAR * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void __RPC_FAR * );

/* interface __MIDL_itf_tpcc_com_all_0000 */
/* [local] */

extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_s_ifspec;

#ifndef TPCCLib_LIBRARY_DEFINED_
#define TPCCLib_LIBRARY_DEFINED_

/* library TPCCLib */
/* [helpstring][version][uuid] */

EXTERN_C const IID LIBID_TPCCLib;

EXTERN_C const CLSID CLSID_TPCC;

#ifdef __cplusplus
class DECLSPEC_UUID("122A3128-2520-11D3-BA71-00C04FBFE08B")
TPCC;
#endif

EXTERN_C const CLSID CLSID_NewOrder;

#ifdef __cplusplus
class DECLSPEC_UUID("975BAABF-84A7-11D2-BA47-00C04FBFE08B")
NewOrder;
#endif

EXTERN_C const CLSID CLSID_OrderStatus;

```

```

#ifdef __cplusplus
class DECLSPEC_UUID("266836AD-A50D-11D2-BA4E-00C04FBFE08B")
OrderStatus;
#endif

EXTERN_C const CLSID CLSID_Payment;

#ifdef __cplusplus
class DECLSPEC_UUID("CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B")
Payment;
#endif

EXTERN_C const CLSID CLSID_StockLevel;

#ifdef __cplusplus
class DECLSPEC_UUID("2668369E-A50D-11D2-BA4E-00C04FBFE08B")
StockLevel;
#endif
#endif /* __TPCCLib_LIBRARY_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

```

tpcc_com_all.idl

```

/* FILE: TPCC.IDL
 * Microsoft TPC-C Kit Ver. 4.20.000
 * Copyright Microsoft, 1999
 *
 * All Rights Reserved
 *
 * not yet audited
 *
 * PURPOSE: IDL source for TPCC.dll. This file is processed by the MIDL
tool to
 * produce the type library (TPCC.tlb) and
marshalling code.
 *
 * Change history:
 * 4.20.000 - first version
 */

interface TPCC;
interface NewOrder;
interface OrderStatus;
interface Payment;
interface StockLevel;

import "oidl.idl";

```

```

import "oidl.idl";
import "..\tpcc_com_ps\src\tpcc_com_ps.idl";

[
    uuid(122A3117-2520-11D3-BA71-00C04FBFE08B),
    version(1.0),
    helpstring("TPC-C 1.0 Type Library")
]
library TPCCLib
{
    importlib("stdole32.tlb");
    importlib("stdole2.tlb");

    [
        uuid(122A3128-2520-11D3-BA71-00C04FBFE08B),
        helpstring("All Txns Class")
    ]
    coclass TPCC
    {
        [default] interface ITPCC;
    };

    [
        uuid(975BAABF-84A7-11D2-BA47-00C04FBFE08B),
        helpstring("NewOrder Class")
    ]
    coclass NewOrder
    {
        [default] interface ITPCC;
    };

    [
        uuid(266836AD-A50D-11D2-BA4E-00C04FBFE08B),
        helpstring("OrderStatus Class")
    ]
    coclass OrderStatus
    {
        [default] interface ITPCC;
    };

    [
        uuid(CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B),
        helpstring("Payment Class")
    ]
    coclass Payment
    {
        [default] interface ITPCC;
    };

    [
        uuid(2668369E-A50D-11D2-BA4E-00C04FBFE08B),
        helpstring("StockLevel Class")
    ]
    coclass StockLevel
    {
        [default] interface ITPCC;
    };
};

```


tpcc_com_all.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

#ifdef _MAC
////////////////////////////////////
//
// Version
//

VS_VERSION_INFO VERSIONINFO
FILEVERSION 1,0,0,1
PRODUCTVERSION 1,0,0,1
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
```

```
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x4L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904B0"
        BEGIN
            VALUE "CompanyName", "\0"
            VALUE "FileDescription", "tpcc_com_all Module\0"
            VALUE "FileVersion", "1, 0, 0, 1\0"
            VALUE "InternalName", "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
////////////////////////////////////

#ifdef APSTUDIO_INVOKED
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//
```

```

1 TYPELIB "tpcc_com_all.tlb"

////////////////////////////////////
#endif // not APSTUDIO_INVOKED

```

tpcc_com_all.rgs

```

HKCR
{
    TPCC.AllTxns.1 = s 'All Txns Class'
    {
        CLSID = s '{122A3128-2520-11D3-BA71-00C04FBFE08B}'
    }
    TPCC.AllTxns = s 'TPCC Class'
    {
        CurVer = s 'TPCC.AllTxns.1'
    }
    NoRemove CLSID
    {
        ForceRemove {122A3128-2520-11D3-BA71-00C04FBFE08B} = s 'TPCC
Class'
        {
            ProgID = s 'TPCC.AllTxns.1'
            VersionIndependentProgID = s 'TPCC.AllTxns'
            InprocServer32 = s '%MODULE%'
            {
                val ThreadingModel = s 'Both'
            }
        }
    }
}

```

tpcc_com_all.i.c

```

#pragma warning( disable: 4049 ) /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jun 12 18:15:19 2000 */
/*
 * Compiler settings for .\src\tpcc_com_all.idl:
 * Oicf (OptLev=i2), Wl, Zp8, env=Win32 (32b run), ms_ext, c_ext
 * error checks: allocation ref bounds_check enum stub_data
 * VC __declspec() decoration level:
 * __declspec(uuid()), __declspec(selectany), __declspec(novtable)
 * DECLSPEC_UUID(), MIDL_INTERFACE()
 */
//@@MIDL_FILE_HEADING( )

#if !defined(_M_IA64) && !defined(_M_AXP64)

#ifdef _cplusplus
extern "C"{
#endif

```

```

#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
    DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
    const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_Payment,0xCD02F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

```

```

MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif /* !defined(_M_IA64) && !defined(_M_AXP64) */

#pragma warning( disable: 4049 ) /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jun 12 18:15:19 2000 */
/*
Compiler settings for .\src\tpcc_com_all.idl:
Oicf (OptLev=i2), Wl, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext, robust
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#ifdef _M_IA64 || defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif

#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else /* !_MIDL_USE_GUIDDEF_

#ifdef __cplusplus
}
#endif

typedef struct _IID
{

```

```

    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char c[8];
} IID;

#endif // __IID_DEFINED__

#ifdef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_Payment,0xCD02F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif /* defined(_M_IA64) || defined(_M_AXP64) */

```

tpcc_com_no.rgs

```

HKCR
{
    TPCC.NewOrder.1 = s 'NewOrder Class'
    {
        CLSID = s '{975BAABF-84A7-11D2-BA47-00C04FBFE08B}'
    }
    TPCC.NewOrder = s 'NewOrder Class'
    {

```

```

        CurVer = s 'TPCC.NewOrder.1'
    }
    NoRemove CLSID
    {
        ForceRemove {975BAABF-84A7-11D2-BA47-00C04FBFE08B} = s 'NewOrder
Class'
        {
            ProgID = s 'TPCC.NewOrder.1'
            VersionIndependentProgID = s 'TPCC.NewOrder'
            InprocServer32 = s '%MODULE%'
            {
                val ThreadingModel = s 'Both'
            }
        }
    }
}

```

tpcc_com_os.rgs

```

HKCR
{
    TPCC.OrderStatus.1 = s 'OrderStatus Class'
    {
        CLSID = s '{266836AD-A50D-11D2-BA4E-00C04FBFE08B}'
    }
    TPCC.OrderStatus = s 'OrderStatus Class'
    {
        CurVer = s 'TPCC.OrderStatus.1'
    }
    NoRemove CLSID
    {
        ForceRemove {266836AD-A50D-11D2-BA4E-00C04FBFE08B} = s
'OrderStatus Class'
        {
            ProgID = s 'TPCC.OrderStatus.1'
            VersionIndependentProgID = s 'TPCC.OrderStatus'
            InprocServer32 = s '%MODULE%'
            {
                val ThreadingModel = s 'Both'
            }
        }
    }
}

```

tpcc_com_pay.rgs

```

HKCR
{
    TPCC.Payment.1 = s 'Payment Class'
    {
        CLSID = s '{CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B}'
    }
    TPCC.Payment = s 'Payment Class'
    {
        CurVer = s 'TPCC.Payment.1'
    }
    NoRemove CLSID
    {
        ForceRemove {CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B} = s 'Payment
Class'
        {

```

```

        ProgID = s 'TPCC.Payment.1'
        VersionIndependentProgID = s 'TPCC.Payment'
        InprocServer32 = s '%MODULE%'
        {
            val ThreadingModel = s 'Both'
        }
    }
}

```

tpcc_com_ps.def

```

LIBRARY      "tpcc_com_ps"

DESCRIPTION  'Proxy/Stub DLL'

EXPORTS
    DllGetClassObject      @1    PRIVATE
    DllCanUnloadNow        @2    PRIVATE
    GetProxyDllInfo        @3    PRIVATE
    DllRegisterServer      @4    PRIVATE
    DllUnregisterServer    @5    PRIVATE

```

tpcc_com_ps.dsp

```

# Microsoft Developer Studio Project File - Name="tpcc_com_ps" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGETTYPE "Win32 (x86) Application" 0x0101

CFG=tpcc_com_ps - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_ps.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_ps.mak" CFG="tpcc_com_ps - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpcc_com_ps - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "tpcc_com_ps - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF "$(CFG)" == "tpcc_com_ps - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0

```

```

# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WIN32_WINNT=0x0400" /D
"REGISTER_PROXY_DLL" /FD /c
# SUBTRACT CPP /YX
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpcrt4.lib oleaut32.lib uuid.lib
/nologo /entry:"DllMain" /subsystem:windows /dll /pdb:none /machine:I386
/def:".src\tpcc_com_ps.def"
# Begin Custom Build - Copying tpcc_com_ps.h
InputPath=.bin\tpcc_com_ps.dll
SOURCE="$(InputPath)"

"..\tpcc_com_all\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
copy .\src\tpcc_com_ps.h ..\tpcc_com_all\src\

# End Custom Build

!ELSEIF "$(CFG)" == "tpcc_com_ps - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD CPP /nologo /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WIN32_WINNT=0x0400" /D
"REGISTER_PROXY_DLL" /FD /c
# ADD BASE MTL /nologo /D "DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe

```

```

# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbcc32.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386 /pdftype:sept
# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpcrt4.lib oleaut32.lib uuid.lib
/nologo /entry:"DllMain" /dll /debug /machine:IX86 /def:".src\tpcc_com_ps.def"
/pdftype:sept
# SUBTRACT LINK32 /pdb:none
# Begin Custom Build - Copying tpcc_com_ps.h
InputPath=.bin\tpcc_com_ps.dll
SOURCE="$(InputPath)"

"..\tpcc_com_all\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
copy .\src\tpcc_com_ps.h ..\tpcc_com_all\src\

# End Custom Build

!ENDIF

# Begin Target

# Name "tpcc_com_ps - Win32 Release"
# Name "tpcc_com_ps - Win32 Debug"
# Begin Group "Source"

# PROP Default_Filter ""
# Begin Source File

SOURCE=.src\dll\data.c
# End Source File
# Begin Source File

SOURCE=.src\tpcc_com_ps.def
# PROP Exclude_From_Build 1
# End Source File
# Begin Source File

SOURCE=.src\tpcc_com_ps.idl

!IF "$(CFG)" == "tpcc_com_ps - Win32 Release"

# PROP Ignore_Default_Tool 1
# Begin Custom Build
InputPath=.src\tpcc_com_ps.idl

BuildCmds= \
midl /Oicf /h "tpcc_com_ps.h" /iid "tpcc_com_ps_i.c"
"\src\tpcc_com_ps.idl" /out ".\src"

"\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
$(BuildCmds)

"\src\tpcc_com_ps_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
$(BuildCmds)

"\src\dll\data.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
$(BuildCmds)

"\src\tpcc_com_ps_p.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
$(BuildCmds)
# End Custom Build

!ELSEIF "$(CFG)" == "tpcc_com_ps - Win32 Debug"

```

```

# PROP Ignore_Default_Tool 1
# Begin Custom Build
InputPath=. \src\tpcc_com_ps.idl

BuildCmds= \
    midl /Oicf /h "tpcc_com_ps.h" /iid "tpcc_com_ps_i.c"
".\src\tpcc_com_ps.idl" /out ".\src"

".\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
$(BuildCmds)

".\src\tpcc_com_ps_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
$(BuildCmds)

".\src\dlldata.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
$(BuildCmds)

".\src\tpcc_com_ps_p.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
$(BuildCmds)
# End Custom Build

!ENDIF

# End Source File
# Begin Source File

SOURCE=. \src\tpcc_com_ps_i.c
# End Source File
# Begin Source File

SOURCE=. \src\tpcc_com_ps_p.c
# End Source File
# End Group
# End Target
# End Project

```

tpcc_com_ps.h

```

#pragma warning( disable: 4049 ) /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the definitions for the interfaces */

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jun 12 18:15:12 2000 */
/*
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
        DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

/* verify that the <rpcndr.h> version is high enough to compile this file*/
#ifndef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440
#endif

```

```

#include "rpc.h"
#include "rpcndr.h"

#ifndef __RPCNDR_H_VERSION__
#error this stub requires an updated version of <rpcndr.h>
#endif // __RPCNDR_H_VERSION__

#ifndef COM_NO_WINDOWS_H
#include "windows.h"
#include "ole2.h"
#endif /*COM_NO_WINDOWS_H*/

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

/* Forward Declarations */

#ifndef __ITPCC_FWD_DEFINED__
#define __ITPCC_FWD_DEFINED__
typedef interface ITPCC ITPCC;
#endif /* __ITPCC_FWD_DEFINED__ */

/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"

#ifdef __cplusplus
extern "C"{
#endif

void __RPC_FAR * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void __RPC_FAR * );

/* interface __MIDL_itf_tpcc_com_ps_0000 */
/* [local] */

extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

/* interface ITPCC */
/* [unique][helpstring][uuid][oleautomation][object] */

EXTERN_C const IID IID_ITPCC;

#if defined(__cplusplus) && !defined(CINTERFACE)

MIDL_INTERFACE("FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
ITPCC : public IUnknown
{
public:
    virtual HRESULT __stdcall NewOrder(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

    virtual HRESULT __stdcall Payment(

```

```

        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

virtual HRESULT __stdcall Delivery(
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

virtual HRESULT __stdcall StockLevel(
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

virtual HRESULT __stdcall OrderStatus(
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

virtual HRESULT __stdcall CallSetComplete( void) = 0;

};

#else    /* C style interface */

typedef struct ITPCCVtbl
{
    BEGIN_INTERFACE

    HRESULT ( STDMETHODCALLTYPE __RPC_FAR *QueryInterface )(
        ITPCC __RPC_FAR * This,
        /* [in] */ REFIID riid,
        /* [iid_is][out] */ void __RPC_FAR * __RPC_FAR *ppvObject);

    ULONG ( STDMETHODCALLTYPE __RPC_FAR *AddRef )(
        ITPCC __RPC_FAR * This);

    ULONG ( STDMETHODCALLTYPE __RPC_FAR *Release )(
        ITPCC __RPC_FAR * This);

    HRESULT ( STDMETHODCALLTYPE __RPC_FAR *NewOrder )(
        ITPCC __RPC_FAR * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT __RPC_FAR *txn_out);

    HRESULT ( STDMETHODCALLTYPE __RPC_FAR *Payment )(
        ITPCC __RPC_FAR * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT __RPC_FAR *txn_out);

    HRESULT ( STDMETHODCALLTYPE __RPC_FAR *Delivery )(
        ITPCC __RPC_FAR * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT __RPC_FAR *txn_out);

    HRESULT ( STDMETHODCALLTYPE __RPC_FAR *StockLevel )(
        ITPCC __RPC_FAR * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT __RPC_FAR *txn_out);

    HRESULT ( STDMETHODCALLTYPE __RPC_FAR *OrderStatus )(
        ITPCC __RPC_FAR * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT __RPC_FAR *txn_out);

    HRESULT ( STDMETHODCALLTYPE __RPC_FAR *CallSetComplete )(
        ITPCC __RPC_FAR * This);
};

```

```

        END_INTERFACE
    } ITPCCVtbl;

interface ITPCC
{
    CONST_VTBL struct ITPCCVtbl __RPC_FAR *lpVtbl;
};

#ifdef COBJMACROS

#define ITPCC_QueryInterface(This,riid,ppvObject) \
    (This->lpVtbl -> QueryInterface(This,riid,ppvObject))

#define ITPCC_AddRef(This) \
    (This->lpVtbl -> AddRef(This))

#define ITPCC_Release(This) \
    (This->lpVtbl -> Release(This))

#define ITPCC_NewOrder(This,txn_in,txn_out) \
    (This->lpVtbl -> NewOrder(This,txn_in,txn_out))

#define ITPCC_Payment(This,txn_in,txn_out) \
    (This->lpVtbl -> Payment(This,txn_in,txn_out))

#define ITPCC_Delivery(This,txn_in,txn_out) \
    (This->lpVtbl -> Delivery(This,txn_in,txn_out))

#define ITPCC_StockLevel(This,txn_in,txn_out) \
    (This->lpVtbl -> StockLevel(This,txn_in,txn_out))

#define ITPCC_OrderStatus(This,txn_in,txn_out) \
    (This->lpVtbl -> OrderStatus(This,txn_in,txn_out))

#define ITPCC_CallSetComplete(This) \
    (This->lpVtbl -> CallSetComplete(This))

#endif /* COBJMACROS */

#endif    /* C style interface */

HRESULT __stdcall ITPCC_NewOrder_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);

void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_Payment_Proxy(

```

```

ITPCC __RPC_FAR * This,
/* [in] */ VARIANT txn_in,
/* [out] */ VARIANT __RPC_FAR *txn_out);

void __RPC_STUB ITPCC_Payment_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_Delivery_Proxy(
ITPCC __RPC_FAR * This,
/* [in] */ VARIANT txn_in,
/* [out] */ VARIANT __RPC_FAR *txn_out);

void __RPC_STUB ITPCC_Delivery_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_StockLevel_Proxy(
ITPCC __RPC_FAR * This,
/* [in] */ VARIANT txn_in,
/* [out] */ VARIANT __RPC_FAR *txn_out);

void __RPC_STUB ITPCC_StockLevel_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_OrderStatus_Proxy(
ITPCC __RPC_FAR * This,
/* [in] */ VARIANT txn_in,
/* [out] */ VARIANT __RPC_FAR *txn_out);

void __RPC_STUB ITPCC_OrderStatus_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
ITPCC __RPC_FAR * This);

void __RPC_STUB ITPCC_CallSetComplete_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

```

```

#endif /* __ITPCC_INTERFACE_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

unsigned long             __RPC_USER  VARIANT_UserSize(      unsigned long __RPC_FAR
*, unsigned long         , VARIANT __RPC_FAR * );
unsigned char __RPC_FAR * __RPC_USER  VARIANT_UserMarshal(  unsigned long __RPC_FAR
*, unsigned char __RPC_FAR *, VARIANT __RPC_FAR * );
unsigned char __RPC_FAR * __RPC_USER  VARIANT_UserUnmarshal(unsigned long __RPC_FAR
*, unsigned char __RPC_FAR *, VARIANT __RPC_FAR * );
void                   __RPC_USER  VARIANT_UserFree(       unsigned long __RPC_FAR
*, VARIANT __RPC_FAR * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

```

tpcc_com_ps.idl

```

/* FILE:                ITPCC.IDL
 *                      Microsoft TPC-C Kit Ver. 4.20.000
 *                      Copyright Microsoft, 1999
 *
 *                      All Rights Reserved
 *
 *                      not yet audited
 *
 * PURPOSE: Defines the interface used by TPCC. This interface can be
implemented by C++ components.
 *
 * Change history:
 * 4.20.000 - first version
 */

// Forward declare all types defined
interface ITPCC;
import "oidl.idl";
import "ocidl.idl";

[
    object,
    oleautomation,
    uuid(FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B),
    helpstring("ITPCC Interface"),
    pointer_default(unique)
]
interface ITPCC : IUnknown
{
    HRESULT __stdcall NewOrder(
        [in] VARIANT txn_in,
        [out] VARIANT *txn_out
    );

    HRESULT __stdcall Payment

```



```

(
[in] VARIANT txn_in,
[out] VARIANT *txn_out
);

HRESULT _stdcall Delivery

(
[in] VARIANT txn_in,
[out] VARIANT *txn_out
);

HRESULT _stdcall StockLevel

(
[in] VARIANT txn_in,
[out] VARIANT *txn_out
);

HRESULT _stdcall OrderStatus

(
[in] VARIANT txn_in,
[out] VARIANT *txn_out
);

HRESULT _stdcall CallSetComplete

(
);

}; // interface ITPCC

```

tpcc_com_ps_i.c

```

#pragma warning( disable: 4049 ) /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jun 12 18:15:12 2000 */
/*
/* Compiler settings for .\src\tpcc_com.ps.idl:
Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@MIDL_FILE_HEADING( )

#if !defined(_M_IA64) && !defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif

#include <rpc.h>
#include <rpcndr.h>

```

```

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
IID_ITPCC,0xFEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

#endif /* !defined(_M_IA64) && !defined(_M_AXP64) */

#pragma warning( disable: 4049 ) /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jun 12 18:15:12 2000 */
*/

```

```

/* Compiler settings for .\src\tpcc_com_ps.idl:
   Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext, robust
   error checks: allocation ref bounds_check enum stub_data
   VC __declspec() decoration level:
       __declspec(uuid()), __declspec(selectany), __declspec(novtable)
       DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if defined(_M_IA64) || defined(_M_AXP64)

#ifdef __cplusplus
extern "C"{
#endif

#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
    DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
    const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif ! _MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

```

```

}
#endif

#endif /* defined(_M_IA64) || defined(_M_AXP64)*/

```

tpcc_com_ps_p.c

```

#pragma warning( disable: 4049 ) /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the proxy stub code */

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jun 12 18:15:12 2000
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
   Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
   error checks: allocation ref bounds_check enum stub_data
   VC __declspec() decoration level:
       __declspec(uuid()), __declspec(selectany), __declspec(novtable)
       DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if !defined(_M_IA64) && !defined(_M_AXP64)
#define USE_STUBLESS_PROXY

/* verify that the <rpcproxy.h> version is high enough to compile this file*/
#ifdef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 440
#endif

#include "rpcproxy.h"
#ifdef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of <rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 997
#define PROC_FORMAT_STRING_SIZE 193
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 1

typedef struct _MIDL_TYPE_FORMAT_STRING
{
    short Pad;
    unsigned char Format[ TYPE_FORMAT_STRING_SIZE ];
} MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
{
    short Pad;
    unsigned char Format[ PROC_FORMAT_STRING_SIZE ];
} MIDL_PROC_FORMAT_STRING;

```

```

extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,
   GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
   GUID={0x00000000,0x0000,0x0000,{0xc0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
   GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[] =
{
    0,
    34,
    68,
    102,
    136,
    170
};

static const MIDL_SERVER_INFO ITPCC_ServerInfo =
{
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0
};

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo =
{
    &Object_StubDesc,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0
};

CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,

```

```

(void *)-1 /* ITPCC::NewOrder */ ,
(void *)-1 /* ITPCC::Payment */ ,
(void *)-1 /* ITPCC::Delivery */ ,
(void *)-1 /* ITPCC::StockLevel */ ,
(void *)-1 /* ITPCC::OrderStatus */ ,
(void *)-1 /* ITPCC::CallSetComplete */
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,
    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

extern const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ];

static const MIDL_STUB_DESC Object_StubDesc =
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.3.280 */
    0,
    UserMarshalRoutines,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* Reserved3 */
    0, /* Reserved4 */
    0 /* Reserved5 */
};

#pragma data_seg(".rdata")

static const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ] =
{
    {
        VARIANT_UserSize
        ,VARIANT_UserMarshal
        ,VARIANT_UserUnmarshal
        ,VARIANT_UserFree
    }
};

#if !defined(_RPC_WIN32_)
#error Invalid build platform for this stub.
#endif

```

```

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this stub because it uses these
features:
#error -Oif or -Oicf, [wire_marshall] or [user_marshall] 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 */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 8 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
NdrFcShort( 0x20 ), /* MIPS Stack size/offset = 32 */
#endif
#endif
NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 10 */ NdrFcShort( 0x0 ), /* 0 */
/* 12 */ NdrFcShort( 0x8 ), /* 8 */
/* 14 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has
return, */
0x3, /* 3 */

        /* Parameter txn_in */
        /* 16 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 18 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
NdrFcShort( 0x8 ), /* MIPS Stack size/offset = 8 */
#endif
#endif
NdrFcShort( 0x8 ), /* PPC Stack size/offset = 8 */
#endif
NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 20 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

        /* Parameter txn_out */
        /* 22 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=16 */

```

```

#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 24 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
NdrFcShort( 0x18 ), /* MIPS Stack size/offset = 24 */
#endif
#endif
NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#else
NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 26 */ NdrFcShort( 0x3da ), /* Type Offset=986 */

        /* Return value */
        /* 28 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 30 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
NdrFcShort( 0x1c ), /* MIPS Stack size/offset = 28 */
#endif
#endif
NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 32 */ 0x8, /* FC_LONG */
0x0, /* 0 */

        /* Procedure Payment */
        /* 34 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
        /* 36 */ NdrFcLong( 0x0 ), /* 0 */
        /* 40 */ NdrFcShort( 0x4 ), /* 4 */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 42 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
NdrFcShort( 0x20 ), /* MIPS Stack size/offset = 32 */
#endif
#endif
NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 44 */ NdrFcShort( 0x0 ), /* 0 */
/* 46 */ NdrFcShort( 0x8 ), /* 8 */
/* 48 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has
return, */
0x3, /* 3 */

        /* Parameter txn_in */
        /* 50 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef _ALPHA_

```

```

#endif
#endif
#endif
/* 52 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
NdrFcShort( 0x8 ), /* MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 54 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Parameter txn_out */

/* 56 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 58 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
NdrFcShort( 0x18 ), /* MIPS Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 60 */ NdrFcShort( 0x3da ), /* Type Offset=986 */

/* Return value */

/* 62 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 64 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
NdrFcShort( 0x1c ), /* MIPS Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 66 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure Delivery */

/* 68 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 70 */ NdrFcLong( 0x0 ), /* 0 */
/* 74 */ NdrFcShort( 0x5 ), /* 5 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 76 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else

```

```

NdrFcShort( 0x20 ), /* MIPS Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 78 */ NdrFcShort( 0x0 ), /* 0 */
/* 80 */ NdrFcShort( 0x8 ), /* 8 */
/* 82 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has
return, */
0x3, /* 3 */

/* Parameter txn_in */

/* 84 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 86 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
NdrFcShort( 0x8 ), /* MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 88 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Parameter txn_out */

/* 90 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 92 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
NdrFcShort( 0x18 ), /* MIPS Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 94 */ NdrFcShort( 0x3da ), /* Type Offset=986 */

/* Return value */

/* 96 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 98 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
NdrFcShort( 0x1c ), /* MIPS Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif

```

```

#endif
#else
                                NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 100 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure StockLevel */

/* 102 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 104 */ NdrFcLong( 0x0 ), /* 0 */
/* 108 */ NdrFcShort( 0x6 ), /* 6 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 110 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
                                NdrFcShort( 0x20 ), /* MIPS Stack size/offset = 32 */
#endif
#endif
#else
                                NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#endif
/* 112 */ NdrFcShort( 0x0 ), /* 0 */
/* 114 */ NdrFcShort( 0x8 ), /* 8 */
/* 116 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has
return, */
0x3, /* 3 */

/* Parameter txn_in */

/* 118 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 120 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
                                NdrFcShort( 0x8 ), /* MIPS Stack size/offset = 8 */
#endif
#endif
#else
                                NdrFcShort( 0x8 ), /* PPC Stack size/offset = 8 */
#endif
#endif
/* 122 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Parameter txn_out */

/* 124 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 126 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
                                NdrFcShort( 0x18 ), /* MIPS Stack size/offset = 24 */
#endif
#endif
#else
                                NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */

```

```

#endif
#else
                                NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 128 */ NdrFcShort( 0x3da ), /* Type Offset=986 */

/* Return value */

/* 130 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 132 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
                                NdrFcShort( 0x1c ), /* MIPS Stack size/offset = 28 */
#endif
#endif
#else
                                NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#endif
/* 134 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure OrderStatus */

/* 136 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 138 */ NdrFcLong( 0x0 ), /* 0 */
/* 142 */ NdrFcShort( 0x7 ), /* 7 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 144 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
                                NdrFcShort( 0x20 ), /* MIPS Stack size/offset = 32 */
#endif
#endif
#else
                                NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#endif
/* 146 */ NdrFcShort( 0x0 ), /* 0 */
/* 148 */ NdrFcShort( 0x8 ), /* 8 */
/* 150 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has
return, */
0x3, /* 3 */

/* Parameter txn_in */

/* 152 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 154 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
#else
                                NdrFcShort( 0x8 ), /* MIPS Stack size/offset = 8 */
#endif
#endif
#else
                                NdrFcShort( 0x8 ), /* PPC Stack size/offset = 8 */

```

```

#else
                NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 156 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

        /* Parameter txn_out */

/* 158 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 160 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
                NdrFcShort( 0x18 ), /* MIPS Stack size/offset = 24 */
#endif
#else
                NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
                NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 162 */ NdrFcShort( 0x3da ), /* Type Offset=986 */

        /* Return value */

/* 164 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined( _MIPS_ )
/* 166 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
                NdrFcShort( 0x1c ), /* MIPS Stack size/offset = 28 */
#endif
#else
                NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#else
                NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 178 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

        /* Procedure CallSetComplete */

/* 170 */ 0x33, /* FC_AUTO_HANDLE */
        0x6c, /* Old Flags: object, Oi2 */

/* 172 */ NdrFcLong( 0x0 ), /* 0 */
/* 176 */ NdrFcShort( 0x8 ), /* 8 */
#ifndef _ALPHA_
/* 178 */ NdrFcShort( 0x8 ), /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                NdrFcShort( 0x10 ), /* Alpha Stack size/offset = 16 */
#endif
/* 180 */ NdrFcShort( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x8 ), /* 8 */
/* 184 */ 0x4, /* Oi2 Flags: has return, */
        0x1, /* 1 */

        /* Return value */

/* 186 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifndef _ALPHA_

```

```

/* 188 */ NdrFcShort( 0x4 ), /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                NdrFcShort( 0x8 ), /* Alpha Stack size/offset = 8 */
#endif
/* 190 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

        0x0

    }
};

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
    0,
    {
        NdrFcShort( 0x0 ), /* 0 */

/* 2 */

        0x12, 0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0x3b0 ), /* Offset= 944 (948) */
/* 6 */

        0x2b, /* FC_NON_ENCAPSULATED_UNION */
        0x9, /* FC_ULONGLONG */
/* 8 */ 0x7, /* Corr desc: FC_USHORT */
        0x0, /* */

/* 10 */ NdrFcShort( 0xffff ), /* -8 */
/* 12 */ NdrFcShort( 0x2 ), /* Offset= 2 (14) */
/* 14 */ NdrFcShort( 0x10 ), /* 16 */
/* 16 */ NdrFcShort( 0x2b ), /* 43 */
/* 18 */ NdrFcLong( 0x3 ), /* 3 */
/* 22 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 24 */ NdrFcLong( 0x11 ), /* 17 */
/* 28 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE */
/* 30 */ NdrFcLong( 0x2 ), /* 2 */
/* 34 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 36 */ NdrFcLong( 0x4 ), /* 4 */
/* 40 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT */
/* 42 */ NdrFcLong( 0x5 ), /* 5 */
/* 46 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 48 */ NdrFcLong( 0xb ), /* 11 */
/* 52 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 54 */ NdrFcLong( 0xa ), /* 10 */
/* 58 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 60 */ NdrFcLong( 0x6 ), /* 6 */
/* 64 */ NdrFcShort( 0xd6 ), /* Offset= 214 (278) */
/* 66 */ NdrFcLong( 0x7 ), /* 7 */
/* 70 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 72 */ NdrFcLong( 0x8 ), /* 8 */
/* 76 */ NdrFcShort( 0xd0 ), /* Offset= 208 (284) */
/* 78 */ NdrFcLong( 0xd ), /* 13 */
/* 82 */ NdrFcShort( 0xe2 ), /* Offset= 226 (308) */
/* 84 */ NdrFcLong( 0x9 ), /* 9 */
/* 88 */ NdrFcShort( 0xee ), /* Offset= 238 (326) */
/* 90 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 94 */ NdrFcShort( 0xfa ), /* Offset= 250 (344) */
/* 96 */ NdrFcLong( 0x24 ), /* 36 */
/* 100 */ NdrFcShort( 0x308 ), /* Offset= 776 (876) */
/* 102 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 106 */ NdrFcShort( 0x302 ), /* Offset= 770 (876) */
/* 108 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 112 */ NdrFcShort( 0x300 ), /* Offset= 768 (880) */
/* 114 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 118 */ NdrFcShort( 0x2fe ), /* Offset= 766 (884) */
/* 120 */ NdrFcLong( 0x4003 ), /* 16387 */

```

```

/* 124 */ NdrFcShort( 0x2fc ), /* Offset= 764 (888) */
/* 126 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 130 */ NdrFcShort( 0x2fa ), /* Offset= 762 (892) */
/* 132 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 136 */ NdrFcShort( 0x2f8 ), /* Offset= 760 (896) */
/* 138 */ NdrFcLong( 0x400b ), /* 16395 */
/* 142 */ NdrFcShort( 0x2e6 ), /* Offset= 742 (884) */
/* 144 */ NdrFcLong( 0x400a ), /* 16394 */
/* 148 */ NdrFcShort( 0x2e4 ), /* Offset= 740 (888) */
/* 150 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 154 */ NdrFcShort( 0x2ea ), /* Offset= 746 (900) */
/* 156 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 160 */ NdrFcShort( 0x2e0 ), /* Offset= 736 (896) */
/* 162 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 166 */ NdrFcShort( 0x2e2 ), /* Offset= 738 (904) */
/* 168 */ NdrFcLong( 0x400d ), /* 16397 */
/* 172 */ NdrFcShort( 0x2e0 ), /* Offset= 736 (908) */
/* 174 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 178 */ NdrFcShort( 0x2de ), /* Offset= 734 (912) */
/* 180 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 184 */ NdrFcShort( 0x2dc ), /* Offset= 732 (916) */
/* 186 */ NdrFcLong( 0x400c ), /* 16396 */
/* 190 */ NdrFcShort( 0x2da ), /* Offset= 730 (920) */
/* 192 */ NdrFcLong( 0x10 ), /* 16 */
/* 196 */ NdrFcShort( 0x8002 ), /* Simple arm type: FC_CHAR */
/* 198 */ NdrFcLong( 0x12 ), /* 18 */
/* 202 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 204 */ NdrFcLong( 0x13 ), /* 19 */
/* 208 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 210 */ NdrFcLong( 0x16 ), /* 22 */
/* 214 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 216 */ NdrFcLong( 0x17 ), /* 23 */
/* 220 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 222 */ NdrFcLong( 0xe ), /* 14 */
/* 226 */ NdrFcShort( 0x2be ), /* Offset= 702 (928) */
/* 228 */ NdrFcLong( 0x400e ), /* 16398 */
/* 232 */ NdrFcShort( 0x2c4 ), /* Offset= 708 (940) */
/* 234 */ NdrFcLong( 0x4010 ), /* 16400 */
/* 238 */ NdrFcShort( 0x2c2 ), /* Offset= 706 (944) */
/* 240 */ NdrFcLong( 0x4012 ), /* 16402 */
/* 244 */ NdrFcShort( 0x280 ), /* Offset= 640 (884) */
/* 246 */ NdrFcLong( 0x4013 ), /* 16403 */
/* 250 */ NdrFcShort( 0x27e ), /* Offset= 638 (888) */
/* 252 */ NdrFcLong( 0x4016 ), /* 16406 */
/* 256 */ NdrFcShort( 0x278 ), /* Offset= 632 (888) */
/* 258 */ NdrFcLong( 0x4017 ), /* 16407 */
/* 262 */ NdrFcShort( 0x272 ), /* Offset= 626 (888) */
/* 264 */ NdrFcLong( 0x0 ), /* 0 */
/* 268 */ NdrFcShort( 0x0 ), /* Offset= 0 (268) */
/* 270 */ NdrFcLong( 0x1 ), /* 1 */
/* 274 */ NdrFcShort( 0x0 ), /* Offset= 0 (274) */
/* 276 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (275) */
/* 278 */

0x15, /* FC_STRUCT */
0x7, /* 7 */

/* 280 */ NdrFcShort( 0x8 ), /* 8 */
/* 282 */ 0xb, /* FC_HYPER */

0x5b, /* FC_END */

/* 284 */

0x12, 0x0, /* FC_UP */
/* 286 */ NdrFcShort( 0xc ), /* Offset= 12 (298) */
/* 288 */

0x1b, /* FC_CARRAY */

```

```

0x1, /* 1 */
/* 290 */ NdrFcShort( 0x2 ), /* 2 */
/* 292 */ 0x9, /* Corr desc: FC_ULONG */
0x0, /* */
/* 294 */ NdrFcShort( 0xffffc ), /* -4 */
/* 296 */ 0x6, /* FC_SHORT */
0x5b, /* FC_END */

/* 298 */

0x17, /* FC_CSTRUCT */
0x3, /* 3 */

/* 300 */ NdrFcShort( 0x8 ), /* 8 */
/* 302 */ NdrFcShort( 0xffffffff2 ), /* Offset= -14 (288) */
/* 304 */ 0x8, /* FC_LONG */
0x8, /* FC_LONG */
/* 306 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */

/* 308 */

0x2f, /* FC_IP */
0x5a, /* FC_CONSTANT_IID */

/* 310 */ NdrFcLong( 0x0 ), /* 0 */
/* 314 */ NdrFcShort( 0x0 ), /* 0 */
/* 316 */ NdrFcShort( 0x0 ), /* 0 */
/* 318 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 320 */ 0x0, /* 0 */
0x0, /* 0 */
/* 322 */ 0x0, /* 0 */
0x0, /* 0 */
/* 324 */ 0x0, /* 0 */
0x46, /* 70 */

/* 326 */

0x2f, /* FC_IP */
0x5a, /* FC_CONSTANT_IID */

/* 328 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 332 */ NdrFcShort( 0x0 ), /* 0 */
/* 334 */ NdrFcShort( 0x0 ), /* 0 */
/* 336 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 338 */ 0x0, /* 0 */
0x0, /* 0 */
/* 340 */ 0x0, /* 0 */
0x0, /* 0 */
/* 342 */ 0x0, /* 0 */
0x46, /* 70 */

/* 344 */

0x12, 0x10, /* FC_UP [pointer_deref] */
/* 346 */ NdrFcShort( 0x2 ), /* Offset= 2 (348) */
/* 348 */

0x12, 0x0, /* FC_UP */
/* 350 */ NdrFcShort( 0x1fc ), /* Offset= 508 (858) */
/* 352 */

0x2a, /* FC_ENCAPSULATED_UNION */
0x49, /* 73 */

/* 354 */ NdrFcShort( 0x18 ), /* 24 */
/* 356 */ NdrFcShort( 0xa ), /* 10 */
/* 358 */ NdrFcLong( 0x8 ), /* 8 */
/* 362 */ NdrFcShort( 0x58 ), /* Offset= 88 (450) */
/* 364 */ NdrFcLong( 0xd ), /* 13 */
/* 368 */ NdrFcShort( 0x78 ), /* Offset= 120 (488) */
/* 370 */ NdrFcLong( 0x9 ), /* 9 */
/* 374 */ NdrFcShort( 0x94 ), /* Offset= 148 (522) */
/* 376 */ NdrFcLong( 0xc ), /* 12 */
/* 380 */ NdrFcShort( 0xbc ), /* Offset= 188 (568) */

```



```

/* 382 */ NdrFcLong( 0x24 ), /* 36 */
/* 386 */ NdrFcShort( 0x114 ), /* Offset= 276 (662) */
/* 388 */ NdrFcLong( 0x800d ), /* 32781 */
/* 392 */ NdrFcShort( 0x130 ), /* Offset= 304 (696) */
/* 394 */ NdrFcLong( 0x10 ), /* 16 */
/* 398 */ NdrFcShort( 0x148 ), /* Offset= 328 (726) */
/* 400 */ NdrFcLong( 0x2 ), /* 2 */
/* 404 */ NdrFcShort( 0x160 ), /* Offset= 352 (756) */
/* 406 */ NdrFcLong( 0x3 ), /* 3 */
/* 410 */ NdrFcShort( 0x178 ), /* Offset= 376 (786) */
/* 412 */ NdrFcLong( 0x14 ), /* 20 */
/* 416 */ NdrFcShort( 0x190 ), /* Offset= 400 (816) */
/* 418 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (417) */
/* 420 */
                                0x1b, /* FC_CARRAY */
                                0x3, /* 3 */
/* 422 */ NdrFcShort( 0x4 ), /* 4 */
/* 424 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                                0x0, /* */
/* 426 */ NdrFcShort( 0x0 ), /* 0 */
/* 428 */
                                0x4b, /* FC_PP */
                                0x5c, /* FC_PAD */
/* 430 */
                                0x48, /* FC_VARIABLE_REPEAT */
                                0x49, /* FC_FIXED_OFFSET */
/* 432 */ NdrFcShort( 0x4 ), /* 4 */
/* 434 */ NdrFcShort( 0x0 ), /* 0 */
/* 436 */ NdrFcShort( 0x1 ), /* 1 */
/* 438 */ NdrFcShort( 0x0 ), /* 0 */
/* 440 */ NdrFcShort( 0x0 ), /* 0 */
/* 442 */ 0x12, 0x0, /* FC_UP */
/* 444 */ NdrFcShort( 0xffffffff6e ), /* Offset= -146 (298) */
/* 446 */
                                0x5b, /* FC_END */
/* 448 */ 0x5c,
                                0x8, /* FC_LONG */
                                0x5b, /* FC_PAD */
/* 450 */
                                0x16, /* FC_PSTRUCT */
                                0x3, /* 3 */
/* 452 */ NdrFcShort( 0x8 ), /* 8 */
/* 454 */
                                0x4b, /* FC_PP */
                                0x5c, /* FC_PAD */
/* 456 */
                                0x46, /* FC_NO_REPEAT */
                                0x5c, /* FC_PAD */
/* 458 */ NdrFcShort( 0x4 ), /* 4 */
/* 460 */ NdrFcShort( 0x4 ), /* 4 */
/* 462 */ 0x11, 0x0, /* FC_RP */
/* 464 */ NdrFcShort( 0xffffffffd4 ), /* Offset= -44 (420) */
/* 466 */
                                0x5b, /* FC_END */
/* 468 */ 0x8,
                                0x8, /* FC_LONG */
                                0x5b, /* FC_PAD */
/* 470 */
                                0x21, /* FC_BOGUS_ARRAY */
                                0x3, /* 3 */
/* 472 */ NdrFcShort( 0x0 ), /* 0 */

```

```

/* 474 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                                0x0, /* */
/* 476 */ NdrFcShort( 0x0 ), /* 0 */
/* 478 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 482 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
                                0x0, /* 0 */
/* 484 */ NdrFcShort( 0xffffffff50 ), /* Offset= -176 (308) */
/* 486 */ 0x5c, /* FC_PAD */
                                0x5b, /* FC_END */
/* 488 */
                                0x1a, /* FC_BOGUS_STRUCT */
                                0x3, /* 3 */
/* 490 */ NdrFcShort( 0x8 ), /* 8 */
/* 492 */ NdrFcShort( 0x0 ), /* 0 */
/* 494 */ NdrFcShort( 0x6 ), /* Offset= 6 (500) */
/* 496 */ 0x8, /* FC_LONG */
                                0x36, /* FC_POINTER */
/* 498 */ 0x5c, /* FC_PAD */
                                0x5b, /* FC_END */
/* 500 */
                                0x11, 0x0, /* FC_RP */
/* 502 */ NdrFcShort( 0xffffffffe0 ), /* Offset= -32 (470) */
/* 504 */
                                0x21, /* FC_BOGUS_ARRAY */
                                0x3, /* 3 */
/* 506 */ NdrFcShort( 0x0 ), /* 0 */
/* 508 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                                0x0, /* */
/* 510 */ NdrFcShort( 0x0 ), /* 0 */
/* 512 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 516 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
                                0x0, /* 0 */
/* 518 */ NdrFcShort( 0xffffffff40 ), /* Offset= -192 (326) */
/* 520 */ 0x5c, /* FC_PAD */
                                0x5b, /* FC_END */
/* 522 */
                                0x1a, /* FC_BOGUS_STRUCT */
                                0x3, /* 3 */
/* 524 */ NdrFcShort( 0x8 ), /* 8 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 530 */ 0x8, /* FC_LONG */
                                0x36, /* FC_POINTER */
/* 532 */ 0x5c, /* FC_PAD */
                                0x5b, /* FC_END */
/* 534 */
                                0x11, 0x0, /* FC_RP */
/* 536 */ NdrFcShort( 0xffffffffe0 ), /* Offset= -32 (504) */
/* 538 */
                                0x1b, /* FC_CARRAY */
                                0x3, /* 3 */
/* 540 */ NdrFcShort( 0x4 ), /* 4 */
/* 542 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                                0x0, /* */
/* 544 */ NdrFcShort( 0x0 ), /* 0 */
/* 546 */
                                0x4b, /* FC_PP */
                                0x5c, /* FC_PAD */
/* 548 */
                                0x48, /* FC_VARIABLE_REPEAT */
                                0x49, /* FC_FIXED_OFFSET */
/* 550 */ NdrFcShort( 0x4 ), /* 4 */
/* 552 */ NdrFcShort( 0x0 ), /* 0 */

```

```

/* 554 */ NdrFcShort( 0x1 ), /* 1 */
/* 556 */ NdrFcShort( 0x0 ), /* 0 */
/* 558 */ NdrFcShort( 0x0 ), /* 0 */
/* 560 */ 0x12, 0x0, /* FC_UP */
/* 562 */ NdrFcShort( 0x182 ), /* Offset= 386 (948) */
/* 564 */
                                0x5b, /* FC_END */
                                0x8, /* FC_LONG */
/* 566 */ 0x5c, /* FC_PAD */
                                0x5b, /* FC_END */
/* 568 */
                                0x1a, /* FC_BOGUS_STRUCT */
                                0x3, /* 3 */
/* 570 */ NdrFcShort( 0x8 ), /* 8 */
/* 572 */ NdrFcShort( 0x0 ), /* 0 */
/* 574 */ NdrFcShort( 0x6 ), /* Offset= 6 (580) */
/* 576 */ 0x8, /* FC_LONG */
/* 578 */ 0x5c, /* FC_PAD */
                                0x36, /* FC_POINTER */
                                0x5b, /* FC_END */
/* 580 */
                                0x11, 0x0, /* FC_RP */
/* 582 */ NdrFcShort( 0xfffffd4 ), /* Offset= -44 (538) */
/* 584 */
                                0x2f, /* FC_IP */
                                0x5a, /* FC_CONSTANT_IID */
/* 586 */ NdrFcLong( 0x2f ), /* 47 */
/* 590 */ NdrFcShort( 0x0 ), /* 0 */
/* 592 */ NdrFcShort( 0x0 ), /* 0 */
/* 594 */ 0xc0, /* 192 */
                                0x0, /* 0 */
/* 596 */ 0x0, /* 0 */
                                0x0, /* 0 */
/* 598 */ 0x0, /* 0 */
                                0x0, /* 0 */
/* 600 */ 0x0, /* 0 */
                                0x46, /* 70 */
/* 602 */
                                0x1b, /* FC_CARRAY */
                                0x0, /* 0 */
/* 604 */ NdrFcShort( 0x1 ), /* 1 */
/* 606 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                                0x0, /* */
/* 608 */ NdrFcShort( 0x4 ), /* 4 */
/* 610 */ 0x1, /* FC_BYTE */
                                0x5b, /* FC_END */
/* 612 */
                                0x1a, /* FC_BOGUS_STRUCT */
                                0x3, /* 3 */
/* 614 */ NdrFcShort( 0x10 ), /* 16 */
/* 616 */ NdrFcShort( 0x0 ), /* 0 */
/* 618 */ NdrFcShort( 0xa ), /* Offset= 10 (628) */
/* 620 */ 0x8, /* FC_LONG */
                                0x8, /* FC_LONG */
/* 622 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
                                0x0, /* 0 */
/* 624 */ NdrFcShort( 0xfffffd8 ), /* Offset= -40 (584) */
/* 626 */ 0x36, /* FC_POINTER */
                                0x5b, /* FC_END */
/* 628 */
                                0x12, 0x0, /* FC_UP */
/* 630 */ NdrFcShort( 0xfffffe4 ), /* Offset= -28 (602) */

```

```

/* 632 */
                                0x1b, /* FC_CARRAY */
                                0x3, /* 3 */
/* 634 */ NdrFcShort( 0x4 ), /* 4 */
/* 636 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                                0x0, /* */
/* 638 */ NdrFcShort( 0x0 ), /* 0 */
/* 640 */
                                0x4b, /* FC_PP */
                                0x5c, /* FC_PAD */
/* 642 */
                                0x48, /* FC_VARIABLE_REPEAT */
                                0x49, /* FC_FIXED_OFFSET */
/* 644 */ NdrFcShort( 0x4 ), /* 4 */
/* 646 */ NdrFcShort( 0x0 ), /* 0 */
/* 648 */ NdrFcShort( 0x1 ), /* 1 */
/* 650 */ NdrFcShort( 0x0 ), /* 0 */
/* 652 */ NdrFcShort( 0x0 ), /* 0 */
/* 654 */ 0x12, 0x0, /* FC_UP */
/* 656 */ NdrFcShort( 0xfffffd4 ), /* Offset= -44 (612) */
/* 658 */
                                0x5b, /* FC_END */
                                0x8, /* FC_LONG */
/* 660 */ 0x5c, /* FC_PAD */
                                0x5b, /* FC_END */
/* 662 */
                                0x1a, /* FC_BOGUS_STRUCT */
                                0x3, /* 3 */
/* 664 */ NdrFcShort( 0x8 ), /* 8 */
/* 666 */ NdrFcShort( 0x0 ), /* 0 */
/* 668 */ NdrFcShort( 0x6 ), /* Offset= 6 (674) */
/* 670 */ 0x8, /* FC_LONG */
                                0x36, /* FC_POINTER */
/* 672 */ 0x5c, /* FC_PAD */
                                0x5b, /* FC_END */
/* 674 */
                                0x11, 0x0, /* FC_RP */
/* 676 */ NdrFcShort( 0xfffffd4 ), /* Offset= -44 (632) */
/* 678 */
                                0x1d, /* FC_SMPARRAY */
                                0x0, /* 0 */
/* 680 */ NdrFcShort( 0x8 ), /* 8 */
/* 682 */ 0x2, /* FC_CHAR */
                                0x5b, /* FC_END */
/* 684 */
                                0x15, /* FC_STRUCT */
                                0x3, /* 3 */
/* 686 */ NdrFcShort( 0x10 ), /* 16 */
/* 688 */ 0x8, /* FC_LONG */
                                0x6, /* FC_SHORT */
/* 690 */ 0x6, /* FC_SHORT */
                                0x4c, /* FC_EMBEDDED_COMPLEX */
/* 692 */ 0x0, /* 0 */
                                NdrFcShort( 0xfffff1 ), /* Offset= -15 (678) */
                                0x5b, /* FC_END */
/* 696 */
                                0x1a, /* FC_BOGUS_STRUCT */
                                0x3, /* 3 */
/* 698 */ NdrFcShort( 0x18 ), /* 24 */
/* 700 */ NdrFcShort( 0x0 ), /* 0 */
/* 702 */ NdrFcShort( 0xa ), /* Offset= 10 (712) */
/* 704 */ 0x8, /* FC_LONG */

```

```

0x36, /* FC_POINTER */
/* 706 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 708 */ NdrFcShort( 0xffffffff8 ), /* Offset= -24 (684) */
/* 710 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 712 */
0x11, 0x0, /* FC_RP */
/* 714 */ NdrFcShort( 0xffffffff0c ), /* Offset= -244 (470) */
/* 716 */
0x1b, /* FC_CARRAY */
0x0, /* 0 */
/* 718 */ NdrFcShort( 0x1 ), /* 1 */
/* 720 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 722 */ NdrFcShort( 0x0 ), /* 0 */
/* 724 */ 0x1, /* FC_BYTE */
0x5b, /* FC_END */
/* 726 */
0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 728 */ NdrFcShort( 0x8 ), /* 8 */
/* 730 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 732 */
0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 734 */ NdrFcShort( 0x4 ), /* 4 */
/* 736 */ NdrFcShort( 0x4 ), /* 4 */
/* 738 */ 0x12, 0x0, /* FC_UP */
/* 740 */ NdrFcShort( 0xffffffff8 ), /* Offset= -24 (716) */
/* 742 */
0x5b, /* FC_END */
/* 744 */ 0x8, /* FC_LONG */
0x5b, /* FC_END */
/* 746 */
0x1b, /* FC_CARRAY */
0x1, /* 1 */
/* 748 */ NdrFcShort( 0x2 ), /* 2 */
/* 750 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 752 */ NdrFcShort( 0x0 ), /* 0 */
/* 754 */ 0x6, /* FC_SHORT */
0x5b, /* FC_END */
/* 756 */
0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 758 */ NdrFcShort( 0x8 ), /* 8 */
/* 760 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 762 */
0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 764 */ NdrFcShort( 0x4 ), /* 4 */
/* 766 */ NdrFcShort( 0x4 ), /* 4 */
/* 768 */ 0x12, 0x0, /* FC_UP */
/* 770 */ NdrFcShort( 0xffffffff8 ), /* Offset= -24 (746) */
/* 772 */
0x5b, /* FC_END */

```

```

0x8, /* FC_LONG */
/* 774 */ 0x8, /* FC_LONG */
0x5b, /* FC_END */
/* 776 */
0x1b, /* FC_CARRAY */
0x3, /* 3 */
/* 778 */ NdrFcShort( 0x4 ), /* 4 */
/* 780 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 782 */ NdrFcShort( 0x0 ), /* 0 */
/* 784 */ 0x8, /* FC_LONG */
0x5b, /* FC_END */
/* 786 */
0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 788 */ NdrFcShort( 0x8 ), /* 8 */
/* 790 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 792 */
0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 794 */ NdrFcShort( 0x4 ), /* 4 */
/* 796 */ NdrFcShort( 0x4 ), /* 4 */
/* 798 */ 0x12, 0x0, /* FC_UP */
/* 800 */ NdrFcShort( 0xffffffff8 ), /* Offset= -24 (776) */
/* 802 */
0x5b, /* FC_END */
/* 804 */ 0x8, /* FC_LONG */
0x5b, /* FC_END */
/* 806 */
0x1b, /* FC_CARRAY */
0x7, /* 7 */
/* 808 */ NdrFcShort( 0x8 ), /* 8 */
/* 810 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 812 */ NdrFcShort( 0x0 ), /* 0 */
/* 814 */ 0xb, /* FC_HYPER */
0x5b, /* FC_END */
/* 816 */
0x16, /* FC_PSTRUCT */
0x3, /* 3 */
/* 818 */ NdrFcShort( 0x8 ), /* 8 */
/* 820 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 822 */
0x46, /* FC_NO_REPEAT */
0x5c, /* FC_PAD */
/* 824 */ NdrFcShort( 0x4 ), /* 4 */
/* 826 */ NdrFcShort( 0x4 ), /* 4 */
/* 828 */ 0x12, 0x0, /* FC_UP */
/* 830 */ NdrFcShort( 0xffffffff8 ), /* Offset= -24 (806) */
/* 832 */
0x5b, /* FC_END */
/* 834 */ 0x8, /* FC_LONG */
0x5b, /* FC_END */
/* 836 */

```

```

0x15, /* FC_STRUCT */
/* 838 */ NdrFcShort( 0x8 ), /* 8 */
/* 840 */ 0x8, /* FC_LONG */
/* 842 */ 0x5c, /* FC_PAD */
/* 844 */ 0x5b, /* FC_END */
0x1b, /* FC_CARRAY */
/* 846 */ NdrFcShort( 0x8 ), /* 8 */
/* 848 */ 0x7, /* Corr desc: FC_USHORT */
/* 850 */ NdrFcShort( 0xfffd8 ), /* -40 */
/* 852 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 854 */ NdrFcShort( 0xfffffee ), /* Offset= -18 (836) */
/* 856 */ 0x5c, /* FC_PAD */
/* 858 */ 0x5b, /* FC_END */
0x1a, /* FC_BOGUS_STRUCT */
/* 860 */ NdrFcShort( 0x28 ), /* 40 */
/* 862 */ NdrFcShort( 0xfffffee ), /* Offset= -18 (844) */
/* 864 */ NdrFcShort( 0x0 ), /* Offset= 0 (864) */
/* 866 */ 0x6, /* FC_SHORT */
/* 868 */ 0x38, /* FC_ALIGNM4 */
/* 870 */ 0x8, /* FC_LONG */
/* 872 */ 0x0, /* FC_EMBEDDED_COMPLEX */
/* 874 */ NdrFcShort( 0xfffffd7 ), /* Offset= -521 (352) */
/* 876 */ 0x5b, /* FC_END */
0x12, 0x0, /* FC_UP */
/* 878 */ NdrFcShort( 0xfffffe6 ), /* Offset= -266 (612) */
/* 880 */ 0x12, 0x8, /* FC_UP [simple_pointer] */
/* 882 */ 0x5c, /* FC_BYTE */
/* 884 */ 0x5c, /* FC_PAD */
0x12, 0x8, /* FC_UP [simple_pointer] */
/* 886 */ 0x6, /* FC_SHORT */
/* 888 */ 0x5c, /* FC_PAD */
0x12, 0x8, /* FC_UP [simple_pointer] */
/* 890 */ 0x8, /* FC_LONG */
/* 892 */ 0x5c, /* FC_PAD */
0x12, 0x8, /* FC_UP [simple_pointer] */
/* 894 */ 0xa, /* FC_FLOAT */
/* 896 */ 0x5c, /* FC_PAD */
0x12, 0x8, /* FC_UP [simple_pointer] */
/* 898 */ 0xc, /* FC_DOUBLE */
/* 900 */ 0x5c, /* FC_PAD */
0x12, 0x0, /* FC_UP */
/* 902 */ NdrFcShort( 0xfffffd90 ), /* Offset= -624 (278) */
/* 904 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 906 */ NdrFcShort( 0xfffffd92 ), /* Offset= -622 (284) */

```

```

/* 908 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 910 */ NdrFcShort( 0xfffffda6 ), /* Offset= -602 (308) */
/* 912 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 914 */ NdrFcShort( 0xfffffdb4 ), /* Offset= -588 (326) */
/* 916 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 918 */ NdrFcShort( 0xfffffdc2 ), /* Offset= -574 (344) */
/* 920 */ 0x12, 0x10, /* FC_UP [pointer_deref] */
/* 922 */ NdrFcShort( 0x2 ), /* Offset= 2 (924) */
/* 924 */ 0x12, 0x0, /* FC_UP */
/* 926 */ NdrFcShort( 0x16 ), /* Offset= 22 (948) */
/* 928 */ 0x15, /* FC_STRUCT */
/* 930 */ NdrFcShort( 0x10 ), /* 16 */
/* 932 */ 0x6, /* FC_SHORT */
/* 934 */ 0x1, /* FC_BYTE */
/* 936 */ 0x8, /* FC_BYTE */
/* 938 */ 0xb, /* FC_ALIGNM4 */
/* 940 */ 0x38, /* FC_LONG */
/* 942 */ 0x39, /* FC_ALIGNM8 */
/* 944 */ 0x5b, /* FC_HYPER */
/* 946 */ 0x12, 0x0, /* FC_UP */
/* 948 */ NdrFcShort( 0xfffffff2 ), /* Offset= -14 (928) */
0x12, 0x8, /* FC_UP [simple_pointer] */
/* 950 */ 0x2, /* FC_CHAR */
/* 952 */ 0x5c, /* FC_PAD */
/* 954 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 956 */ 0x7, /* 7 */
/* 958 */ NdrFcShort( 0x20 ), /* 32 */
/* 960 */ NdrFcShort( 0x0 ), /* 0 */
/* 962 */ NdrFcShort( 0x0 ), /* Offset= 0 (954) */
/* 964 */ 0x8, /* FC_LONG */
/* 966 */ 0x6, /* FC_SHORT */
/* 968 */ 0x6, /* FC_SHORT */
/* 970 */ 0x6, /* FC_SHORT */
/* 972 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 974 */ NdrFcShort( 0xfffffc42 ), /* Offset= -958 (6) */
/* 976 */ 0x5c, /* FC_PAD */
/* 978 */ 0x5b, /* FC_END */
/* 980 */ 0xb4, /* FC_USER_MARSHAL */
/* 982 */ 0x83, /* 131 */
/* 984 */ NdrFcShort( 0x0 ), /* 0 */
/* 986 */ NdrFcShort( 0x10 ), /* 16 */
/* 988 */ NdrFcShort( 0x0 ), /* 0 */
/* 990 */ NdrFcShort( 0xfffffc32 ), /* Offset= -974 (2) */
/* 992 */ 0x11, 0x4, /* FC_UP [simple_pointer] */
/* 994 */ NdrFcShort( 0x6 ), /* Offset= 6 (986) */
0x13, 0x0, /* FC_OP */
/* 996 */ NdrFcShort( 0xfffffddc ), /* Offset= -36 (948) */

```

```

/* 986 */ 0xb4,          /* FC_USER_MARSHAL */
                                0x83,          /* 131 */
/* 988 */ NdrPcShort( 0x0 ), /* 0 */
/* 990 */ NdrPcShort( 0x10 ), /* 16 */
/* 992 */ NdrPcShort( 0x0 ), /* 0 */
/* 994 */ NdrPcShort( 0xffffffff ), /* Offset= -12 (982) */

                                0x0

    }
};

const CInterfaceProxyVtbl * _tpcc_com_ps_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
    0
};

const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[] =
{
    ( CInterfaceStubVtbl *) &_ITPCCStubVtbl,
    0
};

PCInterfaceName const _tpcc_com_ps_InterfaceNamesList[] =
{
    "ITPCC",
    0
};

#define _tpcc_com_ps_CHECK_IID(n)      IID_GENERIC_CHECK_IID( _tpcc_com_ps, pIID,
n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{
    if(!_tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }

    return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) & _tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) & _tpcc_com_ps_StubVtblList,
    (const PCInterfaceName *) & _tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    & _tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0 /* Filler3 */
};

#endif /* !defined(_M_IA64) && !defined(_M_AXP64) */

```

```

#pragma warning( disable: 4049 ) /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the proxy stub code */

/* File created by MIDL compiler version 5.03.0280 */
/* at Mon Jun 12 18:15:12 2000
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), Wl, Zp8, env=Win64 (32b run,appending), ms_ext, c_ext, robust
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
        DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if defined(_M_IA64) || defined(_M_AXP64)
#define USE_STUBLESS_PROXY

/* verify that the <rpcproxy.h> version is high enough to compile this file*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 475
#endif

#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of <rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 979
#define PROC_FORMAT_STRING_SIZE 253
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 1

typedef struct _MIDL_TYPE_FORMAT_STRING
{
    short          Pad;
    unsigned char  Format[ TYPE_FORMAT_STRING_SIZE ];
} MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
{
    short          Pad;
    unsigned char  Format[ PROC_FORMAT_STRING_SIZE ];
} MIDL_PROC_FORMAT_STRING;

extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,
    GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,

```

```

GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[] =
{
    0,
    44,
    88,
    132,
    176,
    220
};

static const MIDL_SERVER_INFO ITPCC_ServerInfo =
{
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0
};

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo =
{
    &Object_StubDesc,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0
};

CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    (void *)-1 /* ITPCC::NewOrder */ ,
    (void *)-1 /* ITPCC::Payment */ ,
    (void *)-1 /* ITPCC::Delivery */ ,
    (void *)-1 /* ITPCC::StockLevel */ ,
    (void *)-1 /* ITPCC::OrderStatus */ ,
    (void *)-1 /* ITPCC::CallSetComplete */
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,

```

```

    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

extern const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ];

static const MIDL_STUB_DESC Object_StubDesc =
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x50002, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.3.280 */
    0,
    UserMarshalRoutines,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* Reserved3 */
    0, /* Reserved4 */
    0 /* Reserved5 */
};

#pragma data_seg(".rdata")

static const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ] =
{
    {
        VARIANT_UserSize
        ,VARIANT_UserMarshal
        ,VARIANT_UserUnmarshal
        ,VARIANT_UserFree
    }
};

#if !defined(__RPC_WIN64__)
#error Invalid build platform for this stub.
#endif

static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
{
    0,
    {
        /* Procedure NewOrder */

                                0x33,          /* FC_AUTO_HANDLE */
                                0x6c,          /* Old Flags: object, Oi2 */
/* 2 */ NdrFcLong( 0x0 ), /* 0 */

```

```

/* 6 */ NdrFcShort( 0x3 ), /* 3 */
#ifdef ALPHA
/* 8 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 10 */ NdrFcShort( 0x0 ), /* 0 */
/* 12 */ NdrFcShort( 0x8 ), /* 8 */
/* 14 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
0x3, /* 3 */
/* 16 */ 0xa, /* 10 */
0x7, /* Ext Flags: new corr desc, clt
corr check, srv corr check, */
/* 18 */ NdrFcShort( 0x20 ), /* 32 */
/* 20 */ NdrFcShort( 0x20 ), /* 32 */
/* 22 */ NdrFcShort( 0x0 ), /* 0 */
/* 24 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 26 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef ALPHA
/* 28 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
NdrFcShort( 0x8 ), /* axp64 Stack size/offset = 8 */
#endif
/* 30 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

/* Parameter txn_out */

/* 32 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=24 */
#ifdef ALPHA
/* 34 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 36 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Return value */

/* 38 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef ALPHA
/* 40 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 42 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure Payment */

/* 44 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 46 */ NdrFcLong( 0x0 ), /* 0 */
/* 50 */ NdrFcShort( 0x4 ), /* 4 */
#ifdef ALPHA
/* 52 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 54 */ NdrFcShort( 0x0 ), /* 0 */

```

```

/* 56 */ NdrFcShort( 0x8 ), /* 8 */
/* 58 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
0x3, /* 3 */
/* 60 */ 0xa, /* 10 */
0x7, /* Ext Flags: new corr desc, clt
corr check, srv corr check, */
/* 62 */ NdrFcShort( 0x20 ), /* 32 */
/* 64 */ NdrFcShort( 0x20 ), /* 32 */
/* 66 */ NdrFcShort( 0x0 ), /* 0 */
/* 68 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 70 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef ALPHA
/* 72 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
NdrFcShort( 0x8 ), /* axp64 Stack size/offset = 8 */
#endif
/* 74 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

/* Parameter txn_out */

/* 76 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=24 */
#ifdef ALPHA
/* 78 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 80 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

/* Return value */

/* 82 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef ALPHA
/* 84 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 86 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure Delivery */

/* 88 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 90 */ NdrFcLong( 0x0 ), /* 0 */
/* 94 */ NdrFcShort( 0x5 ), /* 5 */
#ifdef ALPHA
/* 96 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 98 */ NdrFcShort( 0x0 ), /* 0 */
/* 100 */ NdrFcShort( 0x8 ), /* 8 */
/* 102 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
0x3, /* 3 */
/* 104 */ 0xa, /* 10 */
0x7, /* Ext Flags: new corr desc, clt
corr check, srv corr check, */

```

```

/* 106 */ NdrFcShort( 0x20 ), /* 32 */
/* 108 */ NdrFcShort( 0x20 ), /* 32 */
/* 110 */ NdrFcShort( 0x0 ), /* 0 */
/* 112 */ NdrFcShort( 0x0 ), /* 0 */

    /* Parameter txn_in */

/* 114 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef _ALPHA_
/* 116 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
    NdrFcShort( 0x8 ), /* axp64 Stack size/offset = 8 */
#endif
/* 118 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

    /* Parameter txn_out */

/* 120 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=24 */
#ifdef _ALPHA_
/* 122 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
    NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 124 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

    /* Return value */

/* 126 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA_
/* 128 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
    NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 130 */ 0x8, /* FC_LONG */
0x0, /* 0 */

    /* Procedure StockLevel */

/* 132 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 134 */ NdrFcLong( 0x0 ), /* 0 */
/* 138 */ NdrFcShort( 0x6 ), /* 6 */
#ifdef _ALPHA_
/* 140 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
    NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 142 */ NdrFcShort( 0x0 ), /* 0 */
/* 144 */ NdrFcShort( 0x8 ), /* 8 */
/* 146 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
0x3, /* 3 */
/* 148 */ 0xa, /* 10 */
0x7, /* Ext Flags: new corr desc, clt
corr check, srv corr check, */
/* 150 */ NdrFcShort( 0x20 ), /* 32 */
/* 152 */ NdrFcShort( 0x20 ), /* 32 */
/* 154 */ NdrFcShort( 0x0 ), /* 0 */
/* 156 */ NdrFcShort( 0x0 ), /* 0 */

    /* Parameter txn_in */

```

```

/* 158 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef _ALPHA_
/* 160 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
    NdrFcShort( 0x8 ), /* axp64 Stack size/offset = 8 */
#endif
/* 162 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

    /* Parameter txn_out */

/* 164 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=24 */
#ifdef _ALPHA_
/* 166 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
    NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 168 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

    /* Return value */

/* 170 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef _ALPHA_
/* 172 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
    NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 174 */ 0x8, /* FC_LONG */
0x0, /* 0 */

    /* Procedure OrderStatus */

/* 176 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 178 */ NdrFcLong( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x7 ), /* 7 */
#ifdef _ALPHA_
/* 184 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
    NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 186 */ NdrFcShort( 0x0 ), /* 0 */
/* 188 */ NdrFcShort( 0x8 ), /* 8 */
/* 190 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has
return, has ext, */
0x3, /* 3 */
/* 192 */ 0xa, /* 10 */
0x7, /* Ext Flags: new corr desc, clt
corr check, srv corr check, */
/* 194 */ NdrFcShort( 0x20 ), /* 32 */
/* 196 */ NdrFcShort( 0x20 ), /* 32 */
/* 198 */ NdrFcShort( 0x0 ), /* 0 */
/* 200 */ NdrFcShort( 0x0 ), /* 0 */

    /* Parameter txn_in */

/* 202 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
#ifdef _ALPHA_
/* 204 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
    NdrFcShort( 0x8 ), /* axp64 Stack size/offset = 8 */
#endif
/* 206 */ NdrFcShort( 0x3b6 ), /* Type Offset=950 */

```



```

        /* Parameter txn_out */

/* 208 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple
ref, srv alloc size=24 */
#ifdef ALPHA
/* 210 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
        NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 212 */ NdrFcShort( 0x3c8 ), /* Type Offset=968 */

    /* Return value */

/* 214 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
#ifdef ALPHA
/* 216 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
        NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 218 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

    /* Procedure CallSetComplete */

/* 220 */ 0x33, /* FC_AUTO_HANDLE */
        0x6c, /* Old Flags: object, Oi2 */
/* 222 */ NdrFcLong( 0x0 ), /* 0 */
/* 226 */ NdrFcShort( 0x8 ), /* 8 */
/* 228 */ NdrFcShort( 0x10 ), /* ia64, axp64 Stack size/offset = 16 */
/* 230 */ NdrFcShort( 0x0 ), /* 0 */
/* 232 */ NdrFcShort( 0x8 ), /* 8 */
/* 234 */ 0x44, /* Oi2 Flags: has return, has ext, */
        0x1, /* 1 */
/* 236 */ 0xa, /* 10 */
        0x1, /* Ext Flags: new corr desc, */
/* 238 */ NdrFcShort( 0x0 ), /* 0 */
/* 240 */ NdrFcShort( 0x0 ), /* 0 */
/* 242 */ NdrFcShort( 0x0 ), /* 0 */
/* 244 */ NdrFcShort( 0x0 ), /* 0 */

    /* Return value */

/* 246 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 248 */ NdrFcShort( 0x8 ), /* ia64, axp64 Stack size/offset = 8 */
/* 250 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

        0x0
    }
};

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
    0,
    {
        NdrFcShort( 0x0 ), /* 0 */

/* 2 */
        0x12, 0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0x39e ), /* Offset= 926 (930) */
/* 6 */
        0x2b, /* FC_NON_ENCAPSULATED_UNION */
        0x9, /* FC_ULONG */
    }
};

```

```

/* 8 */ 0x7, /* Corr desc: FC_USHORT */
        0x0, /* */
/* 10 */ NdrFcShort( 0xffff8 ), /* -8 */
/* 12 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 14 */ NdrFcShort( 0x2 ), /* Offset= 2 (16) */
/* 16 */ NdrFcShort( 0x10 ), /* 16 */
/* 18 */ NdrFcShort( 0x2b ), /* 43 */
/* 20 */ NdrFcLong( 0x3 ), /* 3 */
/* 24 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 26 */ NdrFcLong( 0x11 ), /* 17 */
/* 30 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE */
/* 32 */ NdrFcLong( 0x2 ), /* 2 */
/* 36 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 38 */ NdrFcLong( 0x4 ), /* 4 */
/* 42 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT */
/* 44 */ NdrFcLong( 0x5 ), /* 5 */
/* 48 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 50 */ NdrFcLong( 0xb ), /* 11 */
/* 54 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 56 */ NdrFcLong( 0xa ), /* 10 */
/* 60 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 62 */ NdrFcLong( 0x6 ), /* 6 */
/* 66 */ NdrFcShort( 0xd6 ), /* Offset= 214 (280) */
/* 68 */ NdrFcLong( 0x7 ), /* 7 */
/* 72 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 74 */ NdrFcLong( 0x8 ), /* 8 */
/* 78 */ NdrFcShort( 0xd0 ), /* Offset= 208 (286) */
/* 80 */ NdrFcLong( 0xd ), /* 13 */
/* 84 */ NdrFcShort( 0xe4 ), /* Offset= 228 (312) */
/* 86 */ NdrFcLong( 0x9 ), /* 9 */
/* 90 */ NdrFcShort( 0xf0 ), /* Offset= 240 (330) */
/* 92 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 96 */ NdrFcShort( 0xfc ), /* Offset= 252 (348) */
/* 98 */ NdrFcLong( 0x24 ), /* 36 */
/* 102 */ NdrFcShort( 0x2f4 ), /* Offset= 756 (858) */
/* 104 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 108 */ NdrFcShort( 0x2ee ), /* Offset= 750 (858) */
/* 110 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 114 */ NdrFcShort( 0x2ec ), /* Offset= 748 (862) */
/* 116 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 120 */ NdrFcShort( 0x2ea ), /* Offset= 746 (866) */
/* 122 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 126 */ NdrFcShort( 0x2e8 ), /* Offset= 744 (870) */
/* 128 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 132 */ NdrFcShort( 0x2e6 ), /* Offset= 742 (874) */
/* 134 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 138 */ NdrFcShort( 0x2e4 ), /* Offset= 740 (878) */
/* 140 */ NdrFcLong( 0x400b ), /* 16395 */
/* 144 */ NdrFcShort( 0x2d2 ), /* Offset= 722 (866) */
/* 146 */ NdrFcLong( 0x400a ), /* 16394 */
/* 150 */ NdrFcShort( 0x2d0 ), /* Offset= 720 (870) */
/* 152 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 156 */ NdrFcShort( 0x2d6 ), /* Offset= 726 (882) */
/* 158 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 162 */ NdrFcShort( 0x2cc ), /* Offset= 716 (878) */
/* 164 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 168 */ NdrFcShort( 0x2ce ), /* Offset= 718 (886) */
/* 170 */ NdrFcLong( 0x400d ), /* 16397 */
/* 174 */ NdrFcShort( 0x2cc ), /* Offset= 716 (890) */
/* 176 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 180 */ NdrFcShort( 0x2ca ), /* Offset= 714 (894) */
/* 182 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 186 */ NdrFcShort( 0x2c8 ), /* Offset= 712 (898) */

```

```

/* 188 */ NdrFcLong( 0x400c ), /* 16396 */
/* 192 */ NdrFcShort( 0x2c6 ), /* Offset= 710 (902) */
/* 194 */ NdrFcLong( 0x10 ), /* 16 */
/* 198 */ NdrFcShort( 0x8002 ), /* Simple arm type: FC_CHAR */
/* 200 */ NdrFcLong( 0x12 ), /* 18 */
/* 204 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 206 */ NdrFcLong( 0x13 ), /* 19 */
/* 210 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 212 */ NdrFcLong( 0x16 ), /* 22 */
/* 216 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 218 */ NdrFcLong( 0x17 ), /* 23 */
/* 222 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 224 */ NdrFcLong( 0xe ), /* 14 */
/* 228 */ NdrFcShort( 0x2aa ), /* Offset= 682 (910) */
/* 230 */ NdrFcLong( 0x400e ), /* 16398 */
/* 234 */ NdrFcShort( 0x2b0 ), /* Offset= 688 (922) */
/* 236 */ NdrFcLong( 0x4010 ), /* 16400 */
/* 240 */ NdrFcShort( 0x2ae ), /* Offset= 686 (926) */
/* 242 */ NdrFcLong( 0x4012 ), /* 16402 */
/* 246 */ NdrFcShort( 0x26c ), /* Offset= 620 (866) */
/* 248 */ NdrFcLong( 0x4013 ), /* 16403 */
/* 252 */ NdrFcShort( 0x26a ), /* Offset= 618 (870) */
/* 254 */ NdrFcLong( 0x4016 ), /* 16406 */
/* 258 */ NdrFcShort( 0x264 ), /* Offset= 612 (870) */
/* 260 */ NdrFcLong( 0x4017 ), /* 16407 */
/* 264 */ NdrFcShort( 0x25e ), /* Offset= 606 (870) */
/* 266 */ NdrFcLong( 0x0 ), /* 0 */
/* 270 */ NdrFcShort( 0x0 ), /* Offset= 0 (270) */
/* 272 */ NdrFcLong( 0x1 ), /* 1 */
/* 276 */ NdrFcShort( 0x0 ), /* Offset= 0 (276) */
/* 278 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (277) */
/* 280 */

0x15, /* FC_STRUCT */
0x7, /* 7 */
/* 282 */ NdrFcShort( 0x8 ), /* 8 */
/* 284 */ 0xb, /* FC_HYPER */
0x5b, /* FC_END */
/* 286 */

0x12, 0x0, /* FC_UP */
/* 288 */ NdrFcShort( 0xe ), /* Offset= 14 (302) */
/* 290 */

0x1b, /* FC_CARRAY */
0x1, /* 1 */
/* 292 */ NdrFcShort( 0x2 ), /* 2 */
/* 294 */ 0x9, /* Corr desc: FC_ULONG */
0x0, /* */
/* 296 */ NdrFcShort( 0xffffc ), /* -4 */
/* 298 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 300 */ 0x6, /* FC_SHORT */
0x5b, /* FC_END */
/* 302 */

0x17, /* FC_CSTRUCT */
0x3, /* 3 */
/* 304 */ NdrFcShort( 0x8 ), /* 8 */
/* 306 */ NdrFcShort( 0xffffffff0 ), /* Offset= -16 (290) */
/* 308 */ 0x8, /* FC_LONG */
0x8, /* FC_LONG */
/* 310 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 312 */

0x2f, /* FC_IP */
0x5a, /* FC_CONSTANT_IID */
/* 314 */ NdrFcLong( 0x0 ), /* 0 */

```

```

/* 318 */ NdrFcShort( 0x0 ), /* 0 */
/* 320 */ NdrFcShort( 0x0 ), /* 0 */
/* 322 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 324 */ 0x0, /* 0 */
0x0, /* 0 */
/* 326 */ 0x0, /* 0 */
0x0, /* 0 */
/* 328 */ 0x0, /* 0 */
0x46, /* 70 */
/* 330 */

0x2f, /* FC_IP */
0x5a, /* FC_CONSTANT_IID */
/* 332 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 336 */ NdrFcShort( 0x0 ), /* 0 */
/* 338 */ NdrFcShort( 0x0 ), /* 0 */
/* 340 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 342 */ 0x0, /* 0 */
0x0, /* 0 */
/* 344 */ 0x0, /* 0 */
0x0, /* 0 */
/* 346 */ 0x0, /* 0 */
0x46, /* 70 */
/* 348 */

0x12, 0x10, /* FC_UP [pointer_deref] */
/* 350 */ NdrFcShort( 0x2 ), /* Offset= 2 (352) */
/* 352 */

0x12, 0x0, /* FC_UP */
/* 354 */ NdrFcShort( 0x1e6 ), /* Offset= 486 (840) */
/* 356 */

0x2a, /* FC_ENCAPSULATED_UNION */
0x89, /* 137 */
/* 358 */ NdrFcShort( 0x20 ), /* 32 */
/* 360 */ NdrFcShort( 0xa ), /* 10 */
/* 362 */ NdrFcLong( 0x8 ), /* 8 */
/* 366 */ NdrFcShort( 0x50 ), /* Offset= 80 (446) */
/* 368 */ NdrFcLong( 0xd ), /* 13 */
/* 372 */ NdrFcShort( 0x70 ), /* Offset= 112 (484) */
/* 374 */ NdrFcLong( 0x9 ), /* 9 */
/* 378 */ NdrFcShort( 0x90 ), /* Offset= 144 (522) */
/* 380 */ NdrFcLong( 0xc ), /* 12 */
/* 384 */ NdrFcShort( 0xb0 ), /* Offset= 176 (560) */
/* 386 */ NdrFcLong( 0x24 ), /* 36 */
/* 390 */ NdrFcShort( 0x104 ), /* Offset= 260 (650) */
/* 392 */ NdrFcLong( 0x800d ), /* 32781 */
/* 396 */ NdrFcShort( 0x120 ), /* Offset= 288 (684) */
/* 398 */ NdrFcLong( 0x10 ), /* 16 */
/* 402 */ NdrFcShort( 0x13a ), /* Offset= 314 (716) */
/* 404 */ NdrFcLong( 0x2 ), /* 2 */
/* 408 */ NdrFcShort( 0x150 ), /* Offset= 336 (744) */
/* 410 */ NdrFcLong( 0x3 ), /* 3 */
/* 414 */ NdrFcShort( 0x166 ), /* Offset= 358 (772) */
/* 416 */ NdrFcLong( 0x14 ), /* 20 */
/* 420 */ NdrFcShort( 0x17c ), /* Offset= 380 (800) */
/* 422 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (421) */
/* 424 */

0x21, /* FC_BOGUS_ARRAY */
0x3, /* 3 */
/* 426 */ NdrFcShort( 0x0 ), /* 0 */
/* 428 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 430 */ NdrFcShort( 0x0 ), /* 0 */

```

```

/* 432 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 434 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 438 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 440 */
/* 442 */ NdrFcShort( 0x12, 0x0, /* FC_UP */
/* 444 */ 0x5c, /* Offset=-140 (302) */
/* 446 */ 0x5b, /* FC_PAD */
/* 448 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 450 */ 0x3, /* 3 */
/* 452 */ NdrFcShort( 0x10 ), /* 16 */
/* 454 */ NdrFcShort( 0x0 ), /* 0 */
/* 456 */ NdrFcShort( 0x6 ), /* Offset= 6 (458) */
/* 458 */ 0x8, /* FC_LONG */
/* 460 */ 0x39, /* FC_ALIGNM8 */
/* 462 */ 0x5b, /* FC_POINTER */
/* 464 */ 0x11, 0x0, /* FC_RP */
/* 466 */ NdrFcShort( 0xffffffffdc ), /* Offset=-36 (424) */
/* 468 */ 0x21, /* FC_BOGUS_ARRAY */
/* 470 */ 0x3, /* 3 */
/* 472 */ NdrFcShort( 0x0 ), /* 0 */
/* 474 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 476 */ 0x0, /* */
/* 478 */ NdrFcShort( 0x0 ), /* 0 */
/* 480 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 482 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 484 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 486 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 488 */ 0x0, /* 0 */
/* 490 */ NdrFcShort( 0xfffff58 ), /* Offset=-168 (312) */
/* 492 */ 0x5c, /* FC_PAD */
/* 494 */ 0x5b, /* FC_END */
/* 496 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 498 */ 0x3, /* 3 */
/* 500 */ NdrFcShort( 0x10 ), /* 16 */
/* 502 */ NdrFcShort( 0x0 ), /* 0 */
/* 504 */ NdrFcShort( 0x6 ), /* Offset= 6 (496) */
/* 506 */ 0x8, /* FC_LONG */
/* 508 */ 0x39, /* FC_ALIGNM8 */
/* 510 */ 0x5b, /* FC_POINTER */
/* 512 */ 0x11, 0x0, /* FC_RP */
/* 514 */ NdrFcShort( 0xffffffffdc ), /* Offset=-36 (462) */
/* 516 */ 0x21, /* FC_BOGUS_ARRAY */
/* 518 */ 0x3, /* 3 */
/* 520 */ NdrFcShort( 0x0 ), /* 0 */
/* 522 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 524 */ 0x0, /* */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 530 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 532 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 534 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 536 */ 0x0, /* 0 */
/* 538 */ NdrFcShort( 0xfffff44 ), /* Offset=-188 (330) */
/* 540 */ 0x5c, /* FC_PAD */

```

```

/* 522 */ 0x5b, /* FC_END */
/* 524 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 526 */ 0x3, /* 3 */
/* 528 */ NdrFcShort( 0x10 ), /* 16 */
/* 530 */ NdrFcShort( 0x0 ), /* 0 */
/* 532 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 534 */ 0x8, /* FC_LONG */
/* 536 */ 0x39, /* FC_ALIGNM8 */
/* 538 */ 0x5b, /* FC_POINTER */
/* 540 */ 0x11, 0x0, /* FC_RP */
/* 542 */ NdrFcShort( 0xffffffffdc ), /* Offset=-36 (500) */
/* 544 */ 0x21, /* FC_BOGUS_ARRAY */
/* 546 */ 0x3, /* 3 */
/* 548 */ NdrFcShort( 0x0 ), /* 0 */
/* 550 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 552 */ 0x0, /* */
/* 554 */ NdrFcShort( 0x0 ), /* 0 */
/* 556 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 558 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 560 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 562 */ 0x12, 0x0, /* FC_UP */
/* 564 */ NdrFcShort( 0x176 ), /* Offset= 374 (930) */
/* 566 */ 0x5c, /* FC_PAD */
/* 568 */ 0x5b, /* FC_END */
/* 570 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 572 */ 0x3, /* 3 */
/* 574 */ NdrFcShort( 0x10 ), /* 16 */
/* 576 */ NdrFcShort( 0x0 ), /* 0 */
/* 578 */ NdrFcShort( 0x6 ), /* Offset= 6 (572) */
/* 580 */ 0x8, /* FC_LONG */
/* 582 */ 0x39, /* FC_ALIGNM8 */
/* 584 */ 0x5b, /* FC_POINTER */
/* 586 */ 0x11, 0x0, /* FC_RP */
/* 588 */ NdrFcShort( 0xffffffffdc ), /* Offset=-36 (538) */
/* 590 */ 0x2f, /* FC_IP */
/* 592 */ 0x5a, /* FC_CONSTANT_IID */
/* 594 */ NdrFcLong( 0x2f ), /* 47 */
/* 596 */ NdrFcShort( 0x0 ), /* 0 */
/* 598 */ NdrFcShort( 0x0 ), /* 0 */
/* 600 */ 0xc0, /* 192 */
/* 602 */ 0x0, /* 0 */
/* 604 */ 0x0, /* 0 */
/* 606 */ 0x0, /* 0 */
/* 608 */ 0x0, /* 0 */
/* 610 */ 0x0, /* 0 */
/* 612 */ 0x0, /* 0 */
/* 614 */ 0x46, /* 70 */
/* 616 */ 0x1b, /* FC_CARRY */
/* 618 */ 0x0, /* 0 */
/* 620 */ NdrFcShort( 0x1 ), /* 1 */
/* 622 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 624 */ 0x0, /* */
/* 626 */ NdrFcShort( 0x4 ), /* 4 */

```

```

/* 602 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 604 */ 0x1, /* FC_BYTE */
/* 606 */ 0x5b, /* FC_END */

/* 608 */ NdrFcShort( 0x1a ), /* FC_BOGUS_STRUCT */
/* 610 */ 0x3, /* 3 */
/* 612 */ NdrFcShort( 0x18 ), /* 24 */
/* 614 */ 0x0, /* 0 */
/* 616 */ NdrFcShort( 0xc ), /* Offset= 12 (624) */
/* 618 */ 0x8, /* FC_LONG */
/* 620 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 622 */ 0x0, /* 0 */
/* 624 */ NdrFcShort( 0xfffffd6 ), /* Offset= -42 (576) */
/* 626 */ 0x39, /* FC_ALIGNM8 */
/* 628 */ 0x36, /* FC_POINTER */
/* 630 */ 0x5c, /* FC_PAD */
/* 632 */ 0x5b, /* FC_END */
/* 634 */ 0x12, 0x0, /* FC_UP */
/* 636 */ NdrFcShort( 0xfffffe0 ), /* Offset= -32 (594) */
/* 638 */ 0x21, /* FC_BOGUS_ARRAY */
/* 640 */ 0x3, /* 3 */
/* 642 */ NdrFcShort( 0x0 ), /* 0 */
/* 644 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 646 */ 0x0, /* */
/* 648 */ NdrFcShort( 0x0 ), /* 0 */
/* 650 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 652 */ NdrFcLong( 0xfffffff ), /* -1 */
/* 654 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 656 */ 0x12, 0x0, /* FC_UP */
/* 658 */ NdrFcShort( 0xfffffd8 ), /* Offset= -40 (606) */
/* 660 */ 0x5c, /* FC_PAD */
/* 662 */ 0x5b, /* FC_END */
/* 664 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 666 */ 0x3, /* 3 */
/* 668 */ NdrFcShort( 0x10 ), /* 16 */
/* 670 */ NdrFcShort( 0x0 ), /* 0 */
/* 672 */ NdrFcShort( 0x6 ), /* Offset= 6 (662) */
/* 674 */ 0x8, /* FC_LONG */
/* 676 */ 0x39, /* FC_ALIGNM8 */
/* 678 */ 0x5b, /* FC_POINTER */
/* 680 */ 0x5b, /* FC_END */
/* 682 */ 0x11, 0x0, /* FC_UP */
/* 684 */ NdrFcShort( 0xfffffddc ), /* Offset= -36 (628) */
/* 686 */ 0x1d, /* FC_SMFARRAY */
/* 688 */ 0x0, /* 0 */
/* 690 */ NdrFcShort( 0x8 ), /* 8 */
/* 692 */ 0x2, /* FC_CHAR */
/* 694 */ 0x5b, /* FC_END */
/* 696 */ 0x15, /* FC_STRUCT */
/* 698 */ 0x3, /* 3 */
/* 700 */ NdrFcShort( 0x10 ), /* 16 */
/* 702 */ 0x8, /* FC_LONG */
/* 704 */ 0x6, /* FC_SHORT */
/* 706 */ 0x6, /* FC_SHORT */
/* 708 */ 0x4c, /* FC_EMBEDDED_COMPLEX */

```

```

/* 680 */ 0x0, /* 0 */
/* 682 */ NdrFcShort( 0xfffffff1 ), /* Offset= -15 (666) */
/* 684 */ 0x5b, /* FC_END */
/* 686 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 688 */ 0x3, /* 3 */
/* 690 */ NdrFcShort( 0x20 ), /* 32 */
/* 692 */ NdrFcShort( 0x0 ), /* 0 */
/* 694 */ NdrFcShort( 0xa ), /* Offset= 10 (700) */
/* 696 */ 0x8, /* FC_LONG */
/* 698 */ 0x39, /* FC_ALIGNM8 */
/* 700 */ 0x36, /* FC_POINTER */
/* 702 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 704 */ 0x0, /* 0 */
/* 706 */ NdrFcShort( 0xffffffe7 ), /* Offset= -25 (672) */
/* 708 */ 0x5b, /* FC_END */
/* 710 */ 0x11, 0x0, /* FC_UP */
/* 712 */ NdrFcShort( 0xffffff10 ), /* Offset= -240 (462) */
/* 714 */ 0x1b, /* FC_CARRAY */
/* 716 */ 0x0, /* 0 */
/* 718 */ NdrFcShort( 0x1 ), /* 1 */
/* 720 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 722 */ 0x0, /* */
/* 724 */ NdrFcShort( 0x0 ), /* 0 */
/* 726 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 728 */ 0x1, /* FC_BYTE */
/* 730 */ 0x5b, /* FC_END */
/* 732 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 734 */ 0x3, /* 3 */
/* 736 */ NdrFcShort( 0x10 ), /* 16 */
/* 738 */ NdrFcShort( 0x0 ), /* 0 */
/* 740 */ NdrFcShort( 0x6 ), /* Offset= 6 (728) */
/* 742 */ 0x8, /* FC_LONG */
/* 744 */ 0x39, /* FC_ALIGNM8 */
/* 746 */ 0x36, /* FC_POINTER */
/* 748 */ 0x5b, /* FC_END */
/* 750 */ 0x12, 0x0, /* FC_UP */
/* 752 */ NdrFcShort( 0xffffffe6 ), /* Offset= -26 (704) */
/* 754 */ 0x1b, /* FC_CARRAY */
/* 756 */ 0x1, /* 1 */
/* 758 */ NdrFcShort( 0x2 ), /* 2 */
/* 760 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
/* 762 */ 0x0, /* */
/* 764 */ NdrFcShort( 0x0 ), /* 0 */
/* 766 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 768 */ 0x6, /* FC_SHORT */
/* 770 */ 0x5b, /* FC_END */
/* 772 */ 0x1a, /* FC_BOGUS_STRUCT */
/* 774 */ 0x3, /* 3 */
/* 776 */ NdrFcShort( 0x10 ), /* 16 */
/* 778 */ NdrFcShort( 0x0 ), /* 0 */
/* 780 */ NdrFcShort( 0x6 ), /* Offset= 6 (756) */
/* 782 */ 0x8, /* FC_LONG */
/* 784 */ 0x39, /* FC_ALIGNM8 */
/* 786 */ 0x36, /* FC_POINTER */
/* 788 */ 0x5b, /* FC_END */
/* 790 */ 0x15, /* FC_STRUCT */
/* 792 */ 0x3, /* 3 */

```

```

                0x12, 0x0,          /* FC_UP */
/* 758 */ NdrFcShort( 0xffffffe6 ), /* Offset= -26 (732) */
/* 760 */
                0x1b,             /* FC_CARRAY */
                0x3,              /* 3 */
/* 762 */ NdrFcShort( 0x4 ), /* 4 */
/* 764 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                0x0,             /* */
/* 766 */ NdrFcShort( 0x0 ), /* 0 */
/* 768 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 770 */ 0x8, /* FC_LONG */
                0x5b,           /* FC_END */
/* 772 */
                0x1a,           /* FC_BOGUS_STRUCT */
                0x3,           /* 3 */
/* 774 */ NdrFcShort( 0x10 ), /* 16 */
/* 776 */ NdrFcShort( 0x0 ), /* 0 */
/* 778 */ NdrFcShort( 0x6 ), /* Offset= 6 (784) */
/* 780 */ 0x8, /* FC_LONG */
                0x39,          /* FC_ALIGNM8 */
/* 782 */ 0x36, /* FC_POINTER */
                0x5b,          /* FC_END */
/* 784 */
                0x12, 0x0,          /* FC_UP */
/* 786 */ NdrFcShort( 0xffffffe6 ), /* Offset= -26 (760) */
/* 788 */
                0x1b,             /* FC_CARRAY */
                0x7,             /* 7 */
/* 790 */ NdrFcShort( 0x8 ), /* 8 */
/* 792 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
                0x0,             /* */
/* 794 */ NdrFcShort( 0x0 ), /* 0 */
/* 796 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 798 */ 0xb, /* FC_HYPER */
                0x5b,           /* FC_END */
/* 800 */
                0x1a,           /* FC_BOGUS_STRUCT */
                0x3,           /* 3 */
/* 802 */ NdrFcShort( 0x10 ), /* 16 */
/* 804 */ NdrFcShort( 0x0 ), /* 0 */
/* 806 */ NdrFcShort( 0x6 ), /* Offset= 6 (812) */
/* 808 */ 0x8, /* FC_LONG */
                0x39,          /* FC_ALIGNM8 */
/* 810 */ 0x36, /* FC_POINTER */
                0x5b,          /* FC_END */
/* 812 */
                0x12, 0x0,          /* FC_UP */
/* 814 */ NdrFcShort( 0xffffffe6 ), /* Offset= -26 (788) */
/* 816 */
                0x15,             /* FC_STRUCT */
                0x3,             /* 3 */
/* 818 */ NdrFcShort( 0x8 ), /* 8 */
/* 820 */ 0x8, /* FC_LONG */
                0x8,             /* FC_LONG */
/* 822 */ 0x5c, /* FC_PAD */
                0x5b,           /* FC_END */
/* 824 */
                0x1b,             /* FC_CARRAY */
                0x3,             /* 3 */
/* 826 */ NdrFcShort( 0x8 ), /* 8 */
/* 828 */ 0x7, /* Corr desc: FC_USHORT */
                0x0,             /* */
/* 830 */ NdrFcShort( 0xffc8 ), /* -56 */

```

```

/* 832 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 834 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
                0x0,           /* 0 */
/* 836 */ NdrFcShort( 0xfffffec ), /* Offset= -20 (816) */
/* 838 */ 0x5c, /* FC_PAD */
                0x5b,           /* FC_END */
/* 840 */
                0x1a,           /* FC_BOGUS_STRUCT */
                0x3,           /* 3 */
/* 842 */ NdrFcShort( 0x38 ), /* 56 */
/* 844 */ NdrFcShort( 0xfffffec ), /* Offset= -20 (824) */
/* 846 */ NdrFcShort( 0x0 ), /* Offset= 0 (846) */
/* 848 */ 0x6, /* FC_SHORT */
                0x6,           /* FC_SHORT */
/* 850 */ 0x38, /* FC_ALIGNM4 */
                0x8,           /* FC_LONG */
/* 852 */ 0x8, /* FC_LONG */
                0x4c,          /* FC_EMBEDDED_COMPLEX */
/* 854 */ 0x4, /* 4 */
                NdrFcShort( 0xfffffe0d ), /* Offset= -499 (356) */
                0x5b,          /* FC_END */
/* 858 */
                0x12, 0x0,          /* FC_UP */
/* 860 */ NdrFcShort( 0xfffff02 ), /* Offset= -254 (606) */
/* 862 */
                0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 864 */ 0x1, /* FC_BYTE */
                0x5c,          /* FC_PAD */
/* 866 */
                0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 868 */ 0x6, /* FC_SHORT */
                0x5c,          /* FC_PAD */
/* 870 */
                0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 872 */ 0x8, /* FC_LONG */
                0x5c,          /* FC_PAD */
/* 874 */
                0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 876 */ 0xa, /* FC_FLOAT */
                0x5c,          /* FC_PAD */
/* 878 */
                0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 880 */ 0xc, /* FC_DOUBLE */
                0x5c,          /* FC_PAD */
/* 882 */
                0x12, 0x0,          /* FC_UP */
/* 884 */ NdrFcShort( 0xfffffda4 ), /* Offset= -604 (280) */
/* 886 */
                0x12, 0x10,          /* FC_UP [pointer_deref] */
/* 888 */ NdrFcShort( 0xfffffda6 ), /* Offset= -602 (286) */
/* 890 */
                0x12, 0x10,          /* FC_UP [pointer_deref] */
/* 892 */ NdrFcShort( 0xfffffdbc ), /* Offset= -580 (312) */
/* 894 */
                0x12, 0x10,          /* FC_UP [pointer_deref] */
/* 896 */ NdrFcShort( 0xfffffdca ), /* Offset= -566 (330) */
/* 898 */
                0x12, 0x10,          /* FC_UP [pointer_deref] */
/* 900 */ NdrFcShort( 0xfffffdd8 ), /* Offset= -552 (348) */
/* 902 */
                0x12, 0x10,          /* FC_UP [pointer_deref] */
/* 904 */ NdrFcShort( 0x2 ), /* Offset= 2 (906) */
/* 906 */

```

```

        0x12, 0x0,          /* FC_UP */
/* 908 */ NdrFcShort( 0x16 ), /* Offset= 22 (930) */
/* 910 */
        0x15,             /* FC_STRUCT */
        0x7,              /* 7 */
/* 912 */ NdrFcShort( 0x10 ), /* 16 */
/* 914 */ 0x6,           /* FC_SHORT */
        0x1,             /* FC_BYTE */
/* 916 */ 0x1,         /* FC_BYTE */
        0x38,           /* FC_ALIGNM4 */
/* 918 */ 0x8,         /* FC_LONG */
        0x39,           /* FC_ALIGNM8 */
/* 920 */ 0xb,         /* FC_HYPER */
        0x5b,           /* FC_END */
/* 922 */
        0x12, 0x0,          /* FC_UP */
/* 924 */ NdrFcShort( 0xffffffff2 ), /* Offset= -14 (910) */
/* 926 */
        0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 928 */ 0x2,         /* FC_CHAR */
        0x5c,           /* FC_PAD */
/* 930 */
        0x1a,             /* FC_BOGUS_STRUCT */
        0x7,              /* 7 */
/* 932 */ NdrFcShort( 0x20 ), /* 32 */
/* 934 */ NdrFcShort( 0x0 ), /* 0 */
/* 936 */ NdrFcShort( 0x0 ), /* Offset= 0 (936) */
/* 938 */ 0x8,           /* FC_LONG */
        0x8,             /* FC_LONG */
/* 940 */ 0x6,         /* FC_SHORT */
        0x6,             /* FC_SHORT */
/* 942 */ 0x6,         /* FC_SHORT */
        0x6,             /* FC_SHORT */
/* 944 */ 0x4c,        /* FC_EMBEDDED_COMPLEX */
        0x0,             /* 0 */
/* 946 */ NdrFcShort( 0xffffffff54 ), /* Offset= -940 (6) */
/* 948 */ 0x5c,        /* FC_PAD */
        0x5b,           /* FC_END */
/* 950 */ 0xb4,        /* FC_USER_MARSHAL */
        0x83,           /* 131 */
/* 952 */ NdrFcShort( 0x0 ), /* 0 */
/* 954 */ NdrFcShort( 0x18 ), /* 24 */
/* 956 */ NdrFcShort( 0x0 ), /* 0 */
/* 958 */ NdrFcShort( 0xffffffff44 ), /* Offset= -956 (2) */
/* 960 */
        0x11, 0x4,          /* FC_RP [allocated_on_stack] */
/* 962 */ NdrFcShort( 0x6 ), /* Offset= 6 (968) */
/* 964 */
        0x13, 0x0,          /* FC_OP */
/* 966 */ NdrFcShort( 0xffffffffdc ), /* Offset= -36 (930) */
/* 968 */ 0xb4,        /* FC_USER_MARSHAL */
        0x83,           /* 131 */
/* 970 */ NdrFcShort( 0x0 ), /* 0 */
/* 972 */ NdrFcShort( 0x18 ), /* 24 */
/* 974 */ NdrFcShort( 0x0 ), /* 0 */
/* 976 */ NdrFcShort( 0xfffffffff4 ), /* Offset= -12 (964) */
        0x0
    }
};

const CInterfaceProxyVtbl * _tpcc_com_ps_ProxyVtblList[] =
{

```

```

    ( CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
    0
};

const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[] =
{
    ( CInterfaceStubVtbl *) &_ITPCCStubVtbl,
    0
};

PCInterfaceName const _tpcc_com_ps_InterfaceNamesList[] =
{
    "ITPCC",
    0
};

#define _tpcc_com_ps_CHECK_IID(n)      IID_GENERIC_CHECK_IID( _tpcc_com_ps, pIID,
n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{
    if(!_tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }

    return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) &_tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) &_tpcc_com_ps_StubVtblList,
    (const PCInterfaceName *) &_tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    &_tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, /* table of [async_uid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0 /* Filler3 */
};

#endif /* defined(_M_IA64) || defined(_M_AXP64) */



---



## tpcc_com_sl.rgs



```

HKCR
{
 TPCC.StockLevel.1 = s 'StockLevel Class'
 {
 CLSID = s '{2668369E-A50D-11D2-BA4E-00C04FBFE08B}'
 }
 TPCC.StockLevel = s 'StockLevel Class'
 {
 CurVer = s 'TPCC.StockLevel.1'
 }
}

```


```

```

    }
    NoRemove CLSID
    {
        ForceRemove {2668369E-A50D-11D2-BA4E-00C04FBFE08B} = s
'StockLevel Class'
    {
        ProgID = s 'TPCC.StockLevel.1'
        VersionIndependentProgID = s 'TPCC.StockLevel'
        InprocServer32 = s '%MODULE%'
        {
            val ThreadingModel = s 'Both'
        }
    }
}
}
}

```

tpcc_dblib.cpp

```

/*
 * FILE:          TPCC_DBLIB.CPP
 *               Microsoft TPC-C Kit Ver. 4.20.000
 *               Copyright Microsoft, 1999
 *
 * All Rights Reserved
 *
 * Version 4.10.000 audited by Richard Gimarc,
Performance Metrics, 3/17/99
 *
 * PURPOSE:  Implements dblib calls for TPC-C txns.
 * Contact:  Charles Levine (clevine@microsoft.com)
 *
 * Change history:
 * 4.20.000 - updated rev number to match kit
 * 4.10.001 - not deleting error class in catch handler on deadlock
retry;
 *
 * not a functional bug, but a memory leak
 * - had to tweak some declarations to compile
with latest SDK; no functional change
 */

#include <windows.h>
#include <stdio.h>
#include <assert.h>

#define DBNTWIN32
#include <sqlfront.h>
#include <sqlldb.h>

#ifdef ICECAP
#include <icapexp.h>
#endif

// need to declare functions for export
#define DllDecl __declspec( dllexport )

#include "..\..\common\src\error.h"
#include "..\..\common\src\trans.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_dblib.h"

#define DEFCLPACKSIZE 4096

// version string; must match return value from tpcc_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 WINAPIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:
            DisableThreadLibraryCalls(hModule);
            dbinit(); // initialize dblib
            break;

        case DLL_PROCESS_DETACH:
            dbexit(); // close all dblib
            structures/connections break;

        default:
            /* nothing */;
    }
    return TRUE;
}

int err_handler(DBPROCESS *dbproc, int severity, int dberr, int oserr, LPCSTR
dberrstr, LPCSTR oserrstr)
{
    CTPCC_DBLIB *pConn;
    assert(dbproc != NULL);
    pConn = (CTPCC_DBLIB*) dbgetuserdata(dbproc);

    if (pConn != NULL)
    {
        pConn->SetDbLibError( severity, dberr, oserr, dberrstr, oserrstr
);
    }
    return INT_CANCEL;
}

/* FUNCTION: int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int
severity, char *msgtext)
 *
 * PURPOSE: This function handles DB-Library SQL Server error messages
 *
 * ARGUMENTS: DBPROCESS *dbproc DBPROCESS id
pointer
 * DBINT msgno
 * message number
 * int
 * msgstate message state
 * int
 * severity message severity
 * char *msgtext
 * printable message description
 *
 * RETURNS: int INT_CONTINUE
continue if error is SQLETIME else INT_CANCEL action

```

```

*
*          INT_CANCEL          cancel operation
*
* COMMENTS:      This function also sets the dead lock dbproc variable if
necessary.
*
*/

// typedef INT (SQLAPI *DBMSGHANDLE_PROC) (PDBPROCESS, DBINT, INT, INT, LPCSTR,
LPCSTR, LPCSTR, DBUSMALLINT);

int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int severity,
LPCSTR msgtext, LPCSTR srvname, LPCSTR
procname, DBUSMALLINT line)
{
    CTPCC_DBLIB          *pConn;

    assert(dbproc != NULL);
    pConn = (CTPCC_DBLIB*)dbgetuserdata(dbproc);

    if (pConn != NULL)
    {
        pConn->SetSqlError( msgno, msgstate, severity, msgtext );
    }

    return 0;
}

/* FUNCTION: void UtilStrCpy(char * pDest, char * pSrc, int n)
*
* PURPOSE:      This function copies n characters from string pSrc to pDst and
places a
*
*               null character at the end of the destination string.
*
* ARGUMENTS:   char          *pDest  destination string
pointer
*
*               char          *pSrc
*
*               int           n
*
*               number of characters to copy
*
* RETURNS:     None
*
* COMMENTS:    Unlike strncpy this function ensures that the result string is
always null terminated.
*
*/

inline static void UtilStrCpy(char * pDest, const BYTE * pSrc, int n)
{
    strncpy(pDest, (char *)pSrc, n);
    pDest[n] = '\0';

    return;
}

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
*
*/

char* CTPCC_DBLIB_ERR::ErrorText(void)
{
    int i;

```

```

        static SERRORMSG errorMsgs[] =
        {
            { ERR_WRONG_SP_VERSION,          "Wrong version of stored
procs on database server" },
            { ERR_INVALID_CUST,              "Invalid Customer
id,name." },
            { ERR_NO_SUCH_ORDER,             "No orders found for
customer." },
            { ERR_RETRIED_TRANS,             "Retries before
transaction succeeded." },
            { 0,                              },
        }
    };

    static char szNotFound[] = "Unknown error number.";

    for(i=0; errorMsgs[i].szMsg[0]; i++)
    {
        if ( m_errno == errorMsgs[i].iError )
            break;
    }
    if ( !errorMsgs[i].szMsg[0] )
        return szNotFound;
    else
        return errorMsgs[i].szMsg;
}

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_DBLIB* CTPCC_DBLIB_new(
LPCSTR szServer,          // name of SQL server
LPCSTR szUser,            // user name for login
LPCSTR szPassword,       // password for login
LPCSTR szHost,           // workstation name; shows up in
sp_who; max 30 chars, only first 10 kept by SQL Server
LPCSTR szDatabase )      // name of database to use
{
    return new CTPCC_DBLIB( szServer, szUser, szPassword, szHost, szDatabase
);
}

CTPCC_DBLIB::CTPCC_DBLIB (
LPCSTR szServer,          // name of SQL server
LPCSTR szUser,            // user name for login
LPCSTR szPassword,       // password for login
LPCSTR szHost,           // workstation name; shows up in
sp_who; max 30 chars, only first 10 kept by SQL Server
LPCSTR szDatabase )      // name of database to use
{
    LOGINREC *login;
    const BYTE *pData;

    // initialization
    m_dbproc = NULL;
    m_DbLibErr = (CDBLIBERR*)NULL;
    m_SqlErr = (CSQLERR*)NULL;

    m_MaxRetries = 10;          // how many retries on deadlock

    // increase max number of connections if getting close
    if ( dbgetmaxprocs() < (iConnectionCount+5) )

```



```

{
    if ( dbsetmaxprocs(iConnectionCount+10) == FAIL )
        ThrowError(CDBLIBERR::eDbSetMaxProcs);
}

// allocate a login structure
login = dblogin();
if (login == NULL)
    ThrowError(CDBLIBERR::eLogin);
InterlockedIncrement( &iConnectionCount );

// register error and message handler functions
if (dbprocerrhandle(login, err_handler) == NULL)
    ThrowError(CDBLIBERR::eDbProcHandler);

if (dbprocmsghandle(login, msg_handler) == NULL)
    ThrowError(CDBLIBERR::eDbProcHandler);

DBSETLUSER(login, szUser);
DBSETLPWD(login, szPassword);
DBSETLHOST(login, szHost);
DBSETLPACKET(login, (unsigned short)DEFCLPACKSIZE);
DBSETLVERSION(login, DBVER60); // use dblib ver 6.0
client behavior

// set time to wait for login
if (dbsetlogintime(60) == FAIL)
    ThrowError(CDBLIBERR::eDbSet);

// set time to wait for statement execution
if (dbsettime(180) == FAIL)
    ThrowError(CDBLIBERR::eDbSet);

m_dbproc = dbopen(login, szServer);

// deallocate login structure before checking for success
dbfreelogin( login );

if (m_dbproc == NULL)
    ThrowError(CDBLIBERR::eDbOpen);

// save address of class instance so that the message and error handler
// can get to data.
dbsetuserdata(m_dbproc, (LPVOID)this);

// Use the the right database
if (dbuse(m_dbproc, szDatabase) == FAIL)
    ThrowError(CDBLIBERR::eDbUse);

// set connection properties to match those used by ODBC
dbcmd(m_dbproc, "set ANSI_DEFAULTS ON ");
dbcmd(m_dbproc, "set CURSOR_CLOSE_ON_COMMIT OFF ");
dbcmd(m_dbproc, "set IMPLICIT_TRANSACTIONS OFF ");
dbcmd(m_dbproc, "set NOCOUNT ON "); // do not
return row counts
dbcmd(m_dbproc, "set XACT_ABORT ON "); // rollback transaction
on abort

// for coyote
dbcmd(m_dbproc, "set ansi_warnings on "); //
dbcmd(m_dbproc, "set ansi_nulls on "); //

```

```

if (dbsqlexec(m_dbproc) == FAIL)
    ThrowError(CDBLIBERR::eDbSqlExec);

// This value must match the number of commands above.
DiscardNextResults(2);
DiscardNextResults(5); // coyote

// 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) != SUCCEEDED)
    ThrowError(CDBLIBERR::eDbResults);

if (dbnextrow(m_dbproc) != REG_ROW)
    ThrowError(CDBLIBERR::eDbNextRow);

char szSrvVersion[16];
pData=dbdata(m_dbproc, 1);
if (pData)
    UtilStrCpy(szSrvVersion, pData, dbdatlen(m_dbproc, 1));
else
    szSrvVersion[0]=0;
if (strcmp(szSrvVersion,sVersion))
    throw new CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_WRONG_SP_VERSION
);

DiscardNextRows(0);
DiscardNextResults(0);
}

CTPCC_DBLIB::~CTPCC_DBLIB( void )
{
    // close db connection and deallocate resources
    dbclose(m_dbproc);
    InterlockedDecrement( &iConnectionCount );
    if (m_DbLibErr != NULL)
        delete m_DbLibErr;
    if (m_SqlErr != NULL)
        delete m_SqlErr;
}

void CTPCC_DBLIB::SetDbLibError(int severity, int dberr, int oserr, LPCSTR dberrstr,
LPCSTR oserrstr)
{
    delete m_DbLibErr;
    m_DbLibErr = new CDBLIBERR(CDBLIBERR::eUnknown, severity, dberr, oserr);

    if (dberrstr != NULL)
    {
        m_DbLibErr->m_dberrstr = new char[ strlen(dberrstr)+1 ];
        strcpy( m_DbLibErr->m_dberrstr, dberrstr );
    }

    if (oserrstr != NULL)
    {
        m_DbLibErr->m_oserrstr = new char[ strlen(oserrstr)+1 ];
        strcpy( m_DbLibErr->m_oserrstr, oserrstr );
    }
}

```

```

}

void CTPCC_DBLIB::SetSqlError( int /*DBINT*/ msgno, int msgstate, int severity,
LPCSTR msgtext )
{
    if (m_SqlErr == NULL)
        m_SqlErr = new CSQLErr();

    m_SqlErr->m_msgno = msgno;
    m_SqlErr->m_msgstate = msgstate;
    m_SqlErr->m_severity = severity;

    delete [] m_SqlErr->m_msgtext;
    if (msgtext != NULL)
    {
        m_SqlErr->m_msgtext = new char[ strlen(msgtext)+1 ];
        strcpy( m_SqlErr->m_msgtext, msgtext );
    }
}

void CTPCC_DBLIB::ThrowError( CDBLIBERR::ACTION eAction )
{
    // discard anything still in return buffer
    DiscardNextRows(-1);
    DiscardNextResults(-1);

    // check for SQL Server error first; if yes, throw it and ignore any
    DBLib error.
    if (m_SqlErr != NULL)
    {
        CSQLErr *pSqlErr;
        pSqlErr = m_SqlErr;
        m_SqlErr = NULL; // clear our pointer to instance; catch
        handler will delete
        throw pSqlErr;
    }

    CDBLIBERR *pDbLibErr;
    if (m_DbLibErr == NULL)
        // this case isn't expected to happen, since it means that an
        error was returned
        // but the error handlers were not called.
        pDbLibErr = new CDBLIBERR(eAction);
    else
    {
        pDbLibErr = m_DbLibErr;
        pDbLibErr->m_eAction = eAction;
        m_DbLibErr = NULL; // clear our pointer to instance;
        catch handler will delete
    }

    throw pDbLibErr;
}

// Read and discard rows until no more. Throw an exception if number of rows read
doesn't
// match number of rows expected. The row count will be ignored if the expected
count value
// passed in is negative. A typical use of this routine is to verify that there are
no more
// rows to be read.
void CTPCC_DBLIB::DiscardNextRows(int iExpectedCount)

```

```

{
    int          iRowsRead = 0;
    RETCODE     rc;

    while (TRUE)
    {
        rc = dbnextrow(m_dbproc);
        if (rc == NO_MORE_ROWS)
            break;
        if (rc == FAIL)
        {
            if (iExpectedCount >= 0)
                ThrowError(CDBLIBERR::eDbNextRow);
            else
                break;
        }
        iRowsRead++;
    }

    if ((iExpectedCount >= 0) &&
        (iExpectedCount != iRowsRead))
        ThrowError(CDBLIBERR::eWrongRowCount);
}

// Read and discard results until no more. Throw an exception if number of result
sets read doesn't
// match number expected. The result set count will be ignored if the expected
count value
// passed in is negative. A typical use of this routine is to verify that there are
no more
// result sets to be read.
void CTPCC_DBLIB::DiscardNextResults(int iExpectedCount)
{
    int          iResultsRead = 0;
    RETCODE     rc;

    while (TRUE)
    {
        rc = dbresults(m_dbproc);
        if (rc == NO_MORE_RESULTS)
            break;
        if (rc == FAIL)
        {
            if (iExpectedCount >= 0)
                ThrowError(CDBLIBERR::eDbResults);
            else
                break;
        }

        DiscardNextRows(-1);
        iResultsRead++;
    }

    if ((iExpectedCount >= 0) &&
        (iExpectedCount != iResultsRead))
        ThrowError(CDBLIBERR::eWrongRowCount);
}

void CTPCC_DBLIB::StockLevel()
{
    int          iTryCount = 0;
    const BYTE  *pData;

```

```

ResetError();
while (TRUE)
{
    try
    {
        dbrpcinit(m_dbproc, "tpcc_stocklevel", 0);
        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE
*) &m_txn.StockLevel.w_id); // @w_id smallint
        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE
*) &m_txn.StockLevel.d_id); // @d_id tinyint
        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE
*) &m_txn.StockLevel.threshold); // @threshold smallint

        if (dbrpcexec(m_dbproc) == FAIL)
            ThrowError(CDBLIBERR::eDbRpcExec);

        if (dbresults(m_dbproc) != SUCCEED)
            ThrowError(CDBLIBERR::eDbResults);

        if (dbnextrow(m_dbproc) != REG_ROW)
            ThrowError(CDBLIBERR::eDbNextRow);

        if (pData=dbdata(m_dbproc, 1))
            m_txn.StockLevel.low_stock = *((long *)
pData);

        DiscardNextRows(0);
        DiscardNextResults(0);

        m_txn.StockLevel.exec_status_code = eOK;
        return;
    }
    catch (CSQLERR *e)
    {
        if ((e->m_msgno == 1205 ||
(e->m_msgno == iErrOleDbProvider &&
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;

```

```

DBDATERECDaterec;

int iTryCount = 0;
const BYTE *pData;

ResetError();
while (TRUE)
{
    try
    {
        dbrpcinit(m_dbproc, "tpcc_neworder", 0);
        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE
*) &m_txn.NewOrder.w_id);
        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE
*) &m_txn.NewOrder.d_id);
        dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE
*) &m_txn.NewOrder.c_id);
        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE
*) &m_txn.NewOrder.o_ol_cnt);

        // check whether any order lines are for a remote
warehouse
        m_txn.NewOrder.o_all_local = 1;
        for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++)
        {
            if (m_txn.NewOrder.OL[i].ol_supply_w_id !=
m_txn.NewOrder.w_id)
            {
                m_txn.NewOrder.o_all_local = 0;
                // at least one remote warehouse
                break;
            }
        }
        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE
*) &m_txn.NewOrder.o_all_local);

        for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++)
        {
            dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -
1, (BYTE *) &m_txn.NewOrder.OL[i].ol_i_id);
            dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -
1, (BYTE *) &m_txn.NewOrder.OL[i].ol_supply_w_id);
            dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -
1, (BYTE *) &m_txn.NewOrder.OL[i].ol_quantity);
        }

        if (dbrpcexec(m_dbproc) == FAIL)
            ThrowError(CDBLIBERR::eDbRpcExec);

        // Get order line results
        m_txn.NewOrder.total_amount = 0;
        for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++)
        {
            if (dbresults(m_dbproc) != SUCCEED)
                ThrowError(CDBLIBERR::eDbResults);

            if (dbnumcols(m_dbproc) != 5)

                ThrowError(CDBLIBERR::eWrongNumCols);

```

```

        if (dbnextrow(m_dbproc) != REG_ROW)
            ThrowError(CDBLIBERR::eDbNextRow);

        if (pData=dbdata(m_dbproc, 1))

            UtilStrCpy(m_txn.NewOrder.OL[i].ol_i_name, pData, dbdatlen(m_dbproc, 1));
            if (pData=dbdata(m_dbproc, 2))
                m_txn.NewOrder.OL[i].ol_stock =
                (* (DBSMALLINT *) pData);

            if (pData=dbdata(m_dbproc, 3))

                UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_generic, pData,
                dbdatlen(m_dbproc, 3));

            if (pData=dbdata(m_dbproc, 4))
                dbconvert(m_dbproc, SQLNUMERIC,
                (LPCBYTE)pData, dbdatlen(m_dbproc,4),
                SQLFLT8, (BYTE
                *) &m_txn.NewOrder.OL[i].ol_i_price, 8);

            if (pData=dbdata(m_dbproc, 5))

                dbconvert(m_dbproc, SQLNUMERIC,
                (LPCBYTE)pData, dbdatlen(m_dbproc,5),
                SQLFLT8, (BYTE
                *) &m_txn.NewOrder.OL[i].ol_amount, 8);

            m_txn.NewOrder.total_amount =
            m_txn.NewOrder.total_amount + m_txn.NewOrder.OL[i].ol_amount;

            DiscardNextRows(0);
        }

        // get remaining values for w_tax, d_tax, o_id,
        c_last, c_discount, c_credit, o_entry_d, commit_flag
        if (dbresults(m_dbproc) != SUCCEED)
            ThrowError(CDBLIBERR::eDbResults);

        if (dbnextrow(m_dbproc) != REG_ROW)
            ThrowError(CDBLIBERR::eDbNextRow);

        if (dbnumcols(m_dbproc) != 8)
            ThrowError(CDBLIBERR::eWrongNumCols);

        if (pData=dbdata(m_dbproc, 1))

            dbconvert(m_dbproc, SQLNUMERIC,
            (LPCBYTE)pData, dbdatlen(m_dbproc,1), SQLFLT8, (BYTE *) &m_txn.NewOrder.w_tax, 8);
            if (pData=dbdata(m_dbproc, 2))

                dbconvert(m_dbproc, SQLNUMERIC,
                (LPCBYTE)pData, dbdatlen(m_dbproc,2), SQLFLT8, (BYTE *) &m_txn.NewOrder.d_tax, 8);
            if (pData=dbdata(m_dbproc, 3))
                m_txn.NewOrder.o_id = (*(DBINT *) pData);
            if (pData=dbdata(m_dbproc, 4))
                UtilStrCpy(m_txn.NewOrder.c_last, pData,
                dbdatlen(m_dbproc, 4));

            if (pData=dbdata(m_dbproc, 5))
                dbconvert(m_dbproc, SQLNUMERIC,
                (LPCBYTE)pData, dbdatlen(m_dbproc,5), SQLFLT8, (BYTE *) &m_txn.NewOrder.c_discount,
                8);

            if (pData=dbdata(m_dbproc, 6))

```

```

            UtilStrCpy(m_txn.NewOrder.c_credit, pData,
            dbdatlen(m_dbproc, 6));

            if (pData=dbdata(m_dbproc, 7))
            {
                datetime = *((DBDATETIME *) pData);
                dbdatecrack(m_dbproc, &daterec, &datetime);
                m_txn.NewOrder.o_entry_d.year =
                m_txn.NewOrder.o_entry_d.month =
                m_txn.NewOrder.o_entry_d.day =
                m_txn.NewOrder.o_entry_d.hour =
                m_txn.NewOrder.o_entry_d.minute =
                m_txn.NewOrder.o_entry_d.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
                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      datettime;
    DBDATEREC      daterec;

    int
    const BYTE      *pData;

    ResetError();

    while (TRUE)
    {
        try
        {
            dbrpcinit(m_dbproc, "tpcc_payment", 0);

            dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE
*) &m_txn.Payment.w_id);
            dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE
*) &m_txn.Payment.c_w_id);
            dbrpcparam(m_dbproc, NULL, 0, SQLFLT8, -1, -1, (BYTE
*) &m_txn.Payment.h_amount);
            dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE
*) &m_txn.Payment.d_id);
            dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE
*) &m_txn.Payment.c_d_id);
            dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE
*) &m_txn.Payment.c_id);

            // if customer id is zero, then payment is by name
            if (m_txn.Payment.c_id == 0)
                dbrpcparam(m_dbproc, NULL, 0, SQLCHAR, -1,
strlen(m_txn.Payment.c_last), (unsigned char *)m_txn.Payment.c_last);

            if (dbrpcexec(m_dbproc) == FAIL)
                ThrowError(CDBLIBERR::eDbRpcExec);

            if (dbresults(m_dbproc) != SUCCEED)
                ThrowError(CDBLIBERR::eDbResults);

            if (dbnextrow(m_dbproc) != REG_ROW)
                ThrowError(CDBLIBERR::eDbNextRow);

            if (dbnumcols(m_dbproc) != 27)
                ThrowError(CDBLIBERR::eWrongNumCols);

            if (pData=dbdata(m_dbproc, 1))
                m_txn.Payment.c_id = *((DBINT *) pData);
            if (pData=dbdata(m_dbproc, 2))
                UtilStrCpy(m_txn.Payment.c_last, pData,
dbdatlen(m_dbproc, 2));
            if (pData=dbdata(m_dbproc, 3))
            {
                datettime = *((DBDATETIME *) pData);
                dbdatecrack(m_dbproc, &daterec, &datettime);
                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 =
                m_txn.Payment.h_date.second =

            daterec.minute;
            daterec.second;
        }
    }
}

```

```

dbdatlen(m_dbproc, 4));
    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))
    {
        datettime = *((DBDATETIME *) pData);
        dbdatecrack(m_dbproc, &daterec, &datettime);
        m_txn.Payment.c_since.year = daterec.year;
        m_txn.Payment.c_since.month =
        m_txn.Payment.c_since.day = daterec.day;
        m_txn.Payment.c_since.hour = daterec.hour;
        daterec.month;
    }
}

```

```

                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;
DBDATERECC daterec;

int rc;
iTryCount = 0;
const BYTE *pData;

ResetError();

while (TRUE)
{
    try
    {
        dbrpcinit(m_dbproc, "tpcc_orderstatus", 0);

        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE
*) &m_txn.OrderStatus.w_id);
        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE
*) &m_txn.OrderStatus.d_id);
        dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE
*) &m_txn.OrderStatus.c_id);

        // if customer id is zero, then order status is by
name
        if (m_txn.OrderStatus.c_id == 0)
            dbrpcparam(m_dbproc, NULL, 0, SQLCHAR, -1,
strlen(m_txn.OrderStatus.c_last), (unsigned char *)m_txn.OrderStatus.c_last);

        if (dbrpcexec(m_dbproc) == FAIL)
            ThrowError(CDBLIBERR::eDbRpcExec);

        // Get order lines
        if (dbresults(m_dbproc) != SUCCEED)
        {
            if ((m_DbLibErr == NULL) && (m_SqlErr ==
NULL))
                throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
            else
                ThrowError(CDBLIBERR::eDbResults);
        }

        if (dbnumcols(m_dbproc) != 5)
            ThrowError(CDBLIBERR::eWrongNumCols);

        i = 0;
        while (TRUE)
        {
            rc = dbnextrow(m_dbproc);
            if (rc == NO_MORE_ROWS)
                break;
            if (rc != REG_ROW)
                ThrowError(CDBLIBERR::eDbNextRow);

            if(pData=dbdata(m_dbproc, 1))

                m_txn.OrderStatus.OL[i].ol_supply_w_id = (* (DBSMALLINT *) pData);
            if(pData=dbdata(m_dbproc, 2))
                m_txn.OrderStatus.OL[i].ol_i_id =
                (* (DBINT *) pData);

            if(pData=dbdata(m_dbproc, 3))

```

```

        m_txn.OrderStatus.OL[i].ol_quantity = (*(DBSMALLINT *) pData);
        if(pData=dbdata(m_dbproc, 4))
            dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,4),
                                SQLFLT8,
(BYTE *)&m_txn.OrderStatus.OL[i].ol_amount, 8);
        if(pData=dbdata(m_dbproc, 5))
        {
            datetime = *((DBDATETIME *)
pData);
            dbdatecrack(m_dbproc, &daterec,
&datetime);

        m_txn.OrderStatus.OL[i].ol_delivery_d.year = daterec.year;
        m_txn.OrderStatus.OL[i].ol_delivery_d.month = daterec.month;
        m_txn.OrderStatus.OL[i].ol_delivery_d.day = daterec.day;
        m_txn.OrderStatus.OL[i].ol_delivery_d.hour = daterec.hour;
        m_txn.OrderStatus.OL[i].ol_delivery_d.minute = daterec.minute;
        m_txn.OrderStatus.OL[i].ol_delivery_d.second = daterec.second;
        }
        i++;
    }
    m_txn.OrderStatus.o_ol_cnt = i;

    if (dbresults(m_dbproc) != SUCCEED)
        ThrowError(CDBLIBERR::eDbResults);

    if (dbnextrow(m_dbproc) != REG_ROW)
        ThrowError(CDBLIBERR::eDbNextRow);

    if (dbnumcols(m_dbproc) != 8)
        ThrowError(CDBLIBERR::eWrongNumCols);

    if(pData=dbdata(m_dbproc, 1))
        m_txn.OrderStatus.c_id = (*(DBINT *) pData);
    if(pData=dbdata(m_dbproc, 2))
        UtilStrCpy(m_txn.OrderStatus.c_last, pData,
dbdatlen(m_dbproc,2));
    if(pData=dbdata(m_dbproc, 3))
        UtilStrCpy(m_txn.OrderStatus.c_first, pData,
dbdatlen(m_dbproc,3));
    if(pData=dbdata(m_dbproc, 4))
        UtilStrCpy(m_txn.OrderStatus.c_middle,
pData, dbdatlen(m_dbproc, 4));
    if(pData=dbdata(m_dbproc, 5))
    {
        datetime = *((DBDATETIME *) pData);
        dbdatecrack(m_dbproc, &daterec, &datetime);
        m_txn.OrderStatus.o_entry_d.year =
            daterec.year;
        m_txn.OrderStatus.o_entry_d.month =
            daterec.month;
        m_txn.OrderStatus.o_entry_d.day =
            daterec.day;
        m_txn.OrderStatus.o_entry_d.hour =
            daterec.hour;
    }

```

```

        m_txn.OrderStatus.o_entry_d.minute =
daterec.minute;
        m_txn.OrderStatus.o_entry_d.second =
daterec.second;
    }
    if(pData=dbdata(m_dbproc, 6))
        m_txn.OrderStatus.o_carrier_id =
        (*(DBSMALLINT *) pData);
    if(pData=dbdata(m_dbproc, 7))
        dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,7),
                                SQLFLT8, (BYTE
*)&m_txn.OrderStatus.c_balance, 8);
    if(pData=dbdata(m_dbproc, 8))
        m_txn.OrderStatus.o_id = (*(DBINT *) pData);

    DiscardNextRows(0);
    DiscardNextResults(0);

    if (m_txn.OrderStatus.o_ol_cnt == 0)
        throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
    else if (m_txn.OrderStatus.c_id == 0 &&
m_txn.OrderStatus.c_last[0] == 0)
        throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
    else
        m_txn.OrderStatus.exec_status_code = eOK;

    return;
}
catch (CSQLERR *e)
{
    if ((e->m_msgno == 1205 ||
(e->m_msgno == iErrOleDbProvider &&
strstr(e->m_msgtext, sErrTimeoutExpired) !=
NULL)) &&
        (++iTryCount <= iMaxRetries))
    {
        // hit deadlock; backoff for increasingly
        // longer period
        delete e;
        Sleep(10 * iTryCount);
    }
    else
        throw;
}
// while (TRUE)
// if (iTryCount)
// throw new CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_DBLIB::Delivery()
{
    int i;
    int iTryCount = 0;
    const BYTE *pData;

    ResetError();
}

```

```

while (TRUE)
{
    try
    {
        dbrpcinit(m_dbproc, "tpcc_delivery", 0);

        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE
*) &m_txn.Delivery.w_id);
        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE
*) &m_txn.Delivery.o_carrier_id);

        if (dbrpcexec(m_dbproc) == FAIL)
            ThrowError(CDBLIBERR::eDbRpcExec);

        if (dbresults(m_dbproc) != SUCCEED)
            ThrowError(CDBLIBERR::eDbResults);

        if (dbnextrow(m_dbproc) != REG_ROW)
            ThrowError(CDBLIBERR::eDbNextRow);

        if (dbnumcols(m_dbproc) != 10)
            ThrowError(CDBLIBERR::eWrongNumCols);

        for (i=0; i<10; i++)
        {
            if (pData = dbdata(m_dbproc, i+1))
                m_txn.Delivery.o_id[i] = *((DBINT
*)pData);
        }

        DiscardNextRows(0);
        DiscardNextResults(0);

        m_txn.Delivery.exec_status_code = eOK;
        return;
    }
    catch (CSQLERR *e)
    {
        if ((e->m_msgno == 1205 ||
(e->m_msgno == iErrOleDbProvider &&
strstr(e->m_msgtext, sErrTimeoutExpired) !=
NULL)) &&
        {
            (++iTryCount <= iMaxRetries)

            // hit deadlock; backoff for increasingly
            longer period

            delete e;
            Sleep(10 * iTryCount);
        }
        else
            throw;
    }
} // while (TRUE)

// if (iTryCount)
// throw new CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_DBLIB::ResetError()
{
    if (m_DbLibErr != NULL)

```

```

{
    delete m_DbLibErr;
    m_DbLibErr = (CDBLIBERR*)NULL;
}

if (m_SqlErr != NULL)
{
    delete m_SqlErr;
    m_SqlErr = (CSQLERR*)NULL;
}

return;
}

```

tpcc_dblib.h

```

/* FILE: TPC_C_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
            eWrongNumCols, // more or less columns
            eDbResults, // error from
dbresults
            eDbRpcExec, // error from
dbrpcexec
            eDbSetMaxProcs, // error from
dbsetmaxprocs
            eDbProcHandler // error from either
dbprocerrhandle or dbprocmsghandle
        };
        CDBLIBERR(ACTION eAction, int severity = 0, int dberror = 0, int
oserr = 0)
        {
            m_eAction = eAction;
            m_severity = severity;
            m_dberror = dberror;
            m_oserr = oserr;

            m_dberrstr = NULL;
            m_oserrstr = NULL;
        };
        ~CDBLIBERR()
        {
            delete [] m_dberrstr;
            delete [] m_oserrstr;
        };
        ACTION m_eAction;
        int m_severity;
        int m_dberror;
        int m_oserr;
        char *m_dberrstr;
        char *m_oserrstr;
};

```

```

        int ErrorType() {return ERR_TYPE_DBLIB;};
        int ErrorNum() {return m_dberror;};
        char *ErrorText() {return m_dberrstr;};
};

class CTPCC_DBLIB_ERR : public CBaseErr
{
    public:
        enum CTPCC_DBLIB_ERRS
        {
            ERR_WRONG_SP_VERSION = 1, // "Wrong version of
stored procs on database server"
            ERR_INVALID_CUST, // "Invalid
Customer id,name."
            ERR_NO_SUCH_ORDER, // "No orders
found for customer."
            ERR_RETRIED_TRANS, // "Retries
before transaction succeeded."
        };
        CTPCC_DBLIB_ERR( int iErr ) { m_errno = iErr; m_iTryCount = 0;
};
        CTPCC_DBLIB_ERR( int iErr, int iTryCount ) { m_errno = iErr;
m_iTryCount = iTryCount; };
        int m_errno;
        int m_iTryCount;

        int ErrorType() {return ERR_TYPE_TPCC_DBLIB;};
        int ErrorNum() {return m_errno;};

        char *ErrorText();
};

class DllDecl CTPCC_DBLIB : public CTPCC_BASE
{
    private:
        // declare variables and private functions here...
        PDBPROCESS m_dbproc;
        CDBLIBERR *m_DbLibErr; // not allocated until
needed (maybe never)
        CSQLErr *m_SqlErr; //
not allocated until needed (maybe never)
        int m_MaxRetries; //
retry count on deadlock

        void DiscardNextRows(int iExpectedCount);
        void DiscardNextResults(int iExpectedCount);
        void ThrowError( CDBLIBERR::ACTION eAction );
        void ResetError();

        union
        {
            NEW_ORDER_DATA NewOrder;
            PAYMENT_DATA Payment;
            DELIVERY_DATA Delivery;
            STOCK_LEVEL_DATA StockLevel;
            ORDER_STATUS_DATA OrderStatus;
        }
        m_txn;
};

```

```

public:
    CTPCC_DBLIB(LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword,
LPCSTR szHost, LPCSTR szDatabase );
    ~CTPCC_DBLIB(void);

    inline PNEW_ORDER_DATA          BuffAddr_NewOrder()
    { return &m_txn.NewOrder; };
    inline PPAYMENT_DATA            BuffAddr_Payment()
    { return &m_txn.Payment; };
    inline PDELIVERY_DATA           BuffAddr_Delivery()
    { return &m_txn.Delivery; };
    inline PSTOCK_LEVEL_DATA        BuffAddr_StockLevel() {
return &m_txn.StockLevel; };
    inline PORDER_STATUS_DATA       BuffAddr_OrderStatus() {
return &m_txn.OrderStatus; };

    void NewOrder          ();
    void Payment           ();
    void Delivery          ();
    void StockLevel        ();
    void OrderStatus      ();

    // these are public because they must be called from the dblink
err_handler and msg_handler
    // 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);

```

tpcc_odbc.cpp

```

/* FILE: TPC_ODBC.CPP
 * Microsoft TPC-C Kit Ver. 4.20.000
 * Copyright Microsoft, 1999
 *
 * All Rights Reserved
 *
 * Version 4.10.000 audited by Richard Gimarc,
Performance Metrics, 3/17/99
 *
 * PURPOSE: Implements ODBC calls for TPC-C txns.
 * Contact: Charles Levine (clevine@microsoft.com)
 *
 * Change history:
 * 4.20.000 - updated rev number to match kit
 * 4.10.001 - not deleting error class in catch handler on deadlock
retry;
 * not a functional bug, but a memory leak
 */

#include <windows.h>
#include <stdio.h>
#include <assert.h>

```

```

#define DBNTWIN32
#include <sqltypes.h>
#include <sql.h>
#include <sqltext.h>
#include <odbcss.h>

#ifdef ICECAP
#include <icapexp.h>
#endif

// need to declare functions for export
#define DllDecl __declspec( dllexport )

#include "..\..\common\src\error.h"
#include "..\..\common\src\trans.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_odbc.h"

// version string; must match return value from tpcc_version stored proc
const char sVersion[] = "4.10.000";

const iMaxRetries = 10; // how many retries on deadlock

const int iErrOleDbProvider = 7312;
const char sErrTimeoutExpired[] = "Timeout expired";

static SQLHENV henv = SQL_NULL_HENV; // ODBC
environment handle

BOOL APIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:
            DisableThreadLibraryCalls(hModule);
            if ( SQLAllocHandleStd(SQL_HANDLE_ENV,
SQL_NULL_HANDLE, &henv) != SQL_SUCCESS )
                return FALSE;
            break;

        case DLL_PROCESS_DETACH:
            if (henv != NULL)
                SQLFreeEnv(henv);
            break;

        default:
            /* nothing */;
    }
    return TRUE;
}

/* FUNCTION: CTPCC_ODBC_ERR::ErrorText
 *
 */

char* CTPCC_ODBC_ERR::ErrorText(void)
{
    int i;

    static SERRORMSG errorMsgs[] =
    {

```

```

        { ERR_WRONG_SP_VERSION,          "Wrong version of stored
procs on database server" },
        { ERR_INVALID_CUST,             "Invalid Customer
id,name." },
        { ERR_NO_SUCH_ORDER,           "No orders found for
customer." },
        { ERR_RETRIED_TRANS,           "Retries before
transaction succeeded." },
        { 0,                            ""
    }
};

static char szNotFound[] = "Unknown error number.";

for(i=0; errorMsgs[i].szMsg[0]; i++)
{
    if ( m_errno == errorMsgs[i].iError )
        break;
}
if ( !errorMsgs[i].szMsg[0] )
    return szNotFound;
else
    return errorMsgs[i].szMsg;
}

// wrapper routine for class constructor
_declspec(dllexport) CTPCC_ODBC* CTPCC_ODBC_new(
LPCSTR szServer,          // name of SQL server
LPCSTR szUser,           // user name for login
LPCSTR szPassword,       // password for login
LPCSTR szHost,           // not used
LPCSTR szDatabase )     // name of database to use
{
    return new CTPCC_ODBC( szServer, szUser, szPassword, szHost, szDatabase );
}

CTPCC_ODBC::CTPCC_ODBC (
LPCSTR szServer,         // name of SQL server
LPCSTR szUser,          // user name
for login
LPCSTR szPassword,      // password for login
LPCSTR szHost,          // not used
LPCSTR szDatabase       // name of database to
use
)
{
    RETCODE          rc;

    // initialization
    m_hdbc = SQL_NULL_HDBC;
    m_hstmt = SQL_NULL_HSTMT;

    m_hstmtNewOrder = SQL_NULL_HSTMT;
    m_hstmtPayment = SQL_NULL_HSTMT;
    m_hstmtDelivery = SQL_NULL_HSTMT;
    m_hstmtOrderStatus = SQL_NULL_HSTMT;
    m_hstmtStockLevel = SQL_NULL_HSTMT;

    m_descNewOrderCols1 = SQL_NULL_HDESC;
    m_descNewOrderCols2 = SQL_NULL_HDESC;
    m_descOrderStatusCols1 = SQL_NULL_HDESC;

```

```

m_descOrderStatusCols2 = SQL_NULL_HDESC;

if ( SQLAllocHandle(SQL_HANDLE_DBC, henv, &m_hdbc) != SQL_SUCCESS )
    ThrowError(CODBCERR::eAllocHandle);

if ( SQLSetConnectOption(m_hdbc, SQL_PACKET_SIZE, 4096) != SQL_SUCCESS )
    ThrowError(CODBCERR::eConnOption);

{
    char          szConnectStr[256];
    char          szOutStr[1024];
    SQLSMALLINT  iOutStrLen;

    sprintf( szConnectStr, "DRIVER=SQL
Server;SERVER=%s;UID=%s;PWD=%s;DATABASE=%s",
            szServer, szUser, szPassword, szDatabase );

    rc = SQLDriverConnect(m_hdbc, NULL, (SQLCHAR*)szConnectStr,
sizeof(szConnectStr),
            (SQLCHAR*)szOutStr, sizeof(szOutStr), &iOutStrLen,
SQL_DRIVER_NOPROMPT );

    if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
        ThrowError(CODBCERR::eConnect);
}

if (SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmt) != SQL_SUCCESS)
    ThrowError(CODBCERR::eAllocHandle);

{
    char          buffer[128];

    // set some options affecting connection behavior
    strcpy(buffer, "set nocount on ");
    strcat(buffer, "set XACT_ABORT ON ");

    // for coyote
    strcat(buffer, "set ansi_warnings on ");
    strcat(buffer, "set ansi_nulls on ");

    rc = SQLExecDirect(m_hstmt, (unsigned char *)buffer, SQL_NTS);
    if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
        ThrowError(CODBCERR::eExecDirect);

    // verify that version of stored procs on server is correct
    char db_sp_version[10];
    strcpy(buffer, "{call tpcc_version}");
    rc = SQLExecDirect(m_hstmt, (unsigned char *)buffer, SQL_NTS);
    if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
        ThrowError(CODBCERR::eExecDirect);
    if ( SQLBindCol(m_hstmt, 1, SQL_C_CHAR, &db_sp_version,
sizeof(db_sp_version), NULL) != SQL_SUCCESS )
        ThrowError(CODBCERR::eBindCol);
    if ( SQLFetch(m_hstmt) == SQL_ERROR )
        ThrowError(CODBCERR::eFetch);
    if (strcmp(db_sp_version,sVersion))
        throw new CTPCC_ODBC_ERR(
CTPCC_ODBC_ERR::ERR_WRONG_SP_VERSION );

    SQLFreeHandle(SQL_HANDLE_STMT, m_hstmt);
}
}

```

```

        // Bind parameters for each of the transactions
        InitNewOrderParams();
        InitPaymentParams();
        InitOrderStatusParams();
        InitDeliveryParams();
        InitStockLevelParams();
    }

CTPCC_ODBC::~CTPCC_ODBC( void )
{
    // note: descriptors are automatically released when the connection is
    dropped
    SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtNewOrder);
    SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtPayment);
    SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtDelivery);
    SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtOrderStatus);
    SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtStockLevel);

    SQLDisconnect(m_hdbc);
    SQLFreeHandle(SQL_HANDLE_DBC, m_hdbc);
}

void CTPCC_ODBC::ThrowError( CODBCERR::ACTION eAction )
{
    RETCODE          rc;
    SDWORD           lNativeError;
    char             szState[6];
    char             szMsg[SQL_MAX_MESSAGE_LENGTH];
    char             szTmp[6*SQL_MAX_MESSAGE_LENGTH];
    CODBCERR         *pODBCerr; // not allocated until
    needed (maybe never)

    pODBCerr = new CODBCERR();

    pODBCerr->m_NativeError = 0;
    pODBCerr->m_eAction = eAction;
    pODBCerr->m_bDeadLock = FALSE;

    szTmp[0] = 0;
    while (TRUE)
    {
        rc = SQLError(henv, m_hdbc, m_hstmt, (BYTE *)&szState,
&lNativeError, (BYTE *)&szMsg, sizeof(szMsg),
NULL);

        if (rc == SQL_NO_DATA)
            break;

        // check for deadlock
        if (lNativeError == 1205 || (lNativeError == iErrOleDbProvider
&&
            strstr(szMsg, sErrTimeoutExpired) != NULL))
            pODBCerr->m_bDeadLock = TRUE;

        // capture the (first) database error
        if (pODBCerr->m_NativeError == 0 && lNativeError != 0)
            pODBCerr->m_NativeError = lNativeError;

        // quit if there isn't enough room to concatenate error text
        if ( ( strlen(szMsg) + 2 ) > ( sizeof(szTmp) - strlen(szTmp) ) )
            break;

        // include line break after first error msg

```

```

        if (szTmp[0] != 0)
            strcat( szTmp, "\n");
        strcat( szTmp, szMsg );
    }

    if (pODBCerr->m_odbcerrstr != NULL)
    {
        delete [] pODBCerr->m_odbcerrstr;
        pODBCerr->m_odbcerrstr = NULL;
    }

    if (strlen(szTmp) > 0)
    {
        pODBCerr->m_odbcerrstr = new char[ strlen(szTmp)+1 ];
        strcpy( pODBCerr->m_odbcerrstr, szTmp );
    }

    SQLFreeStmt(m_hstmt, SQL_CLOSE);
    throw pODBCerr;
}

void CTPCC_ODBC::InitStockLevelParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmtStockLevel) !=
SQL_SUCCESS )
        ThrowError(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtStockLevel;

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.StockLevel.w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.StockLevel.d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.StockLevel.threshold, 0, NULL) != SQL_SUCCESS
        )
        ThrowError(CODBCERR::eBindParam);

    if ( SQLBindCol(m_hstmt, 1, SQL_C_SLONG, &m_txn.StockLevel.low_stock, 0,
NULL) != SQL_SUCCESS )
        ThrowError(CODBCERR::eBindCol);
}

void CTPCC_ODBC::StockLevel()
{
    RETCODE          rc;
    int              iTryCount = 0;

    m_hstmt = m_hstmtStockLevel;

    while (TRUE)
    {
        try
        {
            rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)"L"(call
tpcc_stocklevel(?,?,?))", SQL_NTS);
            if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                ThrowError(CODBCERR::eExecDirect);

            if ( SQLFetch(m_hstmt) == SQL_ERROR )
                ThrowError(CODBCERR::eFetch);

```

```

        SQLFreeStmt(m_hstmt, SQL_CLOSE);

        m_txn.StockLevel.exec_status_code = eOK;
        bTeak;
    }
    catch (COBDCERR *e)
    {
        if (!(e->m_bDeadLock) || (++iTryCount > iMaxRetries))
            throw;

        // hit deadlock; backoff for increasingly longer

        delete e;
        Sleep(10 * iTryCount);
    }
}

// if (iTryCount)
// throw new CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_ODBC::InitNewOrderParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmtNewOrder) !=
SQL_SUCCESS
        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc, &m_descNewOrderCols1)
!= SQL_SUCCESS
        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc, &m_descNewOrderCols2)
!= SQL_SUCCESS
    )
        ThrowError(COBCERR::eAllocHandle);

    m_hstmt = m_hstmtNewOrder;

    if ( SQLSetStmtAttrW(m_hstmt, SQL_ATTR_APP_ROW_DESC, m_descNewOrderCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )
        ThrowError(COBCERR::eSetStmtAttr);

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.NewOrder.w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.NewOrder.d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_txn.NewOrder.c_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.NewOrder.o_ol_cnt, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.NewOrder.o_all_local, 0, NULL) !=
SQL_SUCCESS
    )
        ThrowError(COBCERR::eBindParam);

    for (int j=0; j<MAX_OL_NEW_ORDER_ITEMS; j++)
    {
        if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SLONG, SQL_INTEGER, 0, 0, &m_txn.NewOrder.OL[j].ol_i_id, 0, NULL) !=
SQL_SUCCESS
            || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SSHORT, SQL_SMALLINT, 0, 0, &m_txn.NewOrder.OL[j].ol_supply_w_id, 0, NULL) !=
SQL_SUCCESS

```

```

        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_SSHORT, SQL_SMALLINT, 0, 0, &m_txn.NewOrder.OL[j].ol_quantity, 0, NULL) !=
SQL_SUCCESS
    )
        ThrowError(COBCERR::eBindParam);
    }

#ifdef new_order_strstr
    // set the bind offset pointer
    if ( SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_BIND_OFFSET_PTR,
&m_BindOffset, SQL_IS_POINTER ) != SQL_SUCCESS )
        ThrowError(COBCERR::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(COBCERR::eBindCol);
#else
    // prototype to eliminate patindex in server; shift work to client
    i = 0;
    if ( SQLBindCol(m_hstmt, ++i, SQL_C_CHAR, &m_ol_i_name,
sizeof(m_ol_i_name), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT, &m_ol_stock, 0, NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR, &m_i_data,
sizeof(m_i_data), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR, &m_s_data,
sizeof(m_s_data), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE, &m_ol_i_price, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE, &m_ol_amount, 0, NULL)
!= SQL_SUCCESS
    )
        ThrowError(COBCERR::eBindCol);
#endif

    // 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(COBCERR::eSetStmtAttr);

    i = 0;
    if ( SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE, &m_txn.NewOrder.w_tax, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.d_tax, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.NewOrder.o_id, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.c_last, sizeof(m_txn.NewOrder.c_last), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.c_discount, 0, NULL) != SQL_SUCCESS

```

```

        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.c_credit, sizeof(m_txn.NewOrder.c_credit), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.NewOrder.o_entry_d, 0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG, &m_no_commit_flag,
0, NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindCol);
}

void CTPCC_ODBC::NewOrder()
{
    int          rc;          i;
    RETCODE     rc;          i;
    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;
    wcsncpy( &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
            break;
        }
    }

    while (TRUE)
    {
        try
        {
            m_BindOffset = 0;
            rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)szSqlTemplate,
SQL_NTS);

            if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                ThrowError(CODBCERR::eExecDirect);

            // Get order line results

```

```

        m_txn.NewOrder.total_amount = 0;
        for ( i = 0; i < m_txn.NewOrder.o_ol_cnt; i++)
        {
#ifdef new_order_strstr
            // set the bind offset value...
            m_BindOffset = i *
sizeof(m_txn.NewOrder.OL[0]);

            if ( SQLFetch(m_hstmt) == SQL_ERROR)
                ThrowError(CODBCERR::eFetch);
#else
            if ( SQLFetch(m_hstmt) == SQL_ERROR)
                ThrowError(CODBCERR::eFetch);

            strcpy( m_txn.NewOrder.OL[i].ol_i_name,
m_ol_i_name );

            if ( strstr(m_i_data, "ORIGINAL") != NULL &&
strstr(m_s_data, "ORIGINAL") != NULL )
                m_txn.NewOrder.OL[i].ol_brand_generic[0] = 'B';
            else
                m_txn.NewOrder.OL[i].ol_brand_generic[0] = 'G';
            m_txn.NewOrder.OL[i].ol_brand_generic[1] =
0;

            m_txn.NewOrder.OL[i].ol_stock
= m_ol_stock;
            m_txn.NewOrder.OL[i].ol_i_price
= m_ol_i_price;
            m_txn.NewOrder.OL[i].ol_amount
= m_ol_amount;
#endif

            // 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 (COBDCERR *e)
        {
            if (!(e->m_bDeadLock) || (++iTryCount > iMaxRetries))
                throw;

            // hit deadlock; backoff for increasingly longer
            delete e;
            Sleep(10 * iTryCount);
        }
    }

    // if (iTryCount)
    // throw new CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_ODBC::InitPaymentParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmtPayment) !=
SQL_SUCCESS )
        ThrowError(COBDCERR::eAllocHandle);

    m_hstmt = m_hstmtPayment;

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Payment.w_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Payment.c_w_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_DOUBLE,
SQL_NUMERIC, 6, 2, &m_txn.Payment.h_amount, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.Payment.d_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.Payment.c_d_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_txn.Payment.c_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, sizeof(m_txn.Payment.c_last), 0, &m_txn.Payment.c_last,
sizeof(m_txn.Payment.c_last), NULL) != SQL_SUCCESS
)
    ThrowError(COBDCERR::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(COBDCERR::eBindCol);
}

void CTPCC_ODBC::Payment()
{
    RETCODE rc;

```

```

int                iTryCount = 0;

m_hstmt = m_hstmtPayment;

if (m_txn.Payment.c_id != 0)
    m_txn.Payment.c_last[0] = 0;

while (TRUE)
{
    try
    {
        rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)L"call
tpcc_payment(?,?,?,?)" , SQL_NTS);
        if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
            ThrowError(CODBCERR::eExecDirect);

        if ( SQLFetch(m_hstmt) == SQL_ERROR)
            ThrowError(CODBCERR::eFetch);

        SQLFreeStmt(m_hstmt, SQL_CLOSE);

        if (m_txn.Payment.c_id == 0)
            throw new CTPCC_ODBC_ERR(
CTPCC_ODBC_ERR::ERR_INVALID_CUST );
        else
            m_txn.Payment.exec_status_code = eOK;

        break;
    }
    catch (CODBCERR *e)
    {
        if (!!(e->m_bDeadLock) || (++iTryCount > iMaxRetries))
            throw;

        // hit deadlock; backoff for increasingly longer
        period

        delete e;
        Sleep(10 * iTryCount);
    }
}

//      if (iTryCount)
//          throw new CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_ODBC::InitOrderStatusParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmtOrderStatus) !=
SQL_SUCCESS
        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols1) != SQL_SUCCESS
        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols2) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtOrderStatus;

    if ( SQLSetStmtAttrW(m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols1, SQL_IS_POINTER ) != SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);
}

```

```

int i = 0;
if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.OrderStatus.w_id, 0, NULL) != SQL_SUCCESS
    || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.OrderStatus.d_id, 0, NULL) != SQL_SUCCESS
    || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_txn.OrderStatus.c_id, 0, NULL) != SQL_SUCCESS
    || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, sizeof(m_txn.OrderStatus.c_last), 0, &m_txn.OrderStatus.c_last,
sizeof(m_txn.OrderStatus.c_last), NULL) != SQL_SUCCESS
)
    ThrowError(CODBCERR::eBindParam);

// configure block cursor
if ( SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_BIND_TYPE,
(SQLPOINTER)sizeof(m_txn.OrderStatus.OL[0]), 0) != SQL_SUCCESS
    || SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROWS_FETCHED_PTR,
&m_RowsFetched, 0) != SQL_SUCCESS
)
    ThrowError(CODBCERR::eSetStmtAttr);

i = 0;
if ( SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.OrderStatus.OL[0].ol_supply_w_id, 0, NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.OL[0].ol_i_id, 0, NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.OrderStatus.OL[0].ol_quantity, 0, NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.OrderStatus.OL[0].ol_amount, 0, NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.OrderStatus.OL[0].ol_delivery_d, 0, NULL) != SQL_SUCCESS
)
    ThrowError(CODBCERR::eBindCol);

if ( SQLSetStmtAttrW(m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols2, SQL_IS_POINTER ) != SQL_SUCCESS )
    ThrowError(CODBCERR::eSetStmtAttr);

i = 0;
if ( SQLBindCol(m_hstmt, ++i, SQL_C_SLONG, &m_txn.OrderStatus.c_id, 0,
NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_last, sizeof(m_txn.OrderStatus.c_last), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_first, sizeof(m_txn.OrderStatus.c_first), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_middle, sizeof(m_txn.OrderStatus.c_middle), NULL) !=
SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.OrderStatus.o_entry_d, 0, NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.OrderStatus.o_carrier_id, 0, NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.OrderStatus.c_balance, 0, NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.o_id, 0, NULL) != SQL_SUCCESS
)
    ThrowError(CODBCERR::eBindCol);
}

void CTPCC_ODBC::OrderStatus()

```



```

{
    int                                     iTryCount = 0;
    RETCODE                                 rc;

    m_hstmt = m_hstmtOrderStatus;

    if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols1, SQL_IS_POINTER ) != SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);

    if ( m_txn.OrderStatus.c_id != 0 )
        m_txn.OrderStatus.c_last[0] = 0;

    while (TRUE)
    {
        try
        {
            // configure block cursor
            if ( SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)1, 0) != SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);

            rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)L"call
tpcc_orderstatus(?,?,?,?)", SQL_NTS);
            if ( ((rc == SQL_SUCCESS_WITH_INFO) && (m_RowsFetched
!= 0)) || (rc == SQL_ERROR) )
                ThrowError(CODBCERR::eExecDirect);

            // configure block cursor
            if ( SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)MAX_OL_ORDER_STATUS_ITEMS, 0) != SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);

            rc = SQLFetchScroll( m_hstmt, SQL_FETCH_NEXT, 0 );
            if ( ((rc == SQL_SUCCESS_WITH_INFO) && (m_RowsFetched
!= 0)) || (rc == SQL_ERROR) )
                ThrowError(CODBCERR::eFetchScroll);

            m_txn.OrderStatus.o_ol_cnt = (short)m_RowsFetched;

            if ( m_txn.OrderStatus.o_ol_cnt != 0 )
            {
                if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols2, SQL_IS_POINTER ) != SQL_SUCCESS )
                    ThrowError(CODBCERR::eSetStmtAttr);

                if ( SQLMoreResults(m_hstmt) == SQL_ERROR )
                    ThrowError(CODBCERR::eMoreResults);

                if ( (rc = SQLFetch(m_hstmt)) == SQL_ERROR )
                    ThrowError(CODBCERR::eFetch);
            }

            SQLFreeStmt(m_hstmt, SQL_CLOSE);

            if ( m_txn.OrderStatus.o_ol_cnt == 0 )
                throw new CTPCC_ODBC_ERR(
CTPCC_ODBC_ERR::ERR_NO_SUCH_ORDER );
            else if ( m_txn.OrderStatus.c_id == 0 &&
m_txn.OrderStatus.c_last[0] == 0 )

```

```

                throw new CTPCC_ODBC_ERR(
CTPCC_ODBC_ERR::ERR_INVALID_CUST );
            else
                m_txn.OrderStatus.exec_status_code = eOK;

            break;
        }
        catch (CODBCERR *e)
        {
            if ((!e->m_bDeadLock) || (++iTryCount > iMaxRetries))
                throw;

            // hit deadlock; backoff for increasingly longer
            period

            delete e;
            Sleep(10 * iTryCount);
        }
    }

    // if (iTryCount)
    // throw new CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_ODBC::InitDeliveryParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmtDelivery) !=
SQL_SUCCESS )
        ThrowError(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtDelivery;

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Delivery.w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Delivery.o_carrier_id, 0, NULL) != SQL_SUCCESS
        )
        ThrowError(CODBCERR::eBindParam);

    for (i=0;i<10;i++)
    {
        if ( SQLBindCol(m_hstmt, (UWORD)(i+1), SQL_C_SLONG,
&m_txn.Delivery.o_id[i], 0, NULL) != SQL_SUCCESS )
            ThrowError(CODBCERR::eBindCol);
    }
}

void CTPCC_ODBC::Delivery()
{
    RETCODE                                 rc;
    int                                     iTryCount = 0;

    m_hstmt = m_hstmtDelivery;

    while (TRUE)
    {
        try
        {
            rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)L"call
tpcc_delivery(?,?,?)", SQL_NTS);
            if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                ThrowError(CODBCERR::eExecDirect);

```

```

        if ( SQLFetch(m_hstmt) == SQL_ERROR )
            ThrowError(CODBCERR::eFetch);

        SQLFreeStmt(m_hstmt, SQL_CLOSE);
        m_txn.Delivery.exec_status_code = eOK;
        break;
    }
    catch (CODBCERR *e)
    {
        if (!(e->m_bDeadLock) || (++iTryCount > iMaxRetries))
            throw;

        // hit deadlock; backoff for increasingly longer
        period

        delete e;
        Sleep(10 * iTryCount);
    }

    //
    // if (iTryCount)
    //     throw new CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
    // iTryCount);
    //
}

```

tpcc_odbc.h

```

/* FILE: TPC_C_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.
#ifdef 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 TPC_C_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
retry count on deadlock                m_MaxRetries;           //

        SQLHENV                m_henv;           //
ODBC environment handle
        SQLHDBC                m_hdbc;
        SQLHSTMT                m_hstmt;           // the current hstmt

        SQLHSTMT                m_hstmtNewOrder;
        SQLHSTMT                m_hstmtPayment;
        SQLHSTMT                m_hstmtDelivery;
        SQLHSTMT                m_hstmtOrderStatus;
        SQLHSTMT                m_hstmtStockLevel;

        SQLHDESC                m_descNewOrderCols1;
        SQLHDESC                m_descNewOrderCols2;
        SQLHDESC                m_descOrderStatusCols1;
        SQLHDESC                m_descOrderStatusCols2;

        // new-order specific fields
        SQLUIINTEGER                m_BindOffset;
        SQLUIINTEGER                m_RowsFetched;
        int                m_no_commit_flag;

#ifdef new_order_strstr
        // for new-order txn;
        // output params
        char                m_ol_i_name[I_NAME_LEN+1];
        double                m_ol_i_price;
        double                m_ol_amount;
        short                m_ol_stock;
        // used locally, but not returned to caller
        char                m_i_data[I_DATA_LEN];
        char                m_s_data[S_DATA_LEN];

#endif

        void ThrowError( COBDCERR::ACTION eAction );

        void InitNewOrderParams();
        void InitPaymentParams();
        void InitDeliveryParams();
        void InitStockLevelParams();
        void InitOrderStatusParams();

        union

```

```

        {
                NEW_ORDER_DATA                NewOrder;
                PAYMENT_DATA                Payment;
                DELIVERY_DATA                Delivery;
                STOCK_LEVEL_DATA                StockLevel;
                ORDER_STATUS_DATA                OrderStatus;
                m_txn;
        }

public:
        CTPCC_ODBC(LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword,
LPCSTR szHost, LPCSTR szDatabase);
        ~CTPCC_ODBC(void);

        inline PNEW_ORDER_DATA                BuffAddr_NewOrder()
        { return &m_txn.NewOrder; };
        inline PPAYMENT_DATA                BuffAddr_Payment()
        { return &m_txn.Payment; };
        inline PDELIVERY_DATA                BuffAddr_Delivery()
        { return &m_txn.Delivery; };
        inline PSTOCK_LEVEL_DATA                BuffAddr_StockLevel()
return &m_txn.StockLevel; };
        inline PORDER_STATUS_DATA                BuffAddr_OrderStatus()
return &m_txn.OrderStatus; };

        void NewOrder                ();
        void Payment                ();
        void Delivery                ();
        void StockLevel                ();
        void OrderStatus                ();

};

// wrapper routine for class constructor
extern "C" DllDecl CTPCC_ODBC* CTPCC_ODBC_new
( LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword, LPCSTR szHost, LPCSTR
szDatabase );

typedef CTPCC_ODBC* (TYPE_CTPCC_ODBC)(LPCSTR, LPCSTR, LPCSTR, LPCSTR, LPCSTR);

trans.h


---


/* FILE: TRANS.H
* Microsoft TPC-C Kit Ver. 4.20.000
* Copyright Microsoft, 1999
* All Rights Reserved
* Version 4.10.000 audited by Richard Gimarc,
Performance Metrics, 3/17/99
* PURPOSE: Header file for TPC-C structure templates.
* Change history:
* 4.20.000 - updated rev number to match kit
*/
#pragma once

// String length constants
#define SERVER_NAME_LEN 20
#define DATABASE_NAME_LEN 20
#define USER_NAME_LEN 20
#define PASSWORD_LEN 20

```

```

#define TABLE_NAME_LEN      20
#define I_DATA_LEN           50
#define I_NAME_LEN           24
#define BRAND_LEN            1
#define LAST_NAME_LEN        16
#define W_NAME_LEN           10
#define ADDRESS_LEN          20
#define STATE_LEN            2
#define ZIP_LEN               9
#define S_DIST_LEN           24
#define S_DATA_LEN           50
#define D_NAME_LEN           10
#define FIRST_NAME_LEN       16
#define MIDDLE_NAME_LEN      2
#define PHONE_LEN            16
#define DATETIME_LEN         30
#define CREDIT_LEN           2
#define C_DATA_LEN           250
#define H_DATA_LEN           24
#define DIST_INFO_LEN        24
#define MAX_OL_NEW_ORDER_ITEMS 15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN           25
#define OL_DIST_INFO_LEN     24

// TIMESTAMP_STRUCT is provided by the ODBC header file sqltypes.h, but is not
// available
// when compiling with dblib, so redefined here. Note: we are using the symbol
// "SQLTYPES"
// (declared in sqltypes.h) as a way to determine if TIMESTAMP_STRUCT has been
// declared.
#ifdef __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       /* SQLINTEGER */ fraction;
} TIMESTAMP_STRUCT;
#endif

// possible values for exec_status_code after transaction completes
enum EXEC_STATUS
{
    eOK,                // 0      "Transaction committed."
    eInvalidItem,       // 1      "Item number is not valid."
    eDeliveryFailed     // 2      "Delivery Post Failed."
};

// transaction structures
typedef struct
{
    // input params
    short                ol_supply_w_id;
    long                 ol_i_id;
    short                ol_quantity;

    // output params
    char                 ol_i_name[I_NAME_LEN+1];

```

```

    char                 ol_brand_generic[BRAND_LEN+1];
    double               ol_i_price;
    double               ol_amount;
    short                ol_stock;
} OL_NEW_ORDER_DATA;

typedef struct
{
    // input params
    short                w_id;
    short                d_id;
    long                 c_id;
    short                o_ol_cnt;

    // output params
    EXEC_STATUS          exec_status_code;
    char                 c_last[LAST_NAME_LEN+1];
    char                 c_credit[CREDIT_LEN+1];
    double               c_discount;
    double               w_tax;
    double               d_tax;
    long                 o_id;
    short                o_commit_flag;
    TIMESTAMP_STRUCT     o_entry_d;
    short                o_all_local;
    double               total_amount;
    OL_NEW_ORDER_DATA    OL[MAX_OL_NEW_ORDER_ITEMS];
} NEW_ORDER_DATA, *PNEW_ORDER_DATA;

typedef struct
{
    // input params
    short                w_id;
    short                d_id;
    long                 c_id;
    short                c_d_id;
    short                c_w_id;
    double               h_amount;
    char                 c_last[LAST_NAME_LEN+1];

    // output params
    EXEC_STATUS          exec_status_code;
    TIMESTAMP_STRUCT     h_date;
    char                 w_street_1[ADDRESS_LEN+1];
    char                 w_street_2[ADDRESS_LEN+1];
    char                 w_city[ADDRESS_LEN+1];
    char                 w_state[STATE_LEN+1];
    char                 w_zip[ZIP_LEN+1];
    char                 d_street_1[ADDRESS_LEN+1];
    char                 d_street_2[ADDRESS_LEN+1];
    char                 d_city[ADDRESS_LEN+1];
    char                 d_state[STATE_LEN+1];
    char                 d_zip[ZIP_LEN+1];
    char                 c_first[FIRST NAME_LEN+1];
    char                 c_middle[MIDDLE NAME_LEN + 1];
    char                 c_street_1[ADDRESS_LEN+1];
    char                 c_street_2[ADDRESS_LEN+1];
    char                 c_city[ADDRESS_LEN+1];
    char                 c_state[STATE_LEN+1];
    char                 c_zip[ZIP_LEN+1];
    char                 c_phone[PHONE_LEN+1];
    TIMESTAMP_STRUCT     c_since;
    char                 c_credit[CREDIT_LEN+1];
    double               c_credit_lim;

```

```

        double                c_discount;
        double                c_balance;
        char                  c_data[200+1];
} PAYMENT_DATA, *PPAYMENT_DATA;

typedef struct
{
    long                    ol_i_id;
    short                   ol_supply_w_id;
    short                   ol_quantity;
    double                  ol_amount;
    TIMESTAMP_STRUCT        ol_delivery_d;
} OL_ORDER_STATUS_DATA;

typedef struct
{
    // input params
    short                   w_id;
    short                   d_id;
    long                    c_id;
    char                    c_last[LAST_NAME_LEN+1];

    // output params
    EXEC_STATUS              exec_status_code;
    char                    c_first[FIRST_NAME_LEN+1];
    char                    c_middle[MIDDLE_NAME_LEN+1];
    double                  c_balance;
    long                    o_id;
    TIMESTAMP_STRUCT        o_entry_d;
    short                   o_carrier_id;
    OL_ORDER_STATUS_DATA    OL[MAX_OL_ORDER_STATUS_ITEMS];
    short                   o_ol_cnt;
} ORDER_STATUS_DATA, *PORDER_STATUS_DATA;

typedef struct
{
    // input params
    short                   w_id;
    short                   o_carrier_id;

    // output params
    EXEC_STATUS              exec_status_code;
    SYSTEMTIME               queue_time;
    long                     o_id[10]; // id's of
delivered orders for districts 1 to 10
} DELIVERY_DATA, *PDELIVERY_DATA;

//This structure is used for posting delivery transactions and for writing them to
the delivery server.
typedef struct _DELIVERY_TRANSACTION
{
    SYSTEMTIME               queue; //time delivery
transaction queued
    short                   w_id; //delivery warehouse
    short                   o_carrier_id; //carrier id
} DELIVERY_TRANSACTION;

typedef struct
{
    // input params
    short                   w_id;
    short                   d_id;
    short                   threshold;

    // output params

```

```

        EXEC_STATUS          exec_status_code;
        long                 low_stock;
} STOCK_LEVEL_DATA, *PSTOCK_LEVEL_DATA;

```

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;
};

```

txnlog.h

```

/* FILE: TXNLOG.H
 * Microsoft TPC-C Kit Ver. 4.10.000
 * not yet audited
 *
 * PURPOSE: Header file for txn log class
 * Copyright Microsoft, 1999
 * All Rights Reserved
 *
 */

```

```

#pragma once

typedef struct _TXN_NEWORDER
{
    BYTE    OL_Count;           //range 0 to 31
    BYTE    OL_Remote_Count;    //range 0 to 31
    WORD    c_id;
    int     o_id;
} TXN_NEWORDER;

typedef struct _TXN_PAYMENT
{
    BYTE    CustByName;
    BYTE    IsRemote;
} TXN_PAYMENT;

typedef struct _TXN_ORDERSTATUS
{
    BYTE    CustByName;
} TXN_ORDERSTATUS;

typedef union _TXN_DETAILS
{
    TXN_NEWORDER    NewOrder;
    TXN_PAYMENT     Payment;
    TXN_ORDERSTATUS OrderStatus;
} TXN_DETAILS;

// Common header for all records in txn log. The TxnType field is
// a switch which identifies the particular variant.
#define TXN_REC_TYPE_CONTROL    1
#define TXN_REC_TYPE_TPCC      2 // replaces
TRANSACTION_TYPE_TPCC
#define TXN_REC_TYPE_TPCC_DELIV_DEF    3

typedef struct _TXN_RECORD_HEADER
{
    JULIAN_TIME    TxnStartT0; // start of
    BYTE    TxnType;           // one of TXN_REC_TYPE_*
    BYTE    TxnSubType;        // depends on
} 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_REC_TYPE_CONTROL    TxnType; // =
    TXN_REC_TYPE_CONTROL    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
    TxnType
    BYTE    TxnSubType;        // depends on
    // 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
    WORD    w_id;             // warehouse ID
    BYTE    d_id;             // assigned district ID
    for this thread
    BYTE    d_id_ThisTxn;     // district ID chosen for this
    particular
    BYTE    TxnStatus;        // completion status for
    txn to indicate errors
    BYTE    reserved;         // for word alignment
    TXN_DETAILS    TxnDetails; //
} TXN_RECORD_TPCC, *PTXN_RECORD_TPCC;

// TPC-C Deferred Delivery Txn Record Layout:
//
//Incorporating delivery transaction information into the above
//structure would increase the size of TXN_DETAILS from 8 to 42 bytes.
//Hence, we store delivery transaction details in a separate structure.
//
typedef struct _TXN_RECORD_TPCC_DELIV_DEF
{
    // common header; must exactly match TXN_RECORD_HEADER

```

```

        JULIAN_TIME      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)
        WORD        w_id;                  // warehouse ID
        BYTE        TxnStatus;             // completion status for
txn to indicate errors
        BYTE        reserved;              // for word alignment
        short       o_carrier_id;         // carrier id
        long        o_id[10];             // returned delivery transaction
ids
    } TXN_RECORD_TPCC_DELIV_DEF, *PTXN_RECORD_TPCC_DELIV_DEF;

#define TXN_LOG_VERSION 1
#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

        // the record map provides a fast way to get close to a
particular timestamp in a sorted log file.
//
//
//
// timestamp of record
// int         iPos;
// byte position in file
//
//
// RecMap[RecMapSize];
// #define RecMapSize 200
} TXN_LOG_HEADER, *PTXN_LOG_HEADER;

```

```

#define READ_BUFFER_SIZE 64*1024
#define WRITE_BUFFER_SIZE 8*1024

#define NUM_READ_BUFFERS 1
#define NUM_WRITE_BUFFERS 2
#define MAX_NUM_BUFFERS 2

// flags passed in to the constructor
#define TXN_LOG_WRITE 0x01
#define TXN_LOG_READ 0x02
#define TXN_LOG_SORTED 0x04

#define TXN_LOG_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.
    int            iNextRec;
//when reading, ordinal value of next record

        // A "save point" is remembered each time GetNextRecord is
called with a start time specified.
        // The next time it is called, if start time is after the save
point, we start scanning from the
        // save point. This is particularly useful in FindBestInterval,
where the log is scanned repeatedly.
        JULIAN_TIME SavePtTime;
        int         iSavePtFilePointer;
        int         iSavePtNextRec;

        JULIAN_TIME lastTS;
//when writing sorted output, used to verify records are sorted
        BOOL        bWrite;
//writing log file

        BOOL        bLogSorted;
// is log file sorted? applies to both input and output
        JULIAN_TIME BeginTxnTS;
// timestamp of first (lowest) txn start
        JULIAN_TIME EndTxnTS; //
timestamp of last (highest) txn completion time
        int         iRecCount;
        // number of records in log file

        BYTE        *pCurrent;
//ptr to current buffer
        BYTE        *pBuffer[MAX_NUM_BUFFERS];

```

```

        PTXN_RECORD_HEADER *TxnArray;          //transaction
record pointer array for sort

        DWORD          dwError;
        HANDLE         hTxnFile;
//handle to log file
        HANDLE         hMapFile;
//map file used when sorting the log
        HANDLE         hIoComplete;
//event to signify that there are no pending IOs
        HANDLE         hLogFileIo;
//event to signal the IO thread to write the inactive buffer

        Spinlock Spin;
//spin lock to protect the txn log file buffers

        int Write(BYTE *ptr, DWORD Size);
        static void LogFileIO(CTxnLog *);

public:
        CTxnLog::CTxnLog(LPCTSTR szFileName, DWORD dwOpts);
        ~CTxnLog(void);

        int WriteToLog(PTXN_RECORD_TPCC pTxnRcrd);
        int WriteToLog(PTXN_RECORD_TPCC_DELIV_DEF pTxnRcrd);
        int WriteToLog(PTXN_RECORD_CONTROL pCtrlRec);
        int WriteToLog(PTXN_RECORD_HEADER pCtrlRec);

        int WriteCtrlRecToLog(BYTE SubType, LPTSTR lpStr, DWORD dwLen);

        void CloseTransactionLogFile(void);

        PTXN_RECORD_HEADER GetNextRecord(BOOL bSkipCtrlRecs = FALSE);
        PTXN_RECORD_HEADER GetNextRecord(JULIAN_TIME SeekTimeT0, BOOL
bSkipCtrlRecs = FALSE);

        int Sort(void);
        PTXN_RECORD_HEADER GetSortedRecord(int index);

        inline BOOL IsSorted(void) { return bLogSorted; };
        inline JULIAN_TIME BeginTS(void) { return BeginTxnTS; };
        inline JULIAN_TIME EndTS(void) { return EndTxnTS; };
        inline int RecordCount(void) { return iRecCount; };

};

class CTXNLOG_ERR : public CBaseErr
{
public:
        enum CTXNLOG_ERRS
        {
                ERR_BAD_FILE_FORMAT,          // "File
format is invalid."
                ERR_UNKNOWN_LOG_VERSION,      // "Log file version is
unknown."
                ERR_BROKEN_LOG_FILE,         // "Log file
is broken."
                ERR_LOG_NOT_SORTED,         // "Log file
is not sorted"
                ERR_INVALID_TIME_SEQ,       // "Internal
Error: Record Time Sequence invalid."
        };

        CTXNLOG_ERR(int iErr) : CBaseErr(iErr) {};

```

```

int ErrorType() {return ERR_TYPE_TXNLOG;};

char *ErrorText()
{
        static char *szMsgs[] = {
                "File format is invalid.",
                "Log file version is unknown.",
                "Log file is broken.",
                "Log file is not sorted",
                "Internal Error: Record Time Sequence
invalid.",
                ""
        };

        for(int i = 0; szMsgs[i][0]; i++)
        {
                if ( m_idMsg == i )
                        break;
        }

        return(szMsgs[i][0] ? szMsgs[i] : ERR_UNKNOWN);
};

```


Appendix B:

Database Design

The TPC-C database was created with the following Transact-SQL scripts:

NOTE: Unless otherwise noted, these SQL scripts were run on each node of the configuration without modification.

backup.sql

```
-- File:      BACKUP.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.20
--           Copyright Microsoft, 1999
-- Purpose:   Creates backup of tpcc database

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

dump database tpcc to tpccback1, tpccback2, tpccback3, tpccback4 with init, stats =
1, blocksize=4096

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

backupdev.sql

```
-- File:      BACKUPDEVB.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.20
--           Copyright Microsoft, 1999
-- Purpose:   Creates tpcc database Backup Devices

use master
go

-- create backup devices

exec sp_addumpdevice 'disk','tpccback1','Y:\tpccback1.dmp'
exec sp_addumpdevice 'disk','tpccback2','Y:\tpccback2.dmp'
exec sp_addumpdevice 'disk','tpccback3','Z:\tpccback3.dmp'
exec sp_addumpdevice 'disk','tpccback4','Z:\tpccback4.dmp'

go
```

createdb.sql

```
-- File:      CREATEDB.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.20
--           Copyright Microsoft, 1999
-- Purpose:   Creates tpcc database and backup files

use master
go

-- Create temporary table for timing

if exists ( select name from sysobjects where name = 'tpcc_timer' )
drop table tpcc_timer
go

create table tpcc_timer
(
    start_date          char(30),
    end_date            char(30)
)

insert into tpcc_timer values (0,0)
go

-- Store starting time

update tpcc_timer
set start_date = (select convert(char(30), getdate(),9))
go

-- create main database files

CREATE DATABASE tpcc
ON PRIMARY
(
    NAME          = MSSQL70_tpcc_root,
    FILENAME     = "T:\MSSQL70_tpcc_root.mdf",
    SIZE         = 8MB,
    FILEGROWTH   = 0),
FILEGROUP misc_fg
(
    NAME          = misc1,
    FILENAME     = "J:",
    SIZE         = 20000MB,
    FILEGROWTH   = 0),
(
    NAME          = misc2,
    FILENAME     = "K:",
    SIZE         = 20000MB,
    FILEGROWTH   = 0),
(
    NAME          = misc3,
    FILENAME     = "L:",
    SIZE         = 20000MB,
    FILEGROWTH   = 0),
(
    NAME          = misc4,
    FILENAME     = "M:",
    SIZE         = 20000MB,
    FILEGROWTH   = 0),
FILEGROUP big_fg
(
    NAME          = big1,
    FILENAME     = "F:",
    SIZE         = 30500MB,
    FILEGROWTH   = 0),
```

```

(
    NAME                = big2,
    FILENAME = "G:",
    SIZE                = 30500MB,
    FILEGROWTH         = 0),
(
    NAME                = big3,
    FILENAME = "H:",
    SIZE                = 30500MB,
    FILEGROWTH         = 0),
(
    NAME                = big4,
    FILENAME = "I:",
    SIZE                = 30500MB,
    FILEGROWTH         = 0)
LOG ON
(
    NAME                =MSSQL70_tpcc_log,
    FILENAME = "E:",
    SIZE                =60000MB,
    FILEGROWTH         =0)
go

-- Store ending time
update tpcc_timer
set end_date = (select convert(char(30), getdate(),9))
go

select "Elapsed time (in seconds): ", datediff(second,(select start_date from
tpcc_timer),(select end_date from tpcc_timer))

-- remove temporary table

if exists ( select name from sysobjects where name = 'tpcc_timer' )
drop table tpcc_timer
go

```

dbopt1.sql

```

-- File:      DBOPT1.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.10
--           Copyright Microsoft, 1999
-- Purpose:   Sets database options for data load

use master
go

exec sp_dboption tpcc,'select into/bulkcopy',true
exec sp_dboption tpcc,'trunc. log on chkpt.',true
exec sp_dboption tpcc,'torn page detection',false
go

use tpcc
go

checkpoint
go

```

dbopt2.sql

```

-- File:      DBOPT2_SHILOH.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.10a
--           Copyright Microsoft, 1999
-- Purpose:   Resets database options after data load (For Shiloh)

```

```

use master
go

sp_dboption tpcc,'select ',false
go

sp_dboption tpcc,'trunc. ',false
go

use tpcc
go

checkpoint
go

sp_configure allow,1
go

reconfigure with override
go

/*                                     */
/* Set option values for user-defined indexes */
/*                                     */

sp_indexoption 't_customer', 'DisallowPageLocks', TRUE
go
sp_indexoption 't_district', 'DisallowPageLocks', TRUE
go
sp_indexoption 't_warehouse', 'DisallowPageLocks', TRUE
go
sp_indexoption 't_stock', 'DisallowPageLocks', TRUE
go
sp_indexoption 't_order_line', 'DisallowRowLocks', TRUE
go
sp_indexoption 't_orders', 'DisallowRowLocks', TRUE
go
sp_indexoption 't_new_order', 'DisallowRowLocks', TRUE
go
sp_indexoption 't_item', 'DisallowRowLocks', TRUE
go
sp_indexoption 't_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("t_warehouse") = id or
         object_id("t_district")   = id or
         object_id("t_customer")   = id or
         object_id("t_stock")      = id or
         object_id("t_orders")     = id or
         object_id("t_order_line") = id or
         object_id("t_history")    = id or
         object_id("t_new_order")  = id or
         object_id("t_item")       = id

order    by lockflags asc
go

sp_configure allow,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 "t_district",      "pintable",true
exec sp_tableoption "t_warehouse",    "pintable",true
exec sp_tableoption "t_new_order",    "pintable",true
exec sp_tableoption "t_item",         "pintable",true
go

```

idxcuscl.sql

```

-- File:      IDXCUSCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.10
--            Copyright Microsoft, 1999
-- Purpose:   Creates clustered index on customer table

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
set QUOTED_IDENTIFIER off
go

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists (select o.name from sysconstraints c, sysobjects o
where c.id = object_id('t_customer') and c.constid = o.id and o.name =
'customer_clPK' )
    alter table t_customer drop constraint customer_clPK

alter table t_customer add constraint customer_clPK primary key clustered (c_w_id,
c_d_id, c_id)
    on big_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

```

```
go
```

idxcusnc.sql

```

-- File:      IDXCUSNC.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.10
--            Copyright Microsoft, 1999
-- Purpose:   Creates non-clustered index on customer table

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
set QUOTED_IDENTIFIER off
go

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'customer_nc1' )
    drop index t_customer.customer_nc1

create unique nonclustered index customer_nc1 on t_customer(c_w_id, c_d_id, c_last,
c_first, c_id)
    on big_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxdiscl.sql

```

-- File:      IDXDISCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.10
--            Copyright Microsoft, 1999
-- Purpose:   Creates clustered index on district table

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
set QUOTED_IDENTIFIER off
go

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists (select o.name from sysconstraints c, sysobjects o
where c.id = object_id('t_district') and c.constid = o.id and o.name =
'district_clPK' )

```

```

alter table t_district drop constraint district_clPK

alter table t_district add constraint district_clPK primary key clustered (d_w_id,
d_id)
        with fillfactor=100
        on misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxitmcl.sql

```

-- File:      IDXITMCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.10
--           Copyright Microsoft, 1999
-- Purpose:   Creates clustered index on item table

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
set QUOTED_IDENTIFIER off
go

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'item_cl' )
        drop index t_item.item_cl

create unique clustered index item_cl on t_item(i_id)
        on misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxnodcl.sql

```

-- File:      IDXNODCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.10
--           Copyright Microsoft, 1999
-- Purpose:   Creates clustered index on new_order table

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
set QUOTED_IDENTIFIER off
go

use tpcc
go

```

```

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists (select o.name from sysconstraints c, sysobjects o
where c.id = object_id('t_new_order') and c.constid = o.id and o.name =
't_new_order_clPK' )
        alter table t_new_order drop constraint new_order_clPK

alter table t_new_order add constraint new_order_clPK primary key clustered
(no_w_id, no_d_id, no_o_id)
        on misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxodlcl.sql

```

-- File:      IDXODLCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.10
--           Copyright Microsoft, 1999
-- Purpose:   Creates clustered index on order_line table

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
set QUOTED_IDENTIFIER off
go

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists (select o.name from sysconstraints c, sysobjects o
where c.id = object_id('t_order_line') and c.constid = o.id and o.name =
'order_line_clPK' )
        alter table t_order_line drop constraint order_line_clPK

alter table t_order_line add constraint order_line_clPK primary key clustered
(ol_w_id, ol_d_id, ol_o_id, ol_number)
        on misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxordcl.sql

```

-- File:      IDXORDCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.10
--           Copyright Microsoft, 1999
-- Purpose:   Creates clustered index on orders table

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
set QUOTED_IDENTIFIER off
go

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists (select o.name from sysconstraints c, sysobjects o
where c.id = object_id('t_orders') and c.constid = o.id and o.name = 'orders_c1PK' )
    alter table t_orders drop constraint orders_c1PK

alter table t_orders add constraint orders_c1PK primary key clustered (o_w_id,
o_d_id, o_id)
    on misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxordnc.sql

```

-- File:      IDXORDNC.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.10
--           Copyright Microsoft, 1999
-- Purpose:   Creates non-clustered index on orders table

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
set QUOTED_IDENTIFIER off
go

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'orders_ncl' )
    drop index t_orders.orders_ncl

create index orders_ncl on t_orders(o_w_id, o_d_id, o_c_id, o_id)
    on misc_fg

select @enddate = getdate()

```

```

select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxstkcl.sql

```

-- File:      IDXSTKCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.10
--           Copyright Microsoft, 1999
-- Purpose:   Creates clustered index on stock table

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
set QUOTED_IDENTIFIER off
go

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists (select o.name from sysconstraints c, sysobjects o
where c.id = object_id('t_stock') and c.constid = o.id and o.name = 'stock_c1PK' )
    alter table t_stock drop constraint stock_c1PK

alter table t_stock add constraint stock_c1PK primary key clustered (s_i_id, s_w_id)
    on big_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxwarcl.sql

```

-- File:      IDXWARCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.10
--           Copyright Microsoft, 1999
-- Purpose:   Creates clustered index on warehouse table

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
set QUOTED_IDENTIFIER off
go

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists (select o.name from sysconstraints c, sysobjects o

```

```

where c.id = object_id('t_warehouse') and c.constid = o.id and o.name =
'warehouse_c1PK' )
alter table t_warehouse drop constraint warehouse_c1PK

alter table t_warehouse add constraint warehouse_c1PK primary key clustered (w_id)
with fillfactor=100
on misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

delivery.sql

```

-- File:      DELIVERY.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.10
--           Copyright Microsoft, 1999
-- Purpose:   Creates delivery transaction stored procedure

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

if exists (select name from sysobjects where name = 'tpcc_delivery' )
drop procedure tpcc_delivery
go

create proc tpcc_delivery    @w_id          int,
                             @o_carrier_id smallint
as

declare @d_id    tinyint,
        @o_id    int,
        @c_id    int,
        @total   numeric(12,2),
        @oid1    int,
        @oid2    int,
        @oid3    int,
        @oid4    int,
        @oid5    int,
        @oid6    int,
        @oid7    int,
        @oid8    int,
        @oid9    int,
        @oid10   int,
        @time    datetime

begin tran d

select    @d_id = 0,
         @time = getdate()

```

```

while (@d_id < 10)
begin

select    @d_id = @d_id + 1,
         @total = 0,
         @o_id = 0

select    top 1
         @o_id = no_o_id
         new_order (serializable uplock)
where     no_w_id = @w_id and
         no_d_id = @d_id
order     by no_o_id asc

if (@@rowcount <> 0)
begin

-- claim the order for this district

delete    new_order
where     no_w_id = @w_id and
         no_d_id = @d_id and
         no_o_id = @o_id

-- set carrier_id on this order (and get customer id)

update    orders
set       o_carrier_id = @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 = @time,
         @total = @total + ol_amount
where     ol_w_id = @w_id and
         ol_d_id = @d_id and
         ol_o_id = @o_id

-- accumulate lineitem amounts for this order into customer

update    customer
set       c_balance = c_balance + @total,
         c_delivery_cnt = c_delivery_cnt + 1
where     c_w_id = @w_id and
         c_d_id = @d_id and
         c_id = @c_id

end

select @oid1 = case @d_id when 1 then @o_id else @oid1 end,
       @oid2 = case @d_id when 2 then @o_id else @oid2 end,
       @oid3 = case @d_id when 3 then @o_id else @oid3 end,
       @oid4 = case @d_id when 4 then @o_id else @oid4 end,
       @oid5 = case @d_id when 5 then @o_id else @oid5 end,
       @oid6 = case @d_id when 6 then @o_id else @oid6 end,
       @oid7 = case @d_id when 7 then @o_id else @oid7 end,
       @oid8 = case @d_id when 8 then @o_id else @oid8 end,
       @oid9 = case @d_id when 9 then @o_id else @oid9 end,

```

```

        @oid10 = case @d_id when 10 then @o_id else @oid10 end

    end

commit tran d

-- return delivery data to client

select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

go

```

neword.sql

```

-- File:      NEWORD.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.30.000
--           Copyright Microsoft, 1999
-- Purpose:   Creates new order transaction stored procedure
--
--           Interface Level: 4.10.000

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

if exists ( select name from sysobjects where name = 'tpcc_neworder' )
    drop procedure tpcc_neworder
go

create proc tpcc_neworder

    @w_id          int,
    @d_id          tinyint,
    @c_id          int,
    @o_ol_cnt      tinyint,
    @o_all_local   tinyint,
    @i_id1 int = 0, @s_w_id1 int =
0, @ol_qty1  smallint = 0,
    @i_id2 int = 0, @s_w_id2 int =
0, @ol_qty2  smallint = 0,
    @i_id3 int = 0, @s_w_id3 int =
0, @ol_qty3  smallint = 0,
    @i_id4 int = 0, @s_w_id4 int =
0, @ol_qty4  smallint = 0,
    @i_id5 int = 0, @s_w_id5 int =
0, @ol_qty5  smallint = 0,
    @i_id6 int = 0, @s_w_id6 int =
0, @ol_qty6  smallint = 0,
    @i_id7 int = 0, @s_w_id7 int =
0, @ol_qty7  smallint = 0,

```

```

    @i_id8 int = 0, @s_w_id8 int =
    @i_id9 int = 0, @s_w_id9 int =
    @i_id10 int = 0, @s_w_id10 int =
    @i_id11 int = 0, @s_w_id11 int =
    @i_id12 int = 0, @s_w_id12 int =
    @i_id13 int = 0, @s_w_id13 int =
    @i_id14 int = 0, @s_w_id14 int =
    @i_id15 int = 0, @s_w_id15 int =

0, @ol_qty8  smallint = 0,
0, @ol_qty9  smallint = 0,
0, @ol_qty10 smallint = 0,
0, @ol_qty11 smallint = 0,
0, @ol_qty12 smallint = 0,
0, @ol_qty13 smallint = 0,
0, @ol_qty14 smallint = 0,
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 available order id and update
-- plus initialize local variables

    update  district
    set     @d_tax          = d_tax,
           @o_id           = d_next_o_id,
           d_next_o_id    = d_next_o_id + 1,
           @li_no         = 0,
           @commit_flag   = 1

    where  d_w_id         = @w_id and
           d_id           = @d_id

-- process orderlines

    while (@li_no < @o_ol_cnt)
    begin

        select @li_no = @li_no + 1

```

```

-- set i_id, s_w_id, and qty for this lineitem
select  @li_id = case @li_no
        when 1 then @i_id1
        when 2 then @i_id2
        when 3 then @i_id3
        when 4 then @i_id4
        when 5 then @i_id5
        when 6 then @i_id6
        when 7 then @i_id7
        when 8 then @i_id8
        when 9 then @i_id9
        when 10 then @i_id10
        when 11 then @i_id11
        when 12 then @i_id12
        when 13 then @i_id13
        when 14 then @i_id14
        when 15 then @i_id15
        end,

        @li_s_w_id = case @li_no
        when 1 then @s_w_id1
        when 2 then @s_w_id2
        when 3 then @s_w_id3
        when 4 then @s_w_id4
        when 5 then @s_w_id5
        when 6 then @s_w_id6
        when 7 then @s_w_id7
        when 8 then @s_w_id8
        when 9 then @s_w_id9
        when 10 then @s_w_id10
        when 11 then @s_w_id11
        when 12 then @s_w_id12
        when 13 then @s_w_id13
        when 14 then @s_w_id14
        when 15 then @s_w_id15
        end,

        @li_qty = case @li_no
        when 1 then @ol_qty1
        when 2 then @ol_qty2
        when 3 then @ol_qty3
        when 4 then @ol_qty4
        when 5 then @ol_qty5
        when 6 then @ol_qty6
        when 7 then @ol_qty7
        when 8 then @ol_qty8
        when 9 then @ol_qty9
        when 10 then @ol_qty10
        when 11 then @ol_qty11
        when 12 then @ol_qty12
        when 13 then @ol_qty13
        when 14 then @ol_qty14
        when 15 then @ol_qty15
        end

-- get item data (no one updates item)
select  @i_price = i_price,
        @i_name = i_name,
        @i_data = i_data
from    item (tablock repeatableread)

```

```

        where    i_id = @li_id

-- update stock values
update  stock
set     s_ytd = s_ytd + @li_qty,
        @s_quantity = s_quantity - @li_qty +
        (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 *
                               @s_dist)

-- send line-item data to client
select  @i_name,
        @s_quantity,
        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,
        @o_entry_d  = getdate()
from    customer (repeatableread)
where   c_id        = @c_id and
        c_w_id      = @w_id and
        c_d_id      = @d_id

-- insert fresh row into orders table
insert into orders values ( @o_id,
                           @d_id,
                           @w_id,
                           @c_id_local,
                           @o_entry_d,
                           0,
                           @o_ol_cnt,
                           @o_all_local)

-- insert corresponding row into new-order table
insert into new_order values ( @o_id,
                              @d_id,
                              @w_id)

-- select warehouse tax
select  @w_tax      = w_tax
from    warehouse (repeatableread)
where   w_id       = @w_id

if (@commit_flag = 1)
        commit transaction n
else
-- all that work for nuthin!!!
        rollback transaction n

-- return order data to client
select  @w_tax,
        @d_tax,
        @o_id,
        @c_last,
        @c_discount,
        @c_credit,
        @o_entry_d,
        @commit_flag

```

```

end
go

```

ordstat.sql

```

-- File:      ORDSTAT.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.10
--           Copyright Microsoft, 1999
-- Purpose:   Creates order status transaction stored procedure

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

if exists ( select name from sysobjects where name = 'tpcc_orderstatus' )
        drop procedure  tpcc_orderstatus
go

create proc tpcc_orderstatus @w_id      int,
                           @d_id      tinyint,
                           @c_id      int,
                           @c_last    char(16) = ''

as

declare @c_balance      numeric(12,2),
        @c_first       char(16),
        @c_middle      char(2),
        @o_id          int,
        @o_entry_d     datetime,
        @o_carrier_id  smallint,
        @cnt           smallint

begin tran o

if (@c_id = 0)
        begin

-- get customer id and info using last name

        select  @cnt      = (count(*)+1)/2
        from    customer (repeatableread)
        where   c_last    = @c_last and
               c_w_id    = @w_id and
               c_d_id    = @d_id

        set     rowcount @cnt

        select  @c_id      = c_id,
               @c_balance = c_balance,
               @c_first   = c_first,
               @c_last    = c_last,
               @c_middle  = c_middle
        from    customer (repeatableread)
        where   c_last    = @c_last and

```

```

        c_w_id          = @w_id and
        c_d_id          = @d_id
    order by c_w_id, c_d_id, c_last, c_first

    set rowcount 0

end

else

begin

-- get customer info if by id

        select @c_balance = c_balance,
               @c_first = c_first,
               @c_middle = c_middle,
               @c_last = c_last
        from customer (repeatableread)
        where c_id = @c_id and
               c_d_id = @d_id and
               c_w_id = @w_id

        select @cnt = @@rowcount

    end

-- if no such customer

    if (@cnt = 0)
    begin
        raiserror('Customer not found',18,1)
        goto custnotfound
    end

-- get order info

    select @o_id = o_id,
           @o_entry_d = o_entry_d,
           @o_carrier_id = o_carrier_id
    from orders (serializable)
    where o_c_id = @c_id and
           o_d_id = @d_id and
           o_w_id = @w_id

    order by o_id asc

-- select order lines for the current order

    select ol_supply_w_id,
           ol_i_id,
           ol_quantity,
           ol_amount,
           ol_delivery_d
    from order_line (repeatableread)
    where ol_o_id = @o_id and
           ol_d_id = @d_id and
           ol_w_id = @w_id

custnotfound:

commit tran o

-- return data to client

```

```

select @c_id,
       @c_last,
       @c_first,
       @c_middle,
       @o_entry_d,
       @o_carrier_id,
       @c_balance,
       @o_id

go

```

payment.sql

```

-- File:      PAYMENT.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.10
--           Copyright Microsoft, 1999
-- Purpose:   Creates payment transaction stored procedure

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

if exists (select name from sysobjects where name = 'tpcc_payment' )
    drop procedure tpcc_payment
go

create proc tpcc_payment @w_id int,
                        @c_w_id int,
                        @h_amount numeric(6,2),
                        @d_id tinyint,
                        @c_d_id tinyint,
                        @c_id int,
                        @c_last char(16) = ''

as

declare @w_street_1 char(20),
        @w_street_2 char(20),
        @w_city char(20),
        @w_state char(2),
        @w_zip char(9),
        @w_name char(10),
        @d_street_1 char(20),
        @d_street_2 char(20),
        @d_city char(20),
        @d_state char(2),
        @d_zip char(9),
        @d_name char(10),
        @c_first char(16),
        @c_middle char(2),
        @c_street_1 char(20),
        @c_street_2 char(20),
        @c_city char(20),
        @c_state char(2),
        @c_zip char(9),
        @c_phone char(16),
        @c_since datetime,

```

```

@c_credit      char(2),
@c_credit_lim  numeric(12,2),
@c_balance     numeric(12,2),
@c_discount    numeric(4,4),
@data         char(500),
@c_data       char(500),
@datetime     datetime,
@w_ytd        numeric(12,2),
@d_ytd        numeric(12,2),
@cnt          smallint,
@val          smallint,
@screen_data  char(200),
@d_id_local   tinyint,
@w_id_local   int,
@c_id_local   int

select @screen_data = ''

begin tran p

-- get payment date

select @datetime = getdate()

if (@c_id = 0)
begin

-- get customer id and info using last name

select top 1 @c_id = c_id
from (
select top 50 percent c_id, c_first
from customer (repeatableread)
where c_last = @c_last and
c_w_id = @c_w_id and
c_d_id = @c_d_id
order by c_first
) top_fifty
order by c_first desc
end

-- get customer info and update balances

update customer
set @c_balance = c_balance - @h_amount,
c_payment_cnt = c_payment_cnt + 1,
c_ytd_payment = c_ytd_payment + @h_amount,
@c_first = c_first,
@c_middle = c_middle,
@c_last = c_last,
@c_street_1 = c_street_1,
@c_street_2 = c_street_2,
@c_city = c_city,
@c_state = c_state,
@c_zip = c_zip,
@c_phone = c_phone,
@c_credit = c_credit,
@c_credit_lim = c_credit_lim,
@c_discount = c_discount,
@c_since = c_since,
@data = c_data,
@c_id_local = c_id
where c_id = @c_id and

```

```

c_w_id = @c_w_id and
c_d_id = @c_d_id

-- if customer has bad credit get some more info

if (@c_credit = 'BC')
begin

-- compute new info

select @c_data = convert(char(5),@c_id) +
convert(char(4),@c_d_id) +
convert(char(5),@c_w_id) +
convert(char(4),@d_id) +
convert(char(5),@w_id) +
convert(char(19),@h_amount) +
substring(@data, 1, 458)

-- update customer info

update customer
set c_data = @c_data
where c_id = @c_id and
c_w_id = @c_w_id and
c_d_id = @c_d_id

select @screen_data = substring (@c_data,1,200)

end

-- get district data and update year-to-date

update district
set d_ytd = d_ytd + @h_amount,
@d_street_1 = d_street_1,
@d_street_2 = d_street_2,
@d_city = d_city,
@d_state = d_state,
@d_zip = d_zip,
@d_name = d_name,
@d_id_local = d_id
where d_w_id = @w_id and
d_id = @d_id

-- get warehouse data and update year-to-date

update warehouse
set w_ytd = w_ytd + @h_amount,
@w_street_1 = w_street_1,
@w_street_2 = w_street_2,
@w_city = w_city,
@w_state = w_state,
@w_zip = w_zip,
@w_name = w_name,
@w_id_local = w_id
where w_id = @w_id

-- create history record

insert into history values ( @c_id_local,
@c_d_id,
@c_w_id,
@d_id_local,
@w_id_local,

```

```

@datetime,
@h_amount,
@w_name + ' ' + @d_name)

commit tran p

-- return data to client

select  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data

go

```

stocklev.sql

```

-- File:      STOCKLEV.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.10
--           Copyright Microsoft, 1999
-- Purpose:   Creates stock level transaction stored procedure

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

if exists (select name from sysobjects where name = 'tpcc_stocklevel' )
drop procedure tpcc_stocklevel
go

create proc tpcc_stocklevel @w_id          int,
                           @d_id          tinyint,
                           @threshold    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   < @threshold

option (order group)

go

```

version.sql

```

-- File:      VERSION.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.10
--           Copyright Microsoft, 1999
-- Purpose:   Returns version level of TPC-C stored procs
-- Note:      Always update the return value of this proc for
--           any interface changes or 'must have' bug fixes.

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

if exists ( select name from sysobjects where name = 'tpcc_version' )
drop procedure tpcc_version
go

create proc tpcc_version
as
declare  @version char(8)

begin
        select @version = '4.10.000'
        select @version as 'Version'
end

go

```

tables.sql

NOTE: The following SQL script was run on each node in the configuration without modification.

```
-- File: TABLES.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.10
-- Copyright Microsoft, 1999
-- Purpose: Creates TPC-C tables

set ANSI_DEFAULTS on
SET ANSI_NULL_DFLT_ON OFF
set IMPLICIT_TRANSACTIONS off
set QUOTED_IDENTIFIER off
go

use tpcc
go

--
-- Remove all existing TPC-C tables, views, etc.
--

if objectproperty(object_id('warehouse'), 'IsView') is not null
    drop view warehouse
go
if objectproperty(object_id('district'), 'IsView') is not null
    drop view district
go
if objectproperty(object_id('customer'), 'IsView') is not null
    drop view customer
go
if objectproperty(object_id('history'), 'IsView') is not null
    drop view history
go
if objectproperty(object_id('stock'), 'IsView') is not null
    drop view stock
go
if objectproperty(object_id('orders'), 'IsView') is not null
    drop view orders
go
if objectproperty(object_id('order_line'), 'IsView') is not null
    drop view order_line
go
if objectproperty(object_id('new_order'), 'IsView') is not null
    drop view new_order
go
if objectproperty(object_id('item'), 'IsView') is not null
    drop view item
go

if objectproperty(object_id('t_warehouse'), 'IsTable') is not null
    drop table t_warehouse
go
if objectproperty(object_id('t_district'), 'IsTable') is not null
    drop table t_district
go
if objectproperty(object_id('t_customer'), 'IsTable') is not null
    drop table t_customer
go
if objectproperty(object_id('t_history'), 'IsTable') is not null
```

```
drop table t_history
go
if objectproperty(object_id('t_new_order'), 'IsTable') is not null
    drop table t_new_order
go
if objectproperty(object_id('t_orders'), 'IsTable') is not null
    drop table t_orders
go
if objectproperty(object_id('t_order_line'), 'IsTable') is not null
    drop table t_order_line
go
if objectproperty(object_id('t_item'), 'IsTable') is not null
    drop table t_item
go
if objectproperty(object_id('t_stock'), 'IsTable') is not null
    drop table t_stock
go

--
-- Create new tables
--

create table t_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 misc_fg
go

create table t_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 misc_fg
go

create table t_customer
(
    c_id int,
    c_d_id tinyint,
    c_w_id int,
    c_first char(16),
    c_middle char(2),
    c_last char(16),
    c_street_1 char(20),
    c_street_2 char(20),
```

```

        c_city                char(20),
        c_state               char(2),
        c_zip                 char(9),
        c_phone               char(16),
        c_since               datetime,
        c_credit              char(2),
        c_credit_lim          numeric(12,2),
        c_discount            numeric(4,4),
        c_balance             numeric(12,2),
        c_ytd_payment         numeric(12,2),
        c_payment_cnt         smallint,
        c_delivery_cnt        smallint,
        c_data                char(500)
) on big_fg
go

create table t_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),
    constraint history_clPK primary key clustered (h_w_id, h_d_id, h_c_id,
h_date)
) on misc_fg
go

create table t_new_order
(
    no_o_id                  int,
    no_d_id                  tinyint,
    no_w_id                  int
) on misc_fg
go

create table t_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 misc_fg
go

create table t_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 misc_fg
go

create table t_item
(
    i_id                    int,
    i_im_id                 int,
    i_name                  char(24),
    i_price                 numeric(5,2),
    i_data                  char(50)
) on misc_fg
go

create table t_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 big_fg
go

```

add_remote_servers.sql

```

--file add_remote_servers.sql
--add remote servers for all the partitions
--This script is run on every node
--
-- partition 1 is on host JAN
--

exec sp_dropserver rmt1, droplogins
exec sp_addlinkedserver 'rmt1'
exec sp_setnetname 'rmt1', 'JAN'
exec sp_serveroption 'rmt1', 'lazy schema validation', 'true'
exec sp_droplinkedsrvlogin rmt1, NULL
exec sp_addlinkedsrvlogin rmt1, 'false', 'sa', 'sa'

--
-- partition 2 is on host FEB
--

exec sp_dropserver rmt2, droplogins
exec sp_addlinkedserver 'rmt2'
exec sp_setnetname 'rmt2', 'FEB'
exec sp_serveroption 'rmt2', 'lazy schema validation', 'true'
exec sp_droplinkedsrvlogin rmt2, NULL

```

```

exec sp_addlinkedserver rmt2, 'false', 'sa', 'sa'

--
-- partition 3 is on host MAR
--

exec sp_dropserver rmt3, droplogins
exec sp_addlinkedserver 'rmt3'
exec sp_setnetname 'rmt3', 'MAR'
exec sp_serveroption 'rmt3', 'lazy schema validation', 'true'
exec sp_droplinkedserver rmt3, NULL
exec sp_addlinkedserver rmt3, 'false', 'sa', 'sa'

--
-- partition 4 is on host APR
--

exec sp_dropserver rmt4, droplogins
exec sp_addlinkedserver 'rmt4'
exec sp_setnetname 'rmt4', 'APR'
exec sp_serveroption 'rmt4', 'lazy schema validation', 'true'
exec sp_droplinkedserver rmt4, NULL
exec sp_addlinkedserver rmt4, 'false', 'sa', 'sa'

--
-- partition 5 is on host MAY
--

exec sp_dropserver rmt5, droplogins
exec sp_addlinkedserver 'rmt5'
exec sp_setnetname 'rmt5', 'MAY'
exec sp_serveroption 'rmt5', 'lazy schema validation', 'true'
exec sp_droplinkedserver rmt5, NULL
exec sp_addlinkedserver rmt5, 'false', 'sa', 'sa'

--
-- partition 6 is on host JUN
--

exec sp_dropserver rmt6, droplogins
exec sp_addlinkedserver 'rmt6'
exec sp_setnetname 'rmt6', 'JUN'
exec sp_serveroption 'rmt6', 'lazy schema validation', 'true'
exec sp_droplinkedserver rmt6, NULL
exec sp_addlinkedserver rmt6, 'false', 'sa', 'sa'

--
-- partition 7 is on host JUL
--

exec sp_dropserver rmt7, droplogins
exec sp_addlinkedserver 'rmt7'
exec sp_setnetname 'rmt7', 'JUL'
exec sp_serveroption 'rmt7', 'lazy schema validation', 'true'
exec sp_droplinkedserver rmt7, NULL
exec sp_addlinkedserver rmt7, 'false', 'sa', 'sa'

--
-- partition 8 is on host AUG
--

exec sp_dropserver rmt8, droplogins
exec sp_addlinkedserver 'rmt8'

```

```

exec sp_setnetname 'rmt8', 'AUG'
exec sp_serveroption 'rmt8', 'lazy schema validation', 'true'
exec sp_droplinkedserver rmt8, NULL
exec sp_addlinkedserver rmt8, 'false', 'sa', 'sa'

--
-- partition 9 is on host SEP
--

exec sp_dropserver rmt9, droplogins
exec sp_addlinkedserver 'rmt9'
exec sp_setnetname 'rmt9', 'SEP'
exec sp_serveroption 'rmt9', 'lazy schema validation', 'true'
exec sp_droplinkedserver rmt9, NULL
exec sp_addlinkedserver rmt9, 'false', 'sa', 'sa'

--
-- partition 10 is on host OCT
--

exec sp_dropserver rmt10, droplogins
exec sp_addlinkedserver 'rmt10'
exec sp_setnetname 'rmt10', 'OCT'
exec sp_serveroption 'rmt10', 'lazy schema validation', 'true'
exec sp_droplinkedserver rmt10, NULL
exec sp_addlinkedserver rmt10, 'false', 'sa', 'sa'

--
-- partition 11 is on host NOV
--

exec sp_dropserver rmt11, droplogins
exec sp_addlinkedserver 'rmt11'
exec sp_setnetname 'rmt11', 'NOV'
exec sp_serveroption 'rmt11', 'lazy schema validation', 'true'
exec sp_droplinkedserver rmt11, NULL
exec sp_addlinkedserver rmt11, 'false', 'sa', 'sa'

--
-- partition 12 is on host DEC
--

exec sp_dropserver rmt12, droplogins
exec sp_addlinkedserver 'rmt12'
exec sp_setnetname 'rmt12', 'DEC'
exec sp_serveroption 'rmt12', 'lazy schema validation', 'true'
exec sp_droplinkedserver rmt12, NULL
exec sp_addlinkedserver rmt12, 'false', 'sa', 'sa'

```

add_views_1.sql

Note: This script is only run on node 1.

```

-- file 1_to_1800

set ansi_warnings on
set ansi_nulls on
go

```

```

use tpcc
go

-- drop any existing views
if object_id('warehouse') is not null drop view warehouse
if object_id('district') is not null drop view district
if object_id('customer') is not null drop view customer
if object_id('history') is not null drop view history
if object_id('stock') is not null drop view stock
if object_id('orders') is not null drop view orders
if object_id('order_line') is not null drop view order_line
if object_id('new_order') is not null drop view new_order
if object_id('item') is not null drop view item
go

--add views for partition 1
create view warehouse as
select * from /*rmt1.tpcc.dbo.t_warehouse
union all
select * from rmt2.tpcc.dbo.t_warehouse
union all
select * from rmt3.tpcc.dbo.t_warehouse
union all
select * from rmt4.tpcc.dbo.t_warehouse
union all
select * from rmt5.tpcc.dbo.t_warehouse
union all
select * from rmt6.tpcc.dbo.t_warehouse
union all
select * from rmt7.tpcc.dbo.t_warehouse
union all
select * from rmt8.tpcc.dbo.t_warehouse
union all
select * from rmt9.tpcc.dbo.t_warehouse
union all
select * from rmt10.tpcc.dbo.t_warehouse
union all
select * from rmt11.tpcc.dbo.t_warehouse
union all
select * from rmt12.tpcc.dbo.t_warehouse
go

create view district as
select * from /*rmt1.tpcc.dbo.t_district
union all
select * from rmt2.tpcc.dbo.t_district
union all
select * from rmt3.tpcc.dbo.t_district
union all
select * from rmt4.tpcc.dbo.t_district
union all
select * from rmt5.tpcc.dbo.t_district
union all
select * from rmt6.tpcc.dbo.t_district
union all
select * from rmt7.tpcc.dbo.t_district
union all
select * from rmt8.tpcc.dbo.t_district
union all
select * from rmt9.tpcc.dbo.t_district
union all
select * from rmt10.tpcc.dbo.t_district

```

```

union all
select * from rmt11.tpcc.dbo.t_district
union all
select * from rmt12.tpcc.dbo.t_district
go

create view customer as
select * from /*rmt1.tpcc.dbo.t_customer
union all
select * from rmt2.tpcc.dbo.t_customer
union all
select * from rmt3.tpcc.dbo.t_customer
union all
select * from rmt4.tpcc.dbo.t_customer
union all
select * from rmt5.tpcc.dbo.t_customer
union all
select * from rmt6.tpcc.dbo.t_customer
union all
select * from rmt7.tpcc.dbo.t_customer
union all
select * from rmt8.tpcc.dbo.t_customer
union all
select * from rmt9.tpcc.dbo.t_customer
union all
select * from rmt10.tpcc.dbo.t_customer
union all
select * from rmt11.tpcc.dbo.t_customer
union all
select * from rmt12.tpcc.dbo.t_customer
go

create view history as
select * from /*rmt1.tpcc.dbo.t_history
union all
select * from rmt2.tpcc.dbo.t_history
union all
select * from rmt3.tpcc.dbo.t_history
union all
select * from rmt4.tpcc.dbo.t_history
union all
select * from rmt5.tpcc.dbo.t_history
union all
select * from rmt6.tpcc.dbo.t_history
union all
select * from rmt7.tpcc.dbo.t_history
union all
select * from rmt8.tpcc.dbo.t_history
union all
select * from rmt9.tpcc.dbo.t_history
union all
select * from rmt10.tpcc.dbo.t_history
union all
select * from rmt11.tpcc.dbo.t_history
union all
select * from rmt12.tpcc.dbo.t_history
go

create view stock as
select * from /*rmt1.tpcc.dbo.t_stock
union all
select * from rmt2.tpcc.dbo.t_stock
union all

```



```

select * from rmt3.tpcc.dbo.t_stock
union all
select * from rmt4.tpcc.dbo.t_stock
union all
select * from rmt5.tpcc.dbo.t_stock
union all
select * from rmt6.tpcc.dbo.t_stock
union all
select * from rmt7.tpcc.dbo.t_stock
union all
select * from rmt8.tpcc.dbo.t_stock
union all
select * from rmt9.tpcc.dbo.t_stock
union all
select * from rmt10.tpcc.dbo.t_stock
union all
select * from rmt11.tpcc.dbo.t_stock
union all
select * from rmt12.tpcc.dbo.t_stock
go

create view orders as
select * from /*rmt1.tpcc.dbo.*/t_orders
union all
select * from rmt2.tpcc.dbo.t_orders
union all
select * from rmt3.tpcc.dbo.t_orders
union all
select * from rmt4.tpcc.dbo.t_orders
union all
select * from rmt5.tpcc.dbo.t_orders
union all
select * from rmt6.tpcc.dbo.t_orders
union all
select * from rmt7.tpcc.dbo.t_orders
union all
select * from rmt8.tpcc.dbo.t_orders
union all
select * from rmt9.tpcc.dbo.t_orders
union all
select * from rmt10.tpcc.dbo.t_orders
union all
select * from rmt11.tpcc.dbo.t_orders
union all
select * from rmt12.tpcc.dbo.t_orders
go

create view order_line as
select * from /*rmt1.tpcc.dbo.*/t_order_line
union all
select * from rmt2.tpcc.dbo.t_order_line
union all
select * from rmt3.tpcc.dbo.t_order_line
union all
select * from rmt4.tpcc.dbo.t_order_line
union all
select * from rmt5.tpcc.dbo.t_order_line
union all
select * from rmt6.tpcc.dbo.t_order_line
union all
select * from rmt7.tpcc.dbo.t_order_line
union all
select * from rmt8.tpcc.dbo.t_order_line

```

```

union all
select * from rmt9.tpcc.dbo.t_order_line
union all
select * from rmt10.tpcc.dbo.t_order_line
union all
select * from rmt11.tpcc.dbo.t_order_line
union all
select * from rmt12.tpcc.dbo.t_order_line
go

create view new_order as
select * from /*rmt1.tpcc.dbo.*/t_new_order
union all
select * from rmt2.tpcc.dbo.t_new_order
union all
select * from rmt3.tpcc.dbo.t_new_order
union all
select * from rmt4.tpcc.dbo.t_new_order
union all
select * from rmt5.tpcc.dbo.t_new_order
union all
select * from rmt6.tpcc.dbo.t_new_order
union all
select * from rmt7.tpcc.dbo.t_new_order
union all
select * from rmt8.tpcc.dbo.t_new_order
union all
select * from rmt9.tpcc.dbo.t_new_order
union all
select * from rmt10.tpcc.dbo.t_new_order
union all
select * from rmt11.tpcc.dbo.t_new_order
union all
select * from rmt12.tpcc.dbo.t_new_order
go

create view item as
select * from t_item
go

```

add_views_2.sql

Note: This script is only run on node 2.

```

-- file 1801_to_3600\add_views.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- drop any existing views
if object_id('warehouse') is not null drop view warehouse
if object_id('district') is not null drop view district
if object_id('customer') is not null drop view customer
if object_id('history') is not null drop view history
if object_id('stock') is not null drop view stock

```

```

if object_id('orders') is not null drop view orders
if object_id('order_line') is not null drop view order_line
if object_id('new_order') is not null drop view new_order
if object_id('item') is not null drop view item
go

```

```

--add views for partition 2
create view warehouse as
select * from /*rmt2.tpcc.dbo.*/t_warehouse
union all
select * from rmt1.tpcc.dbo.t_warehouse
union all
select * from rmt3.tpcc.dbo.t_warehouse
union all
select * from rmt4.tpcc.dbo.t_warehouse
union all
select * from rmt5.tpcc.dbo.t_warehouse
union all
select * from rmt6.tpcc.dbo.t_warehouse
union all
select * from rmt7.tpcc.dbo.t_warehouse
union all
select * from rmt8.tpcc.dbo.t_warehouse
union all
select * from rmt9.tpcc.dbo.t_warehouse
union all
select * from rmt10.tpcc.dbo.t_warehouse
union all
select * from rmt11.tpcc.dbo.t_warehouse
union all
select * from rmt12.tpcc.dbo.t_warehouse
go

```

```

create view district as
select * from /*rmt2.tpcc.dbo.*/t_district
union all
select * from rmt1.tpcc.dbo.t_district
union all
select * from rmt3.tpcc.dbo.t_district
union all
select * from rmt4.tpcc.dbo.t_district
union all
select * from rmt5.tpcc.dbo.t_district
union all
select * from rmt6.tpcc.dbo.t_district
union all
select * from rmt7.tpcc.dbo.t_district
union all
select * from rmt8.tpcc.dbo.t_district
union all
select * from rmt9.tpcc.dbo.t_district
union all
select * from rmt10.tpcc.dbo.t_district
union all
select * from rmt11.tpcc.dbo.t_district
union all
select * from rmt12.tpcc.dbo.t_district
go

```

```

create view customer as
select * from /*rmt2.tpcc.dbo.*/t_customer
union all
select * from rmt1.tpcc.dbo.t_customer

```

```

union all
select * from rmt3.tpcc.dbo.t_customer
union all
select * from rmt4.tpcc.dbo.t_customer
union all
select * from rmt5.tpcc.dbo.t_customer
union all
select * from rmt6.tpcc.dbo.t_customer
union all
select * from rmt7.tpcc.dbo.t_customer
union all
select * from rmt8.tpcc.dbo.t_customer
union all
select * from rmt9.tpcc.dbo.t_customer
union all
select * from rmt10.tpcc.dbo.t_customer
union all
select * from rmt11.tpcc.dbo.t_customer
union all
select * from rmt12.tpcc.dbo.t_customer
go

```

```

create view history as
select * from /*rmt2.tpcc.dbo.*/t_history
union all
select * from rmt1.tpcc.dbo.t_history
union all
select * from rmt3.tpcc.dbo.t_history
union all
select * from rmt4.tpcc.dbo.t_history
union all
select * from rmt5.tpcc.dbo.t_history
union all
select * from rmt6.tpcc.dbo.t_history
union all
select * from rmt7.tpcc.dbo.t_history
union all
select * from rmt8.tpcc.dbo.t_history
union all
select * from rmt9.tpcc.dbo.t_history
union all
select * from rmt10.tpcc.dbo.t_history
union all
select * from rmt11.tpcc.dbo.t_history
union all
select * from rmt12.tpcc.dbo.t_history
go

```

```

create view stock as
select * from /*rmt2.tpcc.dbo.*/t_stock
union all
select * from rmt1.tpcc.dbo.t_stock
union all
select * from rmt3.tpcc.dbo.t_stock
union all
select * from rmt4.tpcc.dbo.t_stock
union all
select * from rmt5.tpcc.dbo.t_stock
union all
select * from rmt6.tpcc.dbo.t_stock
union all
select * from rmt7.tpcc.dbo.t_stock
union all

```

```

select * from rmt8.tpcc.dbo.t_stock
union all
select * from rmt9.tpcc.dbo.t_stock
union all
select * from rmt10.tpcc.dbo.t_stock
union all
select * from rmt11.tpcc.dbo.t_stock
union all
select * from rmt12.tpcc.dbo.t_stock
go

create view orders as
select * from /*rmt2.tpcc.dbo.*/t_orders
union all
select * from rmt1.tpcc.dbo.t_orders
union all
select * from rmt3.tpcc.dbo.t_orders
union all
select * from rmt4.tpcc.dbo.t_orders
union all
select * from rmt5.tpcc.dbo.t_orders
union all
select * from rmt6.tpcc.dbo.t_orders
union all
select * from rmt7.tpcc.dbo.t_orders
union all
select * from rmt8.tpcc.dbo.t_orders
union all
select * from rmt9.tpcc.dbo.t_orders
union all
select * from rmt10.tpcc.dbo.t_orders
union all
select * from rmt11.tpcc.dbo.t_orders
union all
select * from rmt12.tpcc.dbo.t_orders
go

create view order_line as
select * from /*rmt2.tpcc.dbo.*/t_order_line
union all
select * from rmt1.tpcc.dbo.t_order_line
union all
select * from rmt3.tpcc.dbo.t_order_line
union all
select * from rmt4.tpcc.dbo.t_order_line
union all
select * from rmt5.tpcc.dbo.t_order_line
union all
select * from rmt6.tpcc.dbo.t_order_line
union all
select * from rmt7.tpcc.dbo.t_order_line
union all
select * from rmt8.tpcc.dbo.t_order_line
union all
select * from rmt9.tpcc.dbo.t_order_line
union all
select * from rmt10.tpcc.dbo.t_order_line
union all
select * from rmt11.tpcc.dbo.t_order_line
union all
select * from rmt12.tpcc.dbo.t_order_line
go

```

```

create view new_order as
select * from /*rmt2.tpcc.dbo.*/t_new_order
union all
select * from rmt1.tpcc.dbo.t_new_order
union all
select * from rmt3.tpcc.dbo.t_new_order
union all
select * from rmt4.tpcc.dbo.t_new_order
union all
select * from rmt5.tpcc.dbo.t_new_order
union all
select * from rmt6.tpcc.dbo.t_new_order
union all
select * from rmt7.tpcc.dbo.t_new_order
union all
select * from rmt8.tpcc.dbo.t_new_order
union all
select * from rmt9.tpcc.dbo.t_new_order
union all
select * from rmt10.tpcc.dbo.t_new_order
union all
select * from rmt11.tpcc.dbo.t_new_order
union all
select * from rmt12.tpcc.dbo.t_new_order
go

create view item as
select * from t_item
go

```

add_views_3.sql

Note: This script is only run on node 3.

```

-- file 3601_to_5400\add_views.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- drop any existing views
if object_id('warehouse') is not null drop view warehouse
if object_id('district') is not null drop view district
if object_id('customer') is not null drop view customer
if object_id('history') is not null drop view history
if object_id('stock') is not null drop view stock
if object_id('orders') is not null drop view orders
if object_id('order_line') is not null drop view order_line
if object_id('new_order') is not null drop view new_order
if object_id('item') is not null drop view item
go

--add views for partition 3
create view warehouse as
select * from /*rmt3.tpcc.dbo.*/t_warehouse
union all

```

```

select * from rmt1.tpcc.dbo.t_warehouse
union all
select * from rmt2.tpcc.dbo.t_warehouse
union all
select * from rmt4.tpcc.dbo.t_warehouse
union all
select * from rmt5.tpcc.dbo.t_warehouse
union all
select * from rmt6.tpcc.dbo.t_warehouse
union all
select * from rmt7.tpcc.dbo.t_warehouse
union all
select * from rmt8.tpcc.dbo.t_warehouse
union all
select * from rmt9.tpcc.dbo.t_warehouse
union all
select * from rmt10.tpcc.dbo.t_warehouse
union all
select * from rmt11.tpcc.dbo.t_warehouse
union all
select * from rmt12.tpcc.dbo.t_warehouse
go

create view district as
select * from /*rmt3.tpcc.dbo.*/t_district
union all
select * from rmt1.tpcc.dbo.t_district
union all
select * from rmt2.tpcc.dbo.t_district
union all
select * from rmt4.tpcc.dbo.t_district
union all
select * from rmt5.tpcc.dbo.t_district
union all
select * from rmt6.tpcc.dbo.t_district
union all
select * from rmt7.tpcc.dbo.t_district
union all
select * from rmt8.tpcc.dbo.t_district
union all
select * from rmt9.tpcc.dbo.t_district
union all
select * from rmt10.tpcc.dbo.t_district
union all
select * from rmt11.tpcc.dbo.t_district
union all
select * from rmt12.tpcc.dbo.t_district
go

create view customer as
select * from /*rmt3.tpcc.dbo.*/t_customer
union all
select * from rmt1.tpcc.dbo.t_customer
union all
select * from rmt2.tpcc.dbo.t_customer
union all
select * from rmt4.tpcc.dbo.t_customer
union all
select * from rmt5.tpcc.dbo.t_customer
union all
select * from rmt6.tpcc.dbo.t_customer
union all
select * from rmt7.tpcc.dbo.t_customer

```

```

union all
select * from rmt8.tpcc.dbo.t_customer
union all
select * from rmt9.tpcc.dbo.t_customer
union all
select * from rmt10.tpcc.dbo.t_customer
union all
select * from rmt11.tpcc.dbo.t_customer
union all
select * from rmt12.tpcc.dbo.t_customer
go

create view history as
select * from /*rmt3.tpcc.dbo.*/t_history
union all
select * from rmt1.tpcc.dbo.t_history
union all
select * from rmt2.tpcc.dbo.t_history
union all
select * from rmt4.tpcc.dbo.t_history
union all
select * from rmt5.tpcc.dbo.t_history
union all
select * from rmt6.tpcc.dbo.t_history
union all
select * from rmt7.tpcc.dbo.t_history
union all
select * from rmt8.tpcc.dbo.t_history
union all
select * from rmt9.tpcc.dbo.t_history
union all
select * from rmt10.tpcc.dbo.t_history
union all
select * from rmt11.tpcc.dbo.t_history
union all
select * from rmt12.tpcc.dbo.t_history
go

create view stock as
select * from /*rmt3.tpcc.dbo.*/t_stock
union all
select * from rmt1.tpcc.dbo.t_stock
union all
select * from rmt2.tpcc.dbo.t_stock
union all
select * from rmt4.tpcc.dbo.t_stock
union all
select * from rmt5.tpcc.dbo.t_stock
union all
select * from rmt6.tpcc.dbo.t_stock
union all
select * from rmt7.tpcc.dbo.t_stock
union all
select * from rmt8.tpcc.dbo.t_stock
union all
select * from rmt9.tpcc.dbo.t_stock
union all
select * from rmt10.tpcc.dbo.t_stock
union all
select * from rmt11.tpcc.dbo.t_stock
union all
select * from rmt12.tpcc.dbo.t_stock
go

```

```

create view orders as
select * from /*rmt3.tpcc.dbo.*/t_orders
union all
select * from rmt1.tpcc.dbo.t_orders
union all
select * from rmt2.tpcc.dbo.t_orders
union all
select * from rmt4.tpcc.dbo.t_orders
union all
select * from rmt5.tpcc.dbo.t_orders
union all
select * from rmt6.tpcc.dbo.t_orders
union all
select * from rmt7.tpcc.dbo.t_orders
union all
select * from rmt8.tpcc.dbo.t_orders
union all
select * from rmt9.tpcc.dbo.t_orders
union all
select * from rmt10.tpcc.dbo.t_orders
union all
select * from rmt11.tpcc.dbo.t_orders
union all
select * from rmt12.tpcc.dbo.t_orders
go

create view order_line as
select * from /*rmt3.tpcc.dbo.*/t_order_line
union all
select * from rmt1.tpcc.dbo.t_order_line
union all
select * from rmt2.tpcc.dbo.t_order_line
union all
select * from rmt4.tpcc.dbo.t_order_line
union all
select * from rmt5.tpcc.dbo.t_order_line
union all
select * from rmt6.tpcc.dbo.t_order_line
union all
select * from rmt7.tpcc.dbo.t_order_line
union all
select * from rmt8.tpcc.dbo.t_order_line
union all
select * from rmt9.tpcc.dbo.t_order_line
union all
select * from rmt10.tpcc.dbo.t_order_line
union all
select * from rmt11.tpcc.dbo.t_order_line
union all
select * from rmt12.tpcc.dbo.t_order_line
go

create view new_order as
select * from /*rmt3.tpcc.dbo.*/t_new_order
union all
select * from rmt1.tpcc.dbo.t_new_order
union all
select * from rmt2.tpcc.dbo.t_new_order
union all
select * from rmt4.tpcc.dbo.t_new_order
union all
select * from rmt5.tpcc.dbo.t_new_order

```

```

union all
select * from rmt6.tpcc.dbo.t_new_order
union all
select * from rmt7.tpcc.dbo.t_new_order
union all
select * from rmt8.tpcc.dbo.t_new_order
union all
select * from rmt9.tpcc.dbo.t_new_order
union all
select * from rmt10.tpcc.dbo.t_new_order
union all
select * from rmt11.tpcc.dbo.t_new_order
union all
select * from rmt12.tpcc.dbo.t_new_order
go

create view item as
select * from t_item
go

```

add_views_4.sql

Note: This script is only run on node 4.

```

-- file 5401_to_7200\add_views.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- drop any existing views
if object_id('warehouse') is not null drop view warehouse
if object_id('district') is not null drop view district
if object_id('customer') is not null drop view customer
if object_id('history') is not null drop view history
if object_id('stock') is not null drop view stock
if object_id('orders') is not null drop view orders
if object_id('order_line') is not null drop view order_line
if object_id('new_order') is not null drop view new_order
if object_id('item') is not null drop view item
go

--add views for partition 4
create view warehouse as
select * from /*rmt4.tpcc.dbo.*/t_warehouse
union all
select * from rmt1.tpcc.dbo.t_warehouse
union all
select * from rmt2.tpcc.dbo.t_warehouse
union all
select * from rmt3.tpcc.dbo.t_warehouse
union all
select * from rmt5.tpcc.dbo.t_warehouse
union all
select * from rmt6.tpcc.dbo.t_warehouse
union all

```

```

select * from rmt7.tpcc.dbo.t_warehouse
union all
select * from rmt8.tpcc.dbo.t_warehouse
union all
select * from rmt9.tpcc.dbo.t_warehouse
union all
select * from rmt10.tpcc.dbo.t_warehouse
union all
select * from rmt11.tpcc.dbo.t_warehouse
union all
select * from rmt12.tpcc.dbo.t_warehouse
go

```

```

create view district as
select * from /*rmt4.tpcc.dbo.*/t_district
union all
select * from rmt1.tpcc.dbo.t_district
union all
select * from rmt2.tpcc.dbo.t_district
union all
select * from rmt3.tpcc.dbo.t_district
union all
select * from rmt5.tpcc.dbo.t_district
union all
select * from rmt6.tpcc.dbo.t_district
union all
select * from rmt7.tpcc.dbo.t_district
union all
select * from rmt8.tpcc.dbo.t_district
union all
select * from rmt9.tpcc.dbo.t_district
union all
select * from rmt10.tpcc.dbo.t_district
union all
select * from rmt11.tpcc.dbo.t_district
union all
select * from rmt12.tpcc.dbo.t_district
go

```

```

create view customer as
select * from /*rmt4.tpcc.dbo.*/t_customer
union all
select * from rmt1.tpcc.dbo.t_customer
union all
select * from rmt2.tpcc.dbo.t_customer
union all
select * from rmt3.tpcc.dbo.t_customer
union all
select * from rmt5.tpcc.dbo.t_customer
union all
select * from rmt6.tpcc.dbo.t_customer
union all
select * from rmt7.tpcc.dbo.t_customer
union all
select * from rmt8.tpcc.dbo.t_customer
union all
select * from rmt9.tpcc.dbo.t_customer
union all
select * from rmt10.tpcc.dbo.t_customer
union all
select * from rmt11.tpcc.dbo.t_customer
union all
select * from rmt12.tpcc.dbo.t_customer

```

go

```

create view history as
select * from /*rmt4.tpcc.dbo.*/t_history
union all
select * from rmt1.tpcc.dbo.t_history
union all
select * from rmt2.tpcc.dbo.t_history
union all
select * from rmt3.tpcc.dbo.t_history
union all
select * from rmt5.tpcc.dbo.t_history
union all
select * from rmt6.tpcc.dbo.t_history
union all
select * from rmt7.tpcc.dbo.t_history
union all
select * from rmt8.tpcc.dbo.t_history
union all
select * from rmt9.tpcc.dbo.t_history
union all
select * from rmt10.tpcc.dbo.t_history
union all
select * from rmt11.tpcc.dbo.t_history
union all
select * from rmt12.tpcc.dbo.t_history
go

```

```

create view stock as
select * from /*rmt4.tpcc.dbo.*/t_stock
union all
select * from rmt1.tpcc.dbo.t_stock
union all
select * from rmt2.tpcc.dbo.t_stock
union all
select * from rmt3.tpcc.dbo.t_stock
union all
select * from rmt5.tpcc.dbo.t_stock
union all
select * from rmt6.tpcc.dbo.t_stock
union all
select * from rmt7.tpcc.dbo.t_stock
union all
select * from rmt8.tpcc.dbo.t_stock
union all
select * from rmt9.tpcc.dbo.t_stock
union all
select * from rmt10.tpcc.dbo.t_stock
union all
select * from rmt11.tpcc.dbo.t_stock
union all
select * from rmt12.tpcc.dbo.t_stock
go

```

```

create view orders as
select * from /*rmt4.tpcc.dbo.*/t_orders
union all
select * from rmt1.tpcc.dbo.t_orders
union all
select * from rmt2.tpcc.dbo.t_orders
union all
select * from rmt3.tpcc.dbo.t_orders
union all

```

```

select * from rmt5.tpcc.dbo.t_orders
union all
select * from rmt6.tpcc.dbo.t_orders
union all
select * from rmt7.tpcc.dbo.t_orders
union all
select * from rmt8.tpcc.dbo.t_orders
union all
select * from rmt9.tpcc.dbo.t_orders
union all
select * from rmt10.tpcc.dbo.t_orders
union all
select * from rmt11.tpcc.dbo.t_orders
union all
select * from rmt12.tpcc.dbo.t_orders
go

create view order_line as
select * from /*rmt4.tpcc.dbo.*/t_order_line
union all
select * from rmt1.tpcc.dbo.t_order_line
union all
select * from rmt2.tpcc.dbo.t_order_line
union all
select * from rmt3.tpcc.dbo.t_order_line
union all
select * from rmt5.tpcc.dbo.t_order_line
union all
select * from rmt6.tpcc.dbo.t_order_line
union all
select * from rmt7.tpcc.dbo.t_order_line
union all
select * from rmt8.tpcc.dbo.t_order_line
union all
select * from rmt9.tpcc.dbo.t_order_line
union all
select * from rmt10.tpcc.dbo.t_order_line
union all
select * from rmt11.tpcc.dbo.t_order_line
union all
select * from rmt12.tpcc.dbo.t_order_line
go

create view new_order as
select * from /*rmt4.tpcc.dbo.*/t_new_order
union all
select * from rmt1.tpcc.dbo.t_new_order
union all
select * from rmt2.tpcc.dbo.t_new_order
union all
select * from rmt3.tpcc.dbo.t_new_order
union all
select * from rmt5.tpcc.dbo.t_new_order
union all
select * from rmt6.tpcc.dbo.t_new_order
union all
select * from rmt7.tpcc.dbo.t_new_order
union all
select * from rmt8.tpcc.dbo.t_new_order
union all
select * from rmt9.tpcc.dbo.t_new_order
union all
select * from rmt10.tpcc.dbo.t_new_order

```

```

union all
select * from rmt11.tpcc.dbo.t_new_order
union all
select * from rmt12.tpcc.dbo.t_new_order
go

create view item as
select * from t_item
go

```

add_views_5.sql

Note: This script is only run on node 5.

```

-- file 7201_to_9000\add_views.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- drop any existing views
if object_id('warehouse') is not null drop view warehouse
if object_id('district') is not null drop view district
if object_id('customer') is not null drop view customer
if object_id('history') is not null drop view history
if object_id('stock') is not null drop view stock
if object_id('orders') is not null drop view orders
if object_id('order_line') is not null drop view order_line
if object_id('new_order') is not null drop view new_order
if object_id('item') is not null drop view item
go

--add views for partition 5
create view warehouse as
select * from /*rmt5.tpcc.dbo.*/t_warehouse
union all
select * from rmt1.tpcc.dbo.t_warehouse
union all
select * from rmt2.tpcc.dbo.t_warehouse
union all
select * from rmt3.tpcc.dbo.t_warehouse
union all
select * from rmt4.tpcc.dbo.t_warehouse
union all
select * from rmt6.tpcc.dbo.t_warehouse
union all
select * from rmt7.tpcc.dbo.t_warehouse
union all
select * from rmt8.tpcc.dbo.t_warehouse
union all
select * from rmt9.tpcc.dbo.t_warehouse
union all
select * from rmt10.tpcc.dbo.t_warehouse
union all
select * from rmt11.tpcc.dbo.t_warehouse
union all

```

```

select * from rmt12.tpcc.dbo.t_warehouse
go

create view district as
select * from /*rmt5.tpcc.dbo.*/t_district
union all
select * from rmt1.tpcc.dbo.t_district
union all
select * from rmt2.tpcc.dbo.t_district
union all
select * from rmt3.tpcc.dbo.t_district
union all
select * from rmt4.tpcc.dbo.t_district
union all
select * from rmt6.tpcc.dbo.t_district
union all
select * from rmt7.tpcc.dbo.t_district
union all
select * from rmt8.tpcc.dbo.t_district
union all
select * from rmt9.tpcc.dbo.t_district
union all
select * from rmt10.tpcc.dbo.t_district
union all
select * from rmt11.tpcc.dbo.t_district
union all
select * from rmt12.tpcc.dbo.t_district
go

create view customer as
select * from /*rmt5.tpcc.dbo.*/t_customer
union all
select * from rmt1.tpcc.dbo.t_customer
union all
select * from rmt2.tpcc.dbo.t_customer
union all
select * from rmt3.tpcc.dbo.t_customer
union all
select * from rmt4.tpcc.dbo.t_customer
union all
select * from rmt6.tpcc.dbo.t_customer
union all
select * from rmt7.tpcc.dbo.t_customer
union all
select * from rmt8.tpcc.dbo.t_customer
union all
select * from rmt9.tpcc.dbo.t_customer
union all
select * from rmt10.tpcc.dbo.t_customer
union all
select * from rmt11.tpcc.dbo.t_customer
union all
select * from rmt12.tpcc.dbo.t_customer
go

create view history as
select * from /*rmt5.tpcc.dbo.*/t_history
union all
select * from rmt1.tpcc.dbo.t_history
union all
select * from rmt2.tpcc.dbo.t_history
union all
select * from rmt3.tpcc.dbo.t_history

```

```

union all
select * from rmt4.tpcc.dbo.t_history
union all
select * from rmt6.tpcc.dbo.t_history
union all
select * from rmt7.tpcc.dbo.t_history
union all
select * from rmt8.tpcc.dbo.t_history
union all
select * from rmt9.tpcc.dbo.t_history
union all
select * from rmt10.tpcc.dbo.t_history
union all
select * from rmt11.tpcc.dbo.t_history
union all
select * from rmt12.tpcc.dbo.t_history
go

create view stock as
select * from /*rmt5.tpcc.dbo.*/t_stock
union all
select * from rmt1.tpcc.dbo.t_stock
union all
select * from rmt2.tpcc.dbo.t_stock
union all
select * from rmt3.tpcc.dbo.t_stock
union all
select * from rmt4.tpcc.dbo.t_stock
union all
select * from rmt6.tpcc.dbo.t_stock
union all
select * from rmt7.tpcc.dbo.t_stock
union all
select * from rmt8.tpcc.dbo.t_stock
union all
select * from rmt9.tpcc.dbo.t_stock
union all
select * from rmt10.tpcc.dbo.t_stock
union all
select * from rmt11.tpcc.dbo.t_stock
union all
select * from rmt12.tpcc.dbo.t_stock
go

create view orders as
select * from /*rmt5.tpcc.dbo.*/t_orders
union all
select * from rmt1.tpcc.dbo.t_orders
union all
select * from rmt2.tpcc.dbo.t_orders
union all
select * from rmt3.tpcc.dbo.t_orders
union all
select * from rmt4.tpcc.dbo.t_orders
union all
select * from rmt6.tpcc.dbo.t_orders
union all
select * from rmt7.tpcc.dbo.t_orders
union all
select * from rmt8.tpcc.dbo.t_orders
union all
select * from rmt9.tpcc.dbo.t_orders
union all

```



```

select * from rmt10.tpcc.dbo.t_orders
union all
select * from rmt11.tpcc.dbo.t_orders
union all
select * from rmt12.tpcc.dbo.t_orders
go

create view order_line as
select * from /*rmt5.tpcc.dbo.*/t_order_line
union all
select * from rmt1.tpcc.dbo.t_order_line
union all
select * from rmt2.tpcc.dbo.t_order_line
union all
select * from rmt3.tpcc.dbo.t_order_line
union all
select * from rmt4.tpcc.dbo.t_order_line
union all
select * from rmt6.tpcc.dbo.t_order_line
union all
select * from rmt7.tpcc.dbo.t_order_line
union all
select * from rmt8.tpcc.dbo.t_order_line
union all
select * from rmt9.tpcc.dbo.t_order_line
union all
select * from rmt10.tpcc.dbo.t_order_line
union all
select * from rmt11.tpcc.dbo.t_order_line
union all
select * from rmt12.tpcc.dbo.t_order_line
go

create view new_order as
select * from /*rmt5.tpcc.dbo.*/t_new_order
union all
select * from rmt1.tpcc.dbo.t_new_order
union all
select * from rmt2.tpcc.dbo.t_new_order
union all
select * from rmt3.tpcc.dbo.t_new_order
union all
select * from rmt4.tpcc.dbo.t_new_order
union all
select * from rmt6.tpcc.dbo.t_new_order
union all
select * from rmt7.tpcc.dbo.t_new_order
union all
select * from rmt8.tpcc.dbo.t_new_order
union all
select * from rmt9.tpcc.dbo.t_new_order
union all
select * from rmt10.tpcc.dbo.t_new_order
union all
select * from rmt11.tpcc.dbo.t_new_order
union all
select * from rmt12.tpcc.dbo.t_new_order
go

create view item as
select * from t_item
go

```

add_views_6.sql

Note: This script is only run on node 6.

```

-- file 9001_to_10800\add_views.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- drop any existing views
if object_id('warehouse') is not null drop view warehouse
if object_id('district') is not null drop view district
if object_id('customer') is not null drop view customer
if object_id('history') is not null drop view history
if object_id('stock') is not null drop view stock
if object_id('orders') is not null drop view orders
if object_id('order_line') is not null drop view order_line
if object_id('new_order') is not null drop view new_order
if object_id('item') is not null drop view item
go

--add views for partition 6
create view warehouse as
select * from /*rmt6.tpcc.dbo.*/t_warehouse
union all
select * from rmt1.tpcc.dbo.t_warehouse
union all
select * from rmt2.tpcc.dbo.t_warehouse
union all
select * from rmt3.tpcc.dbo.t_warehouse
union all
select * from rmt4.tpcc.dbo.t_warehouse
union all
select * from rmt5.tpcc.dbo.t_warehouse
union all
select * from rmt7.tpcc.dbo.t_warehouse
union all
select * from rmt8.tpcc.dbo.t_warehouse
union all
select * from rmt9.tpcc.dbo.t_warehouse
union all
select * from rmt10.tpcc.dbo.t_warehouse
union all
select * from rmt11.tpcc.dbo.t_warehouse
union all
select * from rmt12.tpcc.dbo.t_warehouse
go

create view district as
select * from /*rmt6.tpcc.dbo.*/t_district
union all
select * from rmt1.tpcc.dbo.t_district
union all
select * from rmt2.tpcc.dbo.t_district
union all

```

```

select * from rmt3.tpcc.dbo.t_district
union all
select * from rmt4.tpcc.dbo.t_district
union all
select * from rmt5.tpcc.dbo.t_district
union all
select * from rmt7.tpcc.dbo.t_district
union all
select * from rmt8.tpcc.dbo.t_district
union all
select * from rmt9.tpcc.dbo.t_district
union all
select * from rmt10.tpcc.dbo.t_district
union all
select * from rmt11.tpcc.dbo.t_district
union all
select * from rmt12.tpcc.dbo.t_district
go

create view customer as
select * from /*rmt6.tpcc.dbo.*/t_customer
union all
select * from rmt1.tpcc.dbo.t_customer
union all
select * from rmt2.tpcc.dbo.t_customer
union all
select * from rmt3.tpcc.dbo.t_customer
union all
select * from rmt4.tpcc.dbo.t_customer
union all
select * from rmt5.tpcc.dbo.t_customer
union all
select * from rmt7.tpcc.dbo.t_customer
union all
select * from rmt8.tpcc.dbo.t_customer
union all
select * from rmt9.tpcc.dbo.t_customer
union all
select * from rmt10.tpcc.dbo.t_customer
union all
select * from rmt11.tpcc.dbo.t_customer
union all
select * from rmt12.tpcc.dbo.t_customer
go

create view history as
select * from /*rmt6.tpcc.dbo.*/t_history
union all
select * from rmt1.tpcc.dbo.t_history
union all
select * from rmt2.tpcc.dbo.t_history
union all
select * from rmt3.tpcc.dbo.t_history
union all
select * from rmt4.tpcc.dbo.t_history
union all
select * from rmt5.tpcc.dbo.t_history
union all
select * from rmt7.tpcc.dbo.t_history
union all
select * from rmt8.tpcc.dbo.t_history
union all
select * from rmt9.tpcc.dbo.t_history

```

```

union all
select * from rmt10.tpcc.dbo.t_history
union all
select * from rmt11.tpcc.dbo.t_history
union all
select * from rmt12.tpcc.dbo.t_history
go

create view stock as
select * from /*rmt6.tpcc.dbo.*/t_stock
union all
select * from rmt1.tpcc.dbo.t_stock
union all
select * from rmt2.tpcc.dbo.t_stock
union all
select * from rmt3.tpcc.dbo.t_stock
union all
select * from rmt4.tpcc.dbo.t_stock
union all
select * from rmt5.tpcc.dbo.t_stock
union all
select * from rmt7.tpcc.dbo.t_stock
union all
select * from rmt8.tpcc.dbo.t_stock
union all
select * from rmt9.tpcc.dbo.t_stock
union all
select * from rmt10.tpcc.dbo.t_stock
union all
select * from rmt11.tpcc.dbo.t_stock
union all
select * from rmt12.tpcc.dbo.t_stock
go

create view orders as
select * from /*rmt6.tpcc.dbo.*/t_orders
union all
select * from rmt1.tpcc.dbo.t_orders
union all
select * from rmt2.tpcc.dbo.t_orders
union all
select * from rmt3.tpcc.dbo.t_orders
union all
select * from rmt4.tpcc.dbo.t_orders
union all
select * from rmt5.tpcc.dbo.t_orders
union all
select * from rmt7.tpcc.dbo.t_orders
union all
select * from rmt8.tpcc.dbo.t_orders
union all
select * from rmt9.tpcc.dbo.t_orders
union all
select * from rmt10.tpcc.dbo.t_orders
union all
select * from rmt11.tpcc.dbo.t_orders
union all
select * from rmt12.tpcc.dbo.t_orders
go

create view order_line as
select * from /*rmt6.tpcc.dbo.*/t_order_line
union all

```

```

select * from rmt1.tpcc.dbo.t_order_line
union all
select * from rmt2.tpcc.dbo.t_order_line
union all
select * from rmt3.tpcc.dbo.t_order_line
union all
select * from rmt4.tpcc.dbo.t_order_line
union all
select * from rmt5.tpcc.dbo.t_order_line
union all
select * from rmt7.tpcc.dbo.t_order_line
union all
select * from rmt8.tpcc.dbo.t_order_line
union all
select * from rmt9.tpcc.dbo.t_order_line
union all
select * from rmt10.tpcc.dbo.t_order_line
union all
select * from rmt11.tpcc.dbo.t_order_line
union all
select * from rmt12.tpcc.dbo.t_order_line
go

create view new_order as
select * from /*rmt6.tpcc.dbo.*/t_new_order
union all
select * from rmt1.tpcc.dbo.t_new_order
union all
select * from rmt2.tpcc.dbo.t_new_order
union all
select * from rmt3.tpcc.dbo.t_new_order
union all
select * from rmt4.tpcc.dbo.t_new_order
union all
select * from rmt5.tpcc.dbo.t_new_order
union all
select * from rmt7.tpcc.dbo.t_new_order
union all
select * from rmt8.tpcc.dbo.t_new_order
union all
select * from rmt9.tpcc.dbo.t_new_order
union all
select * from rmt10.tpcc.dbo.t_new_order
union all
select * from rmt11.tpcc.dbo.t_new_order
union all
select * from rmt12.tpcc.dbo.t_new_order
go

create view item as
select * from t_item
go

```

add_views_7.sql

Note: This script is only run on node 7.

```
-- file 10801_to_12600\add_views.sql
```

```

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- drop any existing views
if object_id('warehouse') is not null drop view warehouse
if object_id('district') is not null drop view district
if object_id('customer') is not null drop view customer
if object_id('history') is not null drop view history
if object_id('stock') is not null drop view stock
if object_id('orders') is not null drop view orders
if object_id('order_line') is not null drop view order_line
if object_id('new_order') is not null drop view new_order
if object_id('item') is not null drop view item
go

--add views for partition 7
create view warehouse as
select * from /*rmt7.tpcc.dbo.*/t_warehouse
union all
select * from rmt1.tpcc.dbo.t_warehouse
union all
select * from rmt2.tpcc.dbo.t_warehouse
union all
select * from rmt3.tpcc.dbo.t_warehouse
union all
select * from rmt4.tpcc.dbo.t_warehouse
union all
select * from rmt5.tpcc.dbo.t_warehouse
union all
select * from rmt6.tpcc.dbo.t_warehouse
union all
select * from rmt8.tpcc.dbo.t_warehouse
union all
select * from rmt9.tpcc.dbo.t_warehouse
union all
select * from rmt10.tpcc.dbo.t_warehouse
union all
select * from rmt11.tpcc.dbo.t_warehouse
union all
select * from rmt12.tpcc.dbo.t_warehouse
go

create view district as
select * from /*rmt7.tpcc.dbo.*/t_district
union all
select * from rmt1.tpcc.dbo.t_district
union all
select * from rmt2.tpcc.dbo.t_district
union all
select * from rmt3.tpcc.dbo.t_district
union all
select * from rmt4.tpcc.dbo.t_district
union all
select * from rmt5.tpcc.dbo.t_district
union all
select * from rmt6.tpcc.dbo.t_district
union all
select * from rmt8.tpcc.dbo.t_district
union all

```

```

select * from rmt9.tpcc.dbo.t_district
union all
select * from rmt10.tpcc.dbo.t_district
union all
select * from rmt11.tpcc.dbo.t_district
union all
select * from rmt12.tpcc.dbo.t_district
go

create view customer as
select * from /*rmt7.tpcc.dbo.*/t_customer
union all
select * from rmt1.tpcc.dbo.t_customer
union all
select * from rmt2.tpcc.dbo.t_customer
union all
select * from rmt3.tpcc.dbo.t_customer
union all
select * from rmt4.tpcc.dbo.t_customer
union all
select * from rmt5.tpcc.dbo.t_customer
union all
select * from rmt6.tpcc.dbo.t_customer
union all
select * from rmt8.tpcc.dbo.t_customer
union all
select * from rmt9.tpcc.dbo.t_customer
union all
select * from rmt10.tpcc.dbo.t_customer
union all
select * from rmt11.tpcc.dbo.t_customer
union all
select * from rmt12.tpcc.dbo.t_customer
go

create view history as
select * from /*rmt7.tpcc.dbo.*/t_history
union all
select * from rmt1.tpcc.dbo.t_history
union all
select * from rmt2.tpcc.dbo.t_history
union all
select * from rmt3.tpcc.dbo.t_history
union all
select * from rmt4.tpcc.dbo.t_history
union all
select * from rmt5.tpcc.dbo.t_history
union all
select * from rmt6.tpcc.dbo.t_history
union all
select * from rmt8.tpcc.dbo.t_history
union all
select * from rmt9.tpcc.dbo.t_history
union all
select * from rmt10.tpcc.dbo.t_history
union all
select * from rmt11.tpcc.dbo.t_history
union all
select * from rmt12.tpcc.dbo.t_history
go

create view stock as
select * from /*rmt7.tpcc.dbo.*/t_stock

```

```

union all
select * from rmt1.tpcc.dbo.t_stock
union all
select * from rmt2.tpcc.dbo.t_stock
union all
select * from rmt3.tpcc.dbo.t_stock
union all
select * from rmt4.tpcc.dbo.t_stock
union all
select * from rmt5.tpcc.dbo.t_stock
union all
select * from rmt6.tpcc.dbo.t_stock
union all
select * from rmt8.tpcc.dbo.t_stock
union all
select * from rmt9.tpcc.dbo.t_stock
union all
select * from rmt10.tpcc.dbo.t_stock
union all
select * from rmt11.tpcc.dbo.t_stock
union all
select * from rmt12.tpcc.dbo.t_stock
go

create view orders as
select * from /*rmt7.tpcc.dbo.*/t_orders
union all
select * from rmt1.tpcc.dbo.t_orders
union all
select * from rmt2.tpcc.dbo.t_orders
union all
select * from rmt3.tpcc.dbo.t_orders
union all
select * from rmt4.tpcc.dbo.t_orders
union all
select * from rmt5.tpcc.dbo.t_orders
union all
select * from rmt6.tpcc.dbo.t_orders
union all
select * from rmt8.tpcc.dbo.t_orders
union all
select * from rmt9.tpcc.dbo.t_orders
union all
select * from rmt10.tpcc.dbo.t_orders
union all
select * from rmt11.tpcc.dbo.t_orders
union all
select * from rmt12.tpcc.dbo.t_orders
go

create view order_line as
select * from /*rmt7.tpcc.dbo.*/t_order_line
union all
select * from rmt1.tpcc.dbo.t_order_line
union all
select * from rmt2.tpcc.dbo.t_order_line
union all
select * from rmt3.tpcc.dbo.t_order_line
union all
select * from rmt4.tpcc.dbo.t_order_line
union all
select * from rmt5.tpcc.dbo.t_order_line
union all

```

```

select * from rmt6.tpcc.dbo.t_order_line
union all
select * from rmt8.tpcc.dbo.t_order_line
union all
select * from rmt9.tpcc.dbo.t_order_line
union all
select * from rmt10.tpcc.dbo.t_order_line
union all
select * from rmt11.tpcc.dbo.t_order_line
union all
select * from rmt12.tpcc.dbo.t_order_line
go

create view new_order as
select * from /*rmt7.tpcc.dbo.*/t_new_order
union all
select * from rmt1.tpcc.dbo.t_new_order
union all
select * from rmt2.tpcc.dbo.t_new_order
union all
select * from rmt3.tpcc.dbo.t_new_order
union all
select * from rmt4.tpcc.dbo.t_new_order
union all
select * from rmt5.tpcc.dbo.t_new_order
union all
select * from rmt6.tpcc.dbo.t_new_order
union all
select * from rmt8.tpcc.dbo.t_new_order
union all
select * from rmt9.tpcc.dbo.t_new_order
union all
select * from rmt10.tpcc.dbo.t_new_order
union all
select * from rmt11.tpcc.dbo.t_new_order
union all
select * from rmt12.tpcc.dbo.t_new_order
go

create view item as
select * from t_item
go

```

add_views_8.sql

Note: This script is only run on node 8.

```

-- file 12601_to_14400\add_views.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- drop any existing views
if object_id('warehouse') is not null drop view warehouse
if object_id('district') is not null drop view district

```

```

if object_id('customer') is not null drop view customer
if object_id('history') is not null drop view history
if object_id('stock') is not null drop view stock
if object_id('orders') is not null drop view orders
if object_id('order_line') is not null drop view order_line
if object_id('new_order') is not null drop view new_order
if object_id('item') is not null drop view item
go

```

```

--add views for partition 8
create view warehouse as
select * from /*rmt8.tpcc.dbo.*/t_warehouse
union all
select * from rmt1.tpcc.dbo.t_warehouse
union all
select * from rmt2.tpcc.dbo.t_warehouse
union all
select * from rmt3.tpcc.dbo.t_warehouse
union all
select * from rmt4.tpcc.dbo.t_warehouse
union all
select * from rmt5.tpcc.dbo.t_warehouse
union all
select * from rmt6.tpcc.dbo.t_warehouse
union all
select * from rmt7.tpcc.dbo.t_warehouse
union all
select * from rmt9.tpcc.dbo.t_warehouse
union all
select * from rmt10.tpcc.dbo.t_warehouse
union all
select * from rmt11.tpcc.dbo.t_warehouse
union all
select * from rmt12.tpcc.dbo.t_warehouse
go

```

```

create view district as
select * from /*rmt8.tpcc.dbo.*/t_district
union all
select * from rmt1.tpcc.dbo.t_district
union all
select * from rmt2.tpcc.dbo.t_district
union all
select * from rmt3.tpcc.dbo.t_district
union all
select * from rmt4.tpcc.dbo.t_district
union all
select * from rmt5.tpcc.dbo.t_district
union all
select * from rmt6.tpcc.dbo.t_district
union all
select * from rmt7.tpcc.dbo.t_district
union all
select * from rmt9.tpcc.dbo.t_district
union all
select * from rmt10.tpcc.dbo.t_district
union all
select * from rmt11.tpcc.dbo.t_district
union all
select * from rmt12.tpcc.dbo.t_district
go

```

```

create view customer as

```

```

select * from /*rmt8.tpcc.dbo.*/t_customer
union all
select * from rmt1.tpcc.dbo.t_customer
union all
select * from rmt2.tpcc.dbo.t_customer
union all
select * from rmt3.tpcc.dbo.t_customer
union all
select * from rmt4.tpcc.dbo.t_customer
union all
select * from rmt5.tpcc.dbo.t_customer
union all
select * from rmt6.tpcc.dbo.t_customer
union all
select * from rmt7.tpcc.dbo.t_customer
union all
select * from rmt9.tpcc.dbo.t_customer
union all
select * from rmt10.tpcc.dbo.t_customer
union all
select * from rmt11.tpcc.dbo.t_customer
union all
select * from rmt12.tpcc.dbo.t_customer
go

```

```

create view history as
select * from /*rmt8.tpcc.dbo.*/t_history
union all
select * from rmt1.tpcc.dbo.t_history
union all
select * from rmt2.tpcc.dbo.t_history
union all
select * from rmt3.tpcc.dbo.t_history
union all
select * from rmt4.tpcc.dbo.t_history
union all
select * from rmt5.tpcc.dbo.t_history
union all
select * from rmt6.tpcc.dbo.t_history
union all
select * from rmt7.tpcc.dbo.t_history
union all
select * from rmt9.tpcc.dbo.t_history
union all
select * from rmt10.tpcc.dbo.t_history
union all
select * from rmt11.tpcc.dbo.t_history
union all
select * from rmt12.tpcc.dbo.t_history
go

```

```

create view stock as
select * from /*rmt8.tpcc.dbo.*/t_stock
union all
select * from rmt1.tpcc.dbo.t_stock
union all
select * from rmt2.tpcc.dbo.t_stock
union all
select * from rmt3.tpcc.dbo.t_stock
union all
select * from rmt4.tpcc.dbo.t_stock
union all
select * from rmt5.tpcc.dbo.t_stock

```

```

union all
select * from rmt6.tpcc.dbo.t_stock
union all
select * from rmt7.tpcc.dbo.t_stock
union all
select * from rmt9.tpcc.dbo.t_stock
union all
select * from rmt10.tpcc.dbo.t_stock
union all
select * from rmt11.tpcc.dbo.t_stock
union all
select * from rmt12.tpcc.dbo.t_stock
go

```

```

create view orders as
select * from /*rmt8.tpcc.dbo.*/t_orders
union all
select * from rmt1.tpcc.dbo.t_orders
union all
select * from rmt2.tpcc.dbo.t_orders
union all
select * from rmt3.tpcc.dbo.t_orders
union all
select * from rmt4.tpcc.dbo.t_orders
union all
select * from rmt5.tpcc.dbo.t_orders
union all
select * from rmt6.tpcc.dbo.t_orders
union all
select * from rmt7.tpcc.dbo.t_orders
union all
select * from rmt9.tpcc.dbo.t_orders
union all
select * from rmt10.tpcc.dbo.t_orders
union all
select * from rmt11.tpcc.dbo.t_orders
union all
select * from rmt12.tpcc.dbo.t_orders
go

```

```

create view order_line as
select * from /*rmt8.tpcc.dbo.*/t_order_line
union all
select * from rmt1.tpcc.dbo.t_order_line
union all
select * from rmt2.tpcc.dbo.t_order_line
union all
select * from rmt3.tpcc.dbo.t_order_line
union all
select * from rmt4.tpcc.dbo.t_order_line
union all
select * from rmt5.tpcc.dbo.t_order_line
union all
select * from rmt6.tpcc.dbo.t_order_line
union all
select * from rmt7.tpcc.dbo.t_order_line
union all
select * from rmt9.tpcc.dbo.t_order_line
union all
select * from rmt10.tpcc.dbo.t_order_line
union all
select * from rmt11.tpcc.dbo.t_order_line
union all

```

```

select * from rmt12.tpcc.dbo.t_order_line
go

create view new_order as
select * from /*rmt8.tpcc.dbo.*/t_new_order
union all
select * from rmt1.tpcc.dbo.t_new_order
union all
select * from rmt2.tpcc.dbo.t_new_order
union all
select * from rmt3.tpcc.dbo.t_new_order
union all
select * from rmt4.tpcc.dbo.t_new_order
union all
select * from rmt5.tpcc.dbo.t_new_order
union all
select * from rmt6.tpcc.dbo.t_new_order
union all
select * from rmt7.tpcc.dbo.t_new_order
union all
select * from rmt9.tpcc.dbo.t_new_order
union all
select * from rmt10.tpcc.dbo.t_new_order
union all
select * from rmt11.tpcc.dbo.t_new_order
union all
select * from rmt12.tpcc.dbo.t_new_order
go

create view item as
select * from t_item
go

```

add_views_9.sql

Note: This script is only run on node 9.

```

-- file 14401_to_16200\add_views.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- drop any existing views
if object_id('warehouse') is not null drop view warehouse
if object_id('district') is not null drop view district
if object_id('customer') is not null drop view customer
if object_id('history') is not null drop view history
if object_id('stock') is not null drop view stock
if object_id('orders') is not null drop view orders
if object_id('order_line') is not null drop view order_line
if object_id('new_order') is not null drop view new_order
if object_id('item') is not null drop view item
go

--add views for partition 9

```

```

create view warehouse as
select * from /*rmt9.tpcc.dbo.*/t_warehouse
union all
select * from rmt1.tpcc.dbo.t_warehouse
union all
select * from rmt2.tpcc.dbo.t_warehouse
union all
select * from rmt3.tpcc.dbo.t_warehouse
union all
select * from rmt4.tpcc.dbo.t_warehouse
union all
select * from rmt5.tpcc.dbo.t_warehouse
union all
select * from rmt6.tpcc.dbo.t_warehouse
union all
select * from rmt7.tpcc.dbo.t_warehouse
union all
select * from rmt8.tpcc.dbo.t_warehouse
union all
select * from rmt10.tpcc.dbo.t_warehouse
union all
select * from rmt11.tpcc.dbo.t_warehouse
union all
select * from rmt12.tpcc.dbo.t_warehouse
go

```

```

create view district as
select * from /*rmt9.tpcc.dbo.*/t_district
union all
select * from rmt1.tpcc.dbo.t_district
union all
select * from rmt2.tpcc.dbo.t_district
union all
select * from rmt3.tpcc.dbo.t_district
union all
select * from rmt4.tpcc.dbo.t_district
union all
select * from rmt5.tpcc.dbo.t_district
union all
select * from rmt6.tpcc.dbo.t_district
union all
select * from rmt7.tpcc.dbo.t_district
union all
select * from rmt8.tpcc.dbo.t_district
union all
select * from rmt10.tpcc.dbo.t_district
union all
select * from rmt11.tpcc.dbo.t_district
union all
select * from rmt12.tpcc.dbo.t_district
go

```

```

create view customer as
select * from /*rmt9.tpcc.dbo.*/t_customer
union all
select * from rmt1.tpcc.dbo.t_customer
union all
select * from rmt2.tpcc.dbo.t_customer
union all
select * from rmt3.tpcc.dbo.t_customer
union all
select * from rmt4.tpcc.dbo.t_customer
union all

```

```

select * from rmt5.tpcc.dbo.t_customer
union all
select * from rmt6.tpcc.dbo.t_customer
union all
select * from rmt7.tpcc.dbo.t_customer
union all
select * from rmt8.tpcc.dbo.t_customer
union all
select * from rmt10.tpcc.dbo.t_customer
union all
select * from rmt11.tpcc.dbo.t_customer
union all
select * from rmt12.tpcc.dbo.t_customer
go

create view history as
select * from /*rmt9.tpcc.dbo.*/t_history
union all
select * from rmt1.tpcc.dbo.t_history
union all
select * from rmt2.tpcc.dbo.t_history
union all
select * from rmt3.tpcc.dbo.t_history
union all
select * from rmt4.tpcc.dbo.t_history
union all
select * from rmt5.tpcc.dbo.t_history
union all
select * from rmt6.tpcc.dbo.t_history
union all
select * from rmt7.tpcc.dbo.t_history
union all
select * from rmt8.tpcc.dbo.t_history
union all
select * from rmt10.tpcc.dbo.t_history
union all
select * from rmt11.tpcc.dbo.t_history
union all
select * from rmt12.tpcc.dbo.t_history
go

create view stock as
select * from /*rmt9.tpcc.dbo.*/t_stock
union all
select * from rmt1.tpcc.dbo.t_stock
union all
select * from rmt2.tpcc.dbo.t_stock
union all
select * from rmt3.tpcc.dbo.t_stock
union all
select * from rmt4.tpcc.dbo.t_stock
union all
select * from rmt5.tpcc.dbo.t_stock
union all
select * from rmt6.tpcc.dbo.t_stock
union all
select * from rmt7.tpcc.dbo.t_stock
union all
select * from rmt8.tpcc.dbo.t_stock
union all
select * from rmt10.tpcc.dbo.t_stock
union all
select * from rmt11.tpcc.dbo.t_stock

```

```

union all
select * from rmt12.tpcc.dbo.t_stock
go

create view orders as
select * from /*rmt9.tpcc.dbo.*/t_orders
union all
select * from rmt1.tpcc.dbo.t_orders
union all
select * from rmt2.tpcc.dbo.t_orders
union all
select * from rmt3.tpcc.dbo.t_orders
union all
select * from rmt4.tpcc.dbo.t_orders
union all
select * from rmt5.tpcc.dbo.t_orders
union all
select * from rmt6.tpcc.dbo.t_orders
union all
select * from rmt7.tpcc.dbo.t_orders
union all
select * from rmt8.tpcc.dbo.t_orders
union all
select * from rmt10.tpcc.dbo.t_orders
union all
select * from rmt11.tpcc.dbo.t_orders
union all
select * from rmt12.tpcc.dbo.t_orders
go

create view order_line as
select * from /*rmt9.tpcc.dbo.*/t_order_line
union all
select * from rmt1.tpcc.dbo.t_order_line
union all
select * from rmt2.tpcc.dbo.t_order_line
union all
select * from rmt3.tpcc.dbo.t_order_line
union all
select * from rmt4.tpcc.dbo.t_order_line
union all
select * from rmt5.tpcc.dbo.t_order_line
union all
select * from rmt6.tpcc.dbo.t_order_line
union all
select * from rmt7.tpcc.dbo.t_order_line
union all
select * from rmt8.tpcc.dbo.t_order_line
union all
select * from rmt10.tpcc.dbo.t_order_line
union all
select * from rmt11.tpcc.dbo.t_order_line
union all
select * from rmt12.tpcc.dbo.t_order_line
go

create view new_order as
select * from /*rmt9.tpcc.dbo.*/t_new_order
union all
select * from rmt1.tpcc.dbo.t_new_order
union all
select * from rmt2.tpcc.dbo.t_new_order
union all

```



```

select * from rmt3.tpc.c.dbo.t_new_order
union all
select * from rmt4.tpc.c.dbo.t_new_order
union all
select * from rmt5.tpc.c.dbo.t_new_order
union all
select * from rmt6.tpc.c.dbo.t_new_order
union all
select * from rmt7.tpc.c.dbo.t_new_order
union all
select * from rmt8.tpc.c.dbo.t_new_order
union all
select * from rmt10.tpc.c.dbo.t_new_order
union all
select * from rmt11.tpc.c.dbo.t_new_order
union all
select * from rmt12.tpc.c.dbo.t_new_order
go

create view item as
select * from t_item
go

```

add_views_10.sql

Note: This script is only run on node 10.

```

-- file 16201_to_18000\add_views.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- drop any existing views
if object_id('warehouse') is not null drop view warehouse
if object_id('district') is not null drop view district
if object_id('customer') is not null drop view customer
if object_id('history') is not null drop view history
if object_id('stock') is not null drop view stock
if object_id('orders') is not null drop view orders
if object_id('order_line') is not null drop view order_line
if object_id('new_order') is not null drop view new_order
if object_id('item') is not null drop view item
go

--add views for partition 10
create view warehouse as
select * from /*rmt10.tpc.c.dbo.*/t_warehouse
union all
select * from rmt1.tpc.c.dbo.t_warehouse
union all
select * from rmt2.tpc.c.dbo.t_warehouse
union all
select * from rmt3.tpc.c.dbo.t_warehouse
union all
select * from rmt4.tpc.c.dbo.t_warehouse

```

```

union all
select * from rmt5.tpc.c.dbo.t_warehouse
union all
select * from rmt6.tpc.c.dbo.t_warehouse
union all
select * from rmt7.tpc.c.dbo.t_warehouse
union all
select * from rmt8.tpc.c.dbo.t_warehouse
union all
select * from rmt9.tpc.c.dbo.t_warehouse
union all
select * from rmt11.tpc.c.dbo.t_warehouse
union all
select * from rmt12.tpc.c.dbo.t_warehouse
go

create view district as
select * from /*rmt10.tpc.c.dbo.*/t_district
union all
select * from rmt1.tpc.c.dbo.t_district
union all
select * from rmt2.tpc.c.dbo.t_district
union all
select * from rmt3.tpc.c.dbo.t_district
union all
select * from rmt4.tpc.c.dbo.t_district
union all
select * from rmt5.tpc.c.dbo.t_district
union all
select * from rmt6.tpc.c.dbo.t_district
union all
select * from rmt7.tpc.c.dbo.t_district
union all
select * from rmt8.tpc.c.dbo.t_district
union all
select * from rmt9.tpc.c.dbo.t_district
union all
select * from rmt11.tpc.c.dbo.t_district
union all
select * from rmt12.tpc.c.dbo.t_district
go

create view customer as
select * from /*rmt10.tpc.c.dbo.*/t_customer
union all
select * from rmt1.tpc.c.dbo.t_customer
union all
select * from rmt2.tpc.c.dbo.t_customer
union all
select * from rmt3.tpc.c.dbo.t_customer
union all
select * from rmt4.tpc.c.dbo.t_customer
union all
select * from rmt5.tpc.c.dbo.t_customer
union all
select * from rmt6.tpc.c.dbo.t_customer
union all
select * from rmt7.tpc.c.dbo.t_customer
union all
select * from rmt8.tpc.c.dbo.t_customer
union all
select * from rmt9.tpc.c.dbo.t_customer
union all

```

```

select * from rmt11.tpcc.dbo.t_customer
union all
select * from rmt12.tpcc.dbo.t_customer
go

create view history as
select * from /*rmt10.tpcc.dbo.*/t_history
union all
select * from rmt1.tpcc.dbo.t_history
union all
select * from rmt2.tpcc.dbo.t_history
union all
select * from rmt3.tpcc.dbo.t_history
union all
select * from rmt4.tpcc.dbo.t_history
union all
select * from rmt5.tpcc.dbo.t_history
union all
select * from rmt6.tpcc.dbo.t_history
union all
select * from rmt7.tpcc.dbo.t_history
union all
select * from rmt8.tpcc.dbo.t_history
union all
select * from rmt9.tpcc.dbo.t_history
union all
select * from rmt11.tpcc.dbo.t_history
union all
select * from rmt12.tpcc.dbo.t_history
go

create view stock as
select * from /*rmt10.tpcc.dbo.*/t_stock
union all
select * from rmt1.tpcc.dbo.t_stock
union all
select * from rmt2.tpcc.dbo.t_stock
union all
select * from rmt3.tpcc.dbo.t_stock
union all
select * from rmt4.tpcc.dbo.t_stock
union all
select * from rmt5.tpcc.dbo.t_stock
union all
select * from rmt6.tpcc.dbo.t_stock
union all
select * from rmt7.tpcc.dbo.t_stock
union all
select * from rmt8.tpcc.dbo.t_stock
union all
select * from rmt9.tpcc.dbo.t_stock
union all
select * from rmt11.tpcc.dbo.t_stock
union all
select * from rmt12.tpcc.dbo.t_stock
go

create view orders as
select * from /*rmt10.tpcc.dbo.*/t_orders
union all
select * from rmt1.tpcc.dbo.t_orders
union all
select * from rmt2.tpcc.dbo.t_orders

```

```

union all
select * from rmt3.tpcc.dbo.t_orders
union all
select * from rmt4.tpcc.dbo.t_orders
union all
select * from rmt5.tpcc.dbo.t_orders
union all
select * from rmt6.tpcc.dbo.t_orders
union all
select * from rmt7.tpcc.dbo.t_orders
union all
select * from rmt8.tpcc.dbo.t_orders
union all
select * from rmt9.tpcc.dbo.t_orders
union all
select * from rmt11.tpcc.dbo.t_orders
union all
select * from rmt12.tpcc.dbo.t_orders
go

create view order_line as
select * from /*rmt10.tpcc.dbo.*/t_order_line
union all
select * from rmt1.tpcc.dbo.t_order_line
union all
select * from rmt2.tpcc.dbo.t_order_line
union all
select * from rmt3.tpcc.dbo.t_order_line
union all
select * from rmt4.tpcc.dbo.t_order_line
union all
select * from rmt5.tpcc.dbo.t_order_line
union all
select * from rmt6.tpcc.dbo.t_order_line
union all
select * from rmt7.tpcc.dbo.t_order_line
union all
select * from rmt8.tpcc.dbo.t_order_line
union all
select * from rmt9.tpcc.dbo.t_order_line
union all
select * from rmt11.tpcc.dbo.t_order_line
union all
select * from rmt12.tpcc.dbo.t_order_line
go

create view new_order as
select * from /*rmt10.tpcc.dbo.*/t_new_order
union all
select * from rmt1.tpcc.dbo.t_new_order
union all
select * from rmt2.tpcc.dbo.t_new_order
union all
select * from rmt3.tpcc.dbo.t_new_order
union all
select * from rmt4.tpcc.dbo.t_new_order
union all
select * from rmt5.tpcc.dbo.t_new_order
union all
select * from rmt6.tpcc.dbo.t_new_order
union all
select * from rmt7.tpcc.dbo.t_new_order
union all

```

```

select * from rmt8.tpcc.dbo.t_new_order
union all
select * from rmt9.tpcc.dbo.t_new_order
union all
select * from rmt11.tpcc.dbo.t_new_order
union all
select * from rmt12.tpcc.dbo.t_new_order
go

create view item as
select * from t_item
go

```

add_views_11.sql

Note: This script is only run on node 11.

```

-- file 18001_to_19800\add_views.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- drop any existing views
if object_id('warehouse') is not null drop view warehouse
if object_id('district') is not null drop view district
if object_id('customer') is not null drop view customer
if object_id('history') is not null drop view history
if object_id('stock') is not null drop view stock
if object_id('orders') is not null drop view orders
if object_id('order_line') is not null drop view order_line
if object_id('new_order') is not null drop view new_order
if object_id('item') is not null drop view item
go

--add views for partition 11
create view warehouse as
select * from /*rmt11.tpcc.dbo.*/t_warehouse
union all
select * from rmt1.tpcc.dbo.t_warehouse
union all
select * from rmt2.tpcc.dbo.t_warehouse
union all
select * from rmt3.tpcc.dbo.t_warehouse
union all
select * from rmt4.tpcc.dbo.t_warehouse
union all
select * from rmt5.tpcc.dbo.t_warehouse
union all
select * from rmt6.tpcc.dbo.t_warehouse
union all
select * from rmt7.tpcc.dbo.t_warehouse
union all
select * from rmt8.tpcc.dbo.t_warehouse
union all
select * from rmt9.tpcc.dbo.t_warehouse

```

```

union all
select * from rmt10.tpcc.dbo.t_warehouse
union all
select * from rmt12.tpcc.dbo.t_warehouse
go

create view district as
select * from /*rmt11.tpcc.dbo.*/t_district
union all
select * from rmt1.tpcc.dbo.t_district
union all
select * from rmt2.tpcc.dbo.t_district
union all
select * from rmt3.tpcc.dbo.t_district
union all
select * from rmt4.tpcc.dbo.t_district
union all
select * from rmt5.tpcc.dbo.t_district
union all
select * from rmt6.tpcc.dbo.t_district
union all
select * from rmt7.tpcc.dbo.t_district
union all
select * from rmt8.tpcc.dbo.t_district
union all
select * from rmt9.tpcc.dbo.t_district
union all
select * from rmt10.tpcc.dbo.t_district
union all
select * from rmt12.tpcc.dbo.t_district
go

create view customer as
select * from /*rmt11.tpcc.dbo.*/t_customer
union all
select * from rmt1.tpcc.dbo.t_customer
union all
select * from rmt2.tpcc.dbo.t_customer
union all
select * from rmt3.tpcc.dbo.t_customer
union all
select * from rmt4.tpcc.dbo.t_customer
union all
select * from rmt5.tpcc.dbo.t_customer
union all
select * from rmt6.tpcc.dbo.t_customer
union all
select * from rmt7.tpcc.dbo.t_customer
union all
select * from rmt8.tpcc.dbo.t_customer
union all
select * from rmt9.tpcc.dbo.t_customer
union all
select * from rmt10.tpcc.dbo.t_customer
union all
select * from rmt12.tpcc.dbo.t_customer
go

create view history as
select * from /*rmt11.tpcc.dbo.*/t_history
union all
select * from rmt1.tpcc.dbo.t_history
union all

```

```

select * from rmt2.tpcc.dbo.t_history
union all
select * from rmt3.tpcc.dbo.t_history
union all
select * from rmt4.tpcc.dbo.t_history
union all
select * from rmt5.tpcc.dbo.t_history
union all
select * from rmt6.tpcc.dbo.t_history
union all
select * from rmt7.tpcc.dbo.t_history
union all
select * from rmt8.tpcc.dbo.t_history
union all
select * from rmt9.tpcc.dbo.t_history
union all
select * from rmt10.tpcc.dbo.t_history
union all
select * from rmt12.tpcc.dbo.t_history
go

create view stock as
select * from /*rmt11.tpcc.dbo.*/t_stock
union all
select * from rmt1.tpcc.dbo.t_stock
union all
select * from rmt2.tpcc.dbo.t_stock
union all
select * from rmt3.tpcc.dbo.t_stock
union all
select * from rmt4.tpcc.dbo.t_stock
union all
select * from rmt5.tpcc.dbo.t_stock
union all
select * from rmt6.tpcc.dbo.t_stock
union all
select * from rmt7.tpcc.dbo.t_stock
union all
select * from rmt8.tpcc.dbo.t_stock
union all
select * from rmt9.tpcc.dbo.t_stock
union all
select * from rmt10.tpcc.dbo.t_stock
union all
select * from rmt12.tpcc.dbo.t_stock
go

create view orders as
select * from /*rmt11.tpcc.dbo.*/t_orders
union all
select * from rmt1.tpcc.dbo.t_orders
union all
select * from rmt2.tpcc.dbo.t_orders
union all
select * from rmt3.tpcc.dbo.t_orders
union all
select * from rmt4.tpcc.dbo.t_orders
union all
select * from rmt5.tpcc.dbo.t_orders
union all
select * from rmt6.tpcc.dbo.t_orders
union all
select * from rmt7.tpcc.dbo.t_orders

```

```

union all
select * from rmt8.tpcc.dbo.t_orders
union all
select * from rmt9.tpcc.dbo.t_orders
union all
select * from rmt10.tpcc.dbo.t_orders
union all
select * from rmt12.tpcc.dbo.t_orders
go

create view order_line as
select * from /*rmt11.tpcc.dbo.*/t_order_line
union all
select * from rmt1.tpcc.dbo.t_order_line
union all
select * from rmt2.tpcc.dbo.t_order_line
union all
select * from rmt3.tpcc.dbo.t_order_line
union all
select * from rmt4.tpcc.dbo.t_order_line
union all
select * from rmt5.tpcc.dbo.t_order_line
union all
select * from rmt6.tpcc.dbo.t_order_line
union all
select * from rmt7.tpcc.dbo.t_order_line
union all
select * from rmt8.tpcc.dbo.t_order_line
union all
select * from rmt9.tpcc.dbo.t_order_line
union all
select * from rmt10.tpcc.dbo.t_order_line
union all
select * from rmt12.tpcc.dbo.t_order_line
go

create view new_order as
select * from /*rmt11.tpcc.dbo.*/t_new_order
union all
select * from rmt1.tpcc.dbo.t_new_order
union all
select * from rmt2.tpcc.dbo.t_new_order
union all
select * from rmt3.tpcc.dbo.t_new_order
union all
select * from rmt4.tpcc.dbo.t_new_order
union all
select * from rmt5.tpcc.dbo.t_new_order
union all
select * from rmt6.tpcc.dbo.t_new_order
union all
select * from rmt7.tpcc.dbo.t_new_order
union all
select * from rmt8.tpcc.dbo.t_new_order
union all
select * from rmt9.tpcc.dbo.t_new_order
union all
select * from rmt10.tpcc.dbo.t_new_order
union all
select * from rmt12.tpcc.dbo.t_new_order
go

create view item as

```

```
select * from t_item
go
```

add_views_12.sql

Note: This script is only run on node 12.

```
-- file 19801_to_21600\add_views.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- drop any existing views
if object_id('warehouse') is not null drop view warehouse
if object_id('district') is not null drop view district
if object_id('customer') is not null drop view customer
if object_id('history') is not null drop view history
if object_id('stock') is not null drop view stock
if object_id('orders') is not null drop view orders
if object_id('order_line') is not null drop view order_line
if object_id('new_order') is not null drop view new_order
if object_id('item') is not null drop view item
go

--add views for partition 12
create view warehouse as
select * from /*rmt12.tpcc.dbo.*/t_warehouse
union all
select * from rmt1.tpcc.dbo.t_warehouse
union all
select * from rmt2.tpcc.dbo.t_warehouse
union all
select * from rmt3.tpcc.dbo.t_warehouse
union all
select * from rmt4.tpcc.dbo.t_warehouse
union all
select * from rmt5.tpcc.dbo.t_warehouse
union all
select * from rmt6.tpcc.dbo.t_warehouse
union all
select * from rmt7.tpcc.dbo.t_warehouse
union all
select * from rmt8.tpcc.dbo.t_warehouse
union all
select * from rmt9.tpcc.dbo.t_warehouse
union all
select * from rmt10.tpcc.dbo.t_warehouse
union all
select * from rmt11.tpcc.dbo.t_warehouse
go

create view district as
select * from /*rmt12.tpcc.dbo.*/t_district
union all
select * from rmt1.tpcc.dbo.t_district
```

```
union all
select * from rmt2.tpcc.dbo.t_district
union all
select * from rmt3.tpcc.dbo.t_district
union all
select * from rmt4.tpcc.dbo.t_district
union all
select * from rmt5.tpcc.dbo.t_district
union all
select * from rmt6.tpcc.dbo.t_district
union all
select * from rmt7.tpcc.dbo.t_district
union all
select * from rmt8.tpcc.dbo.t_district
union all
select * from rmt9.tpcc.dbo.t_district
union all
select * from rmt10.tpcc.dbo.t_district
union all
select * from rmt11.tpcc.dbo.t_district
go

create view customer as
select * from /*rmt12.tpcc.dbo.*/t_customer
union all
select * from rmt1.tpcc.dbo.t_customer
union all
select * from rmt2.tpcc.dbo.t_customer
union all
select * from rmt3.tpcc.dbo.t_customer
union all
select * from rmt4.tpcc.dbo.t_customer
union all
select * from rmt5.tpcc.dbo.t_customer
union all
select * from rmt6.tpcc.dbo.t_customer
union all
select * from rmt7.tpcc.dbo.t_customer
union all
select * from rmt8.tpcc.dbo.t_customer
union all
select * from rmt9.tpcc.dbo.t_customer
union all
select * from rmt10.tpcc.dbo.t_customer
union all
select * from rmt11.tpcc.dbo.t_customer
go

create view history as
select * from /*rmt12.tpcc.dbo.*/t_history
union all
select * from rmt1.tpcc.dbo.t_history
union all
select * from rmt2.tpcc.dbo.t_history
union all
select * from rmt3.tpcc.dbo.t_history
union all
select * from rmt4.tpcc.dbo.t_history
union all
select * from rmt5.tpcc.dbo.t_history
union all
select * from rmt6.tpcc.dbo.t_history
union all
```

```

select * from rmt7.tpcc.dbo.t_history
union all
select * from rmt8.tpcc.dbo.t_history
union all
select * from rmt9.tpcc.dbo.t_history
union all
select * from rmt10.tpcc.dbo.t_history
union all
select * from rmt11.tpcc.dbo.t_history
go

create view stock as
select * from /*rmt12.tpcc.dbo.*/t_stock
union all
select * from rmt1.tpcc.dbo.t_stock
union all
select * from rmt2.tpcc.dbo.t_stock
union all
select * from rmt3.tpcc.dbo.t_stock
union all
select * from rmt4.tpcc.dbo.t_stock
union all
select * from rmt5.tpcc.dbo.t_stock
union all
select * from rmt6.tpcc.dbo.t_stock
union all
select * from rmt7.tpcc.dbo.t_stock
union all
select * from rmt8.tpcc.dbo.t_stock
union all
select * from rmt9.tpcc.dbo.t_stock
union all
select * from rmt10.tpcc.dbo.t_stock
union all
select * from rmt11.tpcc.dbo.t_stock
go

create view orders as
select * from /*rmt12.tpcc.dbo.*/t_orders
union all
select * from rmt1.tpcc.dbo.t_orders
union all
select * from rmt2.tpcc.dbo.t_orders
union all
select * from rmt3.tpcc.dbo.t_orders
union all
select * from rmt4.tpcc.dbo.t_orders
union all
select * from rmt5.tpcc.dbo.t_orders
union all
select * from rmt6.tpcc.dbo.t_orders
union all
select * from rmt7.tpcc.dbo.t_orders
union all
select * from rmt8.tpcc.dbo.t_orders
union all
select * from rmt9.tpcc.dbo.t_orders
union all
select * from rmt10.tpcc.dbo.t_orders
union all
select * from rmt11.tpcc.dbo.t_orders
go

```

```

create view order_line as
select * from /*rmt12.tpcc.dbo.*/t_order_line
union all
select * from rmt1.tpcc.dbo.t_order_line
union all
select * from rmt2.tpcc.dbo.t_order_line
union all
select * from rmt3.tpcc.dbo.t_order_line
union all
select * from rmt4.tpcc.dbo.t_order_line
union all
select * from rmt5.tpcc.dbo.t_order_line
union all
select * from rmt6.tpcc.dbo.t_order_line
union all
select * from rmt7.tpcc.dbo.t_order_line
union all
select * from rmt8.tpcc.dbo.t_order_line
union all
select * from rmt9.tpcc.dbo.t_order_line
union all
select * from rmt10.tpcc.dbo.t_order_line
union all
select * from rmt11.tpcc.dbo.t_order_line
go

create view new_order as
select * from /*rmt12.tpcc.dbo.*/t_new_order
union all
select * from rmt1.tpcc.dbo.t_new_order
union all
select * from rmt2.tpcc.dbo.t_new_order
union all
select * from rmt3.tpcc.dbo.t_new_order
union all
select * from rmt4.tpcc.dbo.t_new_order
union all
select * from rmt5.tpcc.dbo.t_new_order
union all
select * from rmt6.tpcc.dbo.t_new_order
union all
select * from rmt7.tpcc.dbo.t_new_order
union all
select * from rmt8.tpcc.dbo.t_new_order
union all
select * from rmt9.tpcc.dbo.t_new_order
union all
select * from rmt10.tpcc.dbo.t_new_order
union all
select * from rmt11.tpcc.dbo.t_new_order
go

create view item as
select * from t_item
go

```

item_iot_1.sql

Note: This script is only run on node 1.

```

-- file 1_to_1800\item_iot.sql
--add item instead-of-trigger for partition 1

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

drop trigger iot_item
go

create trigger iot_item on item instead of update, insert, delete as
begin

--
-- local node (partition 1)
DELETE A FROM t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT t_item SELECT * FROM INSERTED

--
-- remote node, partition 2
DELETE A FROM rmt2.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt2.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 3
DELETE A FROM rmt3.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt3.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 4
DELETE A FROM rmt4.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt4.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 5
DELETE A FROM rmt5.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt5.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 6
DELETE A FROM rmt6.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt6.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 7
DELETE A FROM rmt7.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt7.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 8
DELETE A FROM rmt8.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt8.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 9
DELETE A FROM rmt9.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt9.tpcc.dbo.t_item SELECT * FROM INSERTED

```

```

--
-- remote node, partition 10
DELETE A FROM rmt10.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt10.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 11
DELETE A FROM rmt11.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt11.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 12
DELETE A FROM rmt12.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt12.tpcc.dbo.t_item SELECT * FROM INSERTED

end

```

item_iot_2.sql

Note: This script is only run on node 2.

```

-- file 1801_to_3600\item_iot.sql
--add item instead-of-trigger for partition 2

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

drop trigger iot_item
go

create trigger iot_item on item instead of update, insert, delete as
begin

--
-- remote node, partition 1
DELETE A FROM rmt1.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt1.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- local node (partition 2)
DELETE A FROM t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT t_item SELECT * FROM INSERTED

--
-- remote node, partition 3
DELETE A FROM rmt3.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt3.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 4
DELETE A FROM rmt4.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt4.tpcc.dbo.t_item SELECT * FROM INSERTED

```

```

--
-- remote node, partition 5
DELETE A FROM rmt5.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt5.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 6
DELETE A FROM rmt6.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt6.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 7
DELETE A FROM rmt7.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt7.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 8
DELETE A FROM rmt8.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt8.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 9
DELETE A FROM rmt9.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt9.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 10
DELETE A FROM rmt10.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt10.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 11
DELETE A FROM rmt11.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt11.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 12
DELETE A FROM rmt12.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt12.tpcc.dbo.t_item SELECT * FROM INSERTED

end

```

item_iot_3.sql

Note: This script is only run on node 3.

```

-- file 3601_to_5400\item_iot.sql
--add item instead-of-trigger for partition 3

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

drop trigger iot_item
go

```

```

create trigger iot_item on item instead of update, insert, delete as
begin

--
-- remote node, partition 1
DELETE A FROM rmt1.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt1.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 2
DELETE A FROM rmt2.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt2.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- local node (partition 3)
DELETE A FROM t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT t_item SELECT * FROM INSERTED

--
-- remote node, partition 4
DELETE A FROM rmt4.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt4.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 5
DELETE A FROM rmt5.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt5.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 6
DELETE A FROM rmt6.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt6.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 7
DELETE A FROM rmt7.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt7.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 8
DELETE A FROM rmt8.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt8.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 9
DELETE A FROM rmt9.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt9.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 10
DELETE A FROM rmt10.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt10.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 11
DELETE A FROM rmt11.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt11.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 12
DELETE A FROM rmt12.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt12.tpcc.dbo.t_item SELECT * FROM INSERTED

```



```
end
```

item_iot_4.sql

Note: This script is only run on node 4.

```
-- file 5401_to_7200\item_iot.sql

--add item instead-of-trigger for partition 4

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

drop trigger iot_item
go

create trigger iot_item on item instead of update, insert, delete as
begin

--
-- remote node, partition 1
DELETE A FROM rmt1.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt1.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 2
DELETE A FROM rmt2.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt2.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 3
DELETE A FROM rmt3.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt3.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- local node (partition 4)
DELETE A FROM t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT t_item SELECT * FROM INSERTED

--
-- remote node, partition 5
DELETE A FROM rmt5.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt5.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 6
DELETE A FROM rmt6.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt6.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 7
DELETE A FROM rmt7.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt7.tpcc.dbo.t_item SELECT * FROM INSERTED
```

```
--
-- remote node, partition 8
DELETE A FROM rmt8.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt8.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 9
DELETE A FROM rmt9.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt9.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 10
DELETE A FROM rmt10.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt10.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 11
DELETE A FROM rmt11.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt11.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 12
DELETE A FROM rmt12.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt12.tpcc.dbo.t_item SELECT * FROM INSERTED

end
```

item_iot_5.sql

Note: This script is only run on node 5.

```
-- file 7201_to_9000\item_iot.sql

--add item instead-of-trigger for partition 5

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

drop trigger iot_item
go

create trigger iot_item on item instead of update, insert, delete as
begin

--
-- remote node, partition 1
DELETE A FROM rmt1.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt1.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 2
DELETE A FROM rmt2.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt2.tpcc.dbo.t_item SELECT * FROM INSERTED

--
```

```

-- remote node, partition 3
DELETE A FROM rmt3.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt3.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 4
DELETE A FROM rmt4.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt4.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- local node (partition 5)
DELETE A FROM t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT t_item SELECT * FROM INSERTED

--
-- remote node, partition 6
DELETE A FROM rmt6.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt6.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 7
DELETE A FROM rmt7.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt7.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 8
DELETE A FROM rmt8.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt8.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 9
DELETE A FROM rmt9.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt9.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 10
DELETE A FROM rmt10.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt10.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 11
DELETE A FROM rmt11.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt11.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 12
DELETE A FROM rmt12.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt12.tpcc.dbo.t_item SELECT * FROM INSERTED

end

```

item_iot_6.sql

Note: This script is only run on node 6.

```

-- file 9001_to_10800\item_iot.sql
--add item instead-of-trigger for partition 6

```

```

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

drop trigger iot_item
go

create trigger iot_item on item instead of update, insert, delete as
begin

--
-- remote node, partition 1
DELETE A FROM rmt1.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt1.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 2
DELETE A FROM rmt2.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt2.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 3
DELETE A FROM rmt3.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt3.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 4
DELETE A FROM rmt4.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt4.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 5
DELETE A FROM rmt5.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt5.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- local node (partition 6)
DELETE A FROM t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT t_item SELECT * FROM INSERTED

--
-- remote node, partition 7
DELETE A FROM rmt7.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt7.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 8
DELETE A FROM rmt8.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt8.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 9
DELETE A FROM rmt9.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt9.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 10
DELETE A FROM rmt10.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt10.tpcc.dbo.t_item SELECT * FROM INSERTED

```

```

--
-- remote node, partition 11
DELETE A FROM rmt11.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt11.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 12
DELETE A FROM rmt12.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt12.tpcc.dbo.t_item SELECT * FROM INSERTED

end

```

item_iot_7.sql

Note: This script is only run on node 7.

```

-- file 10801_to_12600\item_iot.sql
--add item instead-of-trigger for partition 7

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

drop trigger iot_item
go

create trigger iot_item on item instead of update, insert, delete as
begin

--
-- remote node, partition 1
DELETE A FROM rmt1.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt1.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 2
DELETE A FROM rmt2.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt2.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 3
DELETE A FROM rmt3.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt3.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 4
DELETE A FROM rmt4.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt4.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 5
DELETE A FROM rmt5.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt5.tpcc.dbo.t_item SELECT * FROM INSERTED

--

```

```

-- remote node, partition 6
DELETE A FROM rmt6.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt6.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- local node (partition 7)
DELETE A FROM t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT t_item SELECT * FROM INSERTED

--
-- remote node, partition 8
DELETE A FROM rmt8.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt8.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 9
DELETE A FROM rmt9.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt9.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 10
DELETE A FROM rmt10.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt10.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 11
DELETE A FROM rmt11.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt11.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 12
DELETE A FROM rmt12.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt12.tpcc.dbo.t_item SELECT * FROM INSERTED

end

```

item_iot_8.sql

Note: This script is only run on node 8.

```

-- file 12601_to_14400\item_iot.sql
--add item instead-of-trigger for partition 8

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

drop trigger iot_item
go

create trigger iot_item on item instead of update, insert, delete as
begin

--
-- remote node, partition 1

```

```

DELETE A FROM rmt1.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt1.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 2
DELETE A FROM rmt2.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt2.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 3
DELETE A FROM rmt3.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt3.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 4
DELETE A FROM rmt4.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt4.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 5
DELETE A FROM rmt5.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt5.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 6
DELETE A FROM rmt6.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt6.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 7
DELETE A FROM rmt7.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt7.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- local node (partition 8)
DELETE A FROM t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT t_item SELECT * FROM INSERTED

--
-- remote node, partition 9
DELETE A FROM rmt9.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt9.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 10
DELETE A FROM rmt10.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt10.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 11
DELETE A FROM rmt11.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt11.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 12
DELETE A FROM rmt12.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt12.tpcc.dbo.t_item SELECT * FROM INSERTED

end

```

item_iot_9.sql

Note: This script is only run on node 9.

```

-- file 14401_to_16200\item_iot.sql

--add item instead-of-trigger for partition 9

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

drop trigger iot_item
go

create trigger iot_item on item instead of update, insert, delete as
begin

--
-- remote node, partition 1
DELETE A FROM rmt1.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt1.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 2
DELETE A FROM rmt2.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt2.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 3
DELETE A FROM rmt3.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt3.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 4
DELETE A FROM rmt4.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt4.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 5
DELETE A FROM rmt5.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt5.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 6
DELETE A FROM rmt6.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt6.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 7
DELETE A FROM rmt7.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt7.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 8
DELETE A FROM rmt8.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt8.tpcc.dbo.t_item SELECT * FROM INSERTED

--

```

```

-- local node (partition 9)
DELETE A FROM t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT t_item SELECT * FROM INSERTED

--
-- remote node, partition 10
DELETE A FROM rmt10.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt10.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 11
DELETE A FROM rmt11.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt11.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 12
DELETE A FROM rmt12.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt12.tpcc.dbo.t_item SELECT * FROM INSERTED

end

```

item_iot_10.sql

Note: This script is only run on node 10.

```

-- file 16201_to_18000\item_iot.sql
--add item instead-of-trigger for partition 10

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

drop trigger iot_item
go

create trigger iot_item on item instead of update, insert, delete as
begin

--
-- remote node, partition 1
DELETE A FROM rmt1.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt1.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 2
DELETE A FROM rmt2.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt2.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 3
DELETE A FROM rmt3.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt3.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 4

```

```

DELETE A FROM rmt4.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt4.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 5
DELETE A FROM rmt5.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt5.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 6
DELETE A FROM rmt6.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt6.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 7
DELETE A FROM rmt7.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt7.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 8
DELETE A FROM rmt8.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt8.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 9
DELETE A FROM rmt9.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt9.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- local node (partition 10)
DELETE A FROM t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT t_item SELECT * FROM INSERTED

--
-- remote node, partition 11
DELETE A FROM rmt11.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt11.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 12
DELETE A FROM rmt12.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt12.tpcc.dbo.t_item SELECT * FROM INSERTED

end

```

item_iot_11.sql

Note: This script is only run on node 11.

```

-- file 18001_to_19800\item_iot.sql
--add item instead-of-trigger for partition 11

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

```

```

drop trigger iot_item
go

create trigger iot_item on item instead of update, insert, delete as
begin

--
-- remote node, partition 1
DELETE A FROM rmt1.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt1.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 2
DELETE A FROM rmt2.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt2.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 3
DELETE A FROM rmt3.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt3.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 4
DELETE A FROM rmt4.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt4.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 5
DELETE A FROM rmt5.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt5.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 6
DELETE A FROM rmt6.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt6.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 7
DELETE A FROM rmt7.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt7.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 8
DELETE A FROM rmt8.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt8.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 9
DELETE A FROM rmt9.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt9.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 10
DELETE A FROM rmt10.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt10.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- local node (partition 11)
DELETE A FROM t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT t_item SELECT * FROM INSERTED

--

```

```

-- remote node, partition 12
DELETE A FROM rmt12.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt12.tpcc.dbo.t_item SELECT * FROM INSERTED

end

```

item_iot_12.sql

Note: This script is only run on node 12.

```

-- file 19801_to_21600\item_iot.sql

--add item instead-of-trigger for partition 12

set ANSI_DEFAULTS on
set IMPLICIT_TRANSACTIONS off
go

use tpcc
go

drop trigger iot_item
go

create trigger iot_item on item instead of update, insert, delete as
begin

--
-- remote node, partition 1
DELETE A FROM rmt1.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt1.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 2
DELETE A FROM rmt2.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt2.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 3
DELETE A FROM rmt3.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt3.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 4
DELETE A FROM rmt4.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt4.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 5
DELETE A FROM rmt5.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt5.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 6
DELETE A FROM rmt6.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt6.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 7

```

```

DELETE A FROM rmt7.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt7.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 8
DELETE A FROM rmt8.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt8.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 9
DELETE A FROM rmt9.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt9.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 10
DELETE A FROM rmt10.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt10.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- remote node, partition 11
DELETE A FROM rmt11.tpcc.dbo.t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT rmt11.tpcc.dbo.t_item SELECT * FROM INSERTED

--
-- local node (partition 12)
DELETE A FROM t_item A, DELETED D WHERE A.i_id = D.i_id
INSERT t_item SELECT * FROM INSERTED

end

```

add_constraints_t1.sql

Note: This script is only run on node 1.

```

--file 1_to_1800\add_constraints_t.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- Drop any existing partitioning constraints
--

alter table t_warehouse drop constraint cnst_w_id
go

alter table t_district drop constraint cnst_d_w_id
go

alter table t_customer drop constraint cnst_c_w_id
go

alter table t_history drop constraint cnst_h_w_id
go

alter table t_stock drop constraint cnst_s_w_id

```

```

go

alter table t_orders drop constraint cnst_o_w_id
go

alter table t_order_line drop constraint cnst_ol_w_id
go

alter table t_new_order drop constraint cnst_no_w_id
go

-- Add partitioning constraints between 1 and 1800
--

alter table t_warehouse add constraint cnst_w_id check (w_id <= convert(int,1800))
go

alter table t_district add constraint cnst_d_w_id check (d_w_id <=
convert(int,1800))
go

alter table t_customer add constraint cnst_c_w_id check (c_w_id <=
convert(int,1800))
go

alter table t_history add constraint cnst_h_w_id check (h_w_id <= convert(int,1800))
go

alter table t_stock add constraint cnst_s_w_id check (s_w_id <= convert(int,1800))
go

alter table t_orders add constraint cnst_o_w_id check (o_w_id <= convert(int,1800))
go

alter table t_order_line add constraint cnst_ol_w_id check (ol_w_id <=
convert(int,1800))
go

alter table t_new_order add constraint cnst_no_w_id check (no_w_id <=
convert(int,1800))
go

```

add_constraints_t2.sql

Note: This script is only run on node 2.

```

--file 1801_to_3600\add_constraints_t.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- Drop any existing partitioning constraints
--

```

```

alter table t_warehouse drop constraint cnst_w_id
go

alter table t_district drop constraint cnst_d_w_id
go

alter table t_customer drop constraint cnst_c_w_id
go

alter table t_history drop constraint cnst_h_w_id
go

alter table t_stock drop constraint cnst_s_w_id
go

alter table t_orders drop constraint cnst_o_w_id
go

alter table t_order_line drop constraint cnst_ol_w_id
go

alter table t_new_order drop constraint cnst_no_w_id
go

-- Add partitioning constraints between 1801 and 3600
--

alter table t_warehouse add constraint cnst_w_id check (w_id between
convert(int,1801) and convert(int,3600))
go

alter table t_district add constraint cnst_d_w_id check (d_w_id between
convert(int,1801) and convert(int,3600))
go

alter table t_customer add constraint cnst_c_w_id check (c_w_id between
convert(int,1801) and convert(int,3600))
go

alter table t_history add constraint cnst_h_w_id check (h_w_id between
convert(int,1801) and convert(int,3600))
go

alter table t_stock add constraint cnst_s_w_id check (s_w_id between
convert(int,1801) and convert(int,3600))
go

alter table t_orders add constraint cnst_o_w_id check (o_w_id between
convert(int,1801) and convert(int,3600))
go

alter table t_order_line add constraint cnst_ol_w_id check (ol_w_id between
convert(int,1801) and convert(int,3600))
go

alter table t_new_order add constraint cnst_no_w_id check (no_w_id between
convert(int,1801) and convert(int,3600))
go

```

add_constraints_t3.sql

Note: This script is only run on node 3.

```

--file 3601_to_5400\add_constraints_t.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- Drop any existing partitioning constraints
--

alter table t_warehouse drop constraint cnst_w_id
go

alter table t_district drop constraint cnst_d_w_id
go

alter table t_customer drop constraint cnst_c_w_id
go

alter table t_history drop constraint cnst_h_w_id
go

alter table t_stock drop constraint cnst_s_w_id
go

alter table t_orders drop constraint cnst_o_w_id
go

alter table t_order_line drop constraint cnst_ol_w_id
go

alter table t_new_order drop constraint cnst_no_w_id
go

-- Add partitioning constraints between 3601 and 5400
--

alter table t_warehouse add constraint cnst_w_id check (w_id between
convert(int,3601) and convert(int,5400))
go

alter table t_district add constraint cnst_d_w_id check (d_w_id between
convert(int,3601) and convert(int,5400))
go

alter table t_customer add constraint cnst_c_w_id check (c_w_id between
convert(int,3601) and convert(int,5400))
go

alter table t_history add constraint cnst_h_w_id check (h_w_id between
convert(int,3601) and convert(int,5400))
go

alter table t_stock add constraint cnst_s_w_id check (s_w_id between
convert(int,3601) and convert(int,5400))
go

```



```

alter table t_orders add constraint cnst_o_w_id check (o_w_id between
convert(int,3601) and convert(int,5400))
go

alter table t_order_line add constraint cnst_ol_w_id check (ol_w_id between
convert(int,3601) and convert(int,5400))
go

alter table t_new_order add constraint cnst_no_w_id check (no_w_id between
convert(int,3601) and convert(int,5400))
go

```

add_constraints_t4.sql

Note: This script is only run on node 4.

```

--file 5401_to_7200\add_constraints_t.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- Drop any existing partitioning constraints
--

alter table t_warehouse drop constraint cnst_w_id
go

alter table t_district drop constraint cnst_d_w_id
go

alter table t_customer drop constraint cnst_c_w_id
go

alter table t_history drop constraint cnst_h_w_id
go

alter table t_stock drop constraint cnst_s_w_id
go

alter table t_orders drop constraint cnst_o_w_id
go

alter table t_order_line drop constraint cnst_ol_w_id
go

alter table t_new_order drop constraint cnst_no_w_id
go

-- Add partitioning constraints between 5401 and 7200
--

alter table t_warehouse add constraint cnst_w_id check (w_id between
convert(int,5401) and convert(int,7200))

```

```

go

alter table t_district add constraint cnst_d_w_id check (d_w_id between
convert(int,5401) and convert(int,7200))
go

alter table t_customer add constraint cnst_c_w_id check (c_w_id between
convert(int,5401) and convert(int,7200))
go

alter table t_history add constraint cnst_h_w_id check (h_w_id between
convert(int,5401) and convert(int,7200))
go

alter table t_stock add constraint cnst_s_w_id check (s_w_id between
convert(int,5401) and convert(int,7200))
go

alter table t_orders add constraint cnst_o_w_id check (o_w_id between
convert(int,5401) and convert(int,7200))
go

alter table t_order_line add constraint cnst_ol_w_id check (ol_w_id between
convert(int,5401) and convert(int,7200))
go

alter table t_new_order add constraint cnst_no_w_id check (no_w_id between
convert(int,5401) and convert(int,7200))
go

```

add_constraints_t5.sql

Note: This script is only run on node 5.

```

--file 7201_to_9000\add_constraints_t.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- Drop any existing partitioning constraints
--

alter table t_warehouse drop constraint cnst_w_id
go

alter table t_district drop constraint cnst_d_w_id
go

alter table t_customer drop constraint cnst_c_w_id
go

alter table t_history drop constraint cnst_h_w_id
go

```

```

alter table t_stock drop constraint cnst_s_w_id
go

alter table t_orders drop constraint cnst_o_w_id
go

alter table t_order_line drop constraint cnst_ol_w_id
go

alter table t_new_order drop constraint cnst_no_w_id
go

-- Add partitioning constraints between 7201 and 9000
--

alter table t_warehouse add constraint cnst_w_id check (w_id between
convert(int,7201) and convert(int,9000))
go

alter table t_district add constraint cnst_d_w_id check (d_w_id between
convert(int,7201) and convert(int,9000))
go

alter table t_customer add constraint cnst_c_w_id check (c_w_id between
convert(int,7201) and convert(int,9000))
go

alter table t_history add constraint cnst_h_w_id check (h_w_id between
convert(int,7201) and convert(int,9000))
go

alter table t_stock add constraint cnst_s_w_id check (s_w_id between
convert(int,7201) and convert(int,9000))
go

alter table t_orders add constraint cnst_o_w_id check (o_w_id between
convert(int,7201) and convert(int,9000))
go

alter table t_order_line add constraint cnst_ol_w_id check (ol_w_id between
convert(int,7201) and convert(int,9000))
go

alter table t_new_order add constraint cnst_no_w_id check (no_w_id between
convert(int,7201) and convert(int,9000))
go

```

add_constraints_t6.sql

Note: This script is only run on node 6.

```

--file 9001_to_10800\add_constraints_t.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc

```

```

go

-- Drop any existing partitioning constraints
--

alter table t_warehouse drop constraint cnst_w_id
go

alter table t_district drop constraint cnst_d_w_id
go

alter table t_customer drop constraint cnst_c_w_id
go

alter table t_history drop constraint cnst_h_w_id
go

alter table t_stock drop constraint cnst_s_w_id
go

alter table t_orders drop constraint cnst_o_w_id
go

alter table t_order_line drop constraint cnst_ol_w_id
go

alter table t_new_order drop constraint cnst_no_w_id
go

-- Add partitioning constraints between 9001 and 10800
--

alter table t_warehouse add constraint cnst_w_id check (w_id between
convert(int,9001) and convert(int,10800))
go

alter table t_district add constraint cnst_d_w_id check (d_w_id between
convert(int,9001) and convert(int,10800))
go

alter table t_customer add constraint cnst_c_w_id check (c_w_id between
convert(int,9001) and convert(int,10800))
go

alter table t_history add constraint cnst_h_w_id check (h_w_id between
convert(int,9001) and convert(int,10800))
go

alter table t_stock add constraint cnst_s_w_id check (s_w_id between
convert(int,9001) and convert(int,10800))
go

alter table t_orders add constraint cnst_o_w_id check (o_w_id between
convert(int,9001) and convert(int,10800))
go

alter table t_order_line add constraint cnst_ol_w_id check (ol_w_id between
convert(int,9001) and convert(int,10800))
go

alter table t_new_order add constraint cnst_no_w_id check (no_w_id between
convert(int,9001) and convert(int,10800))
go

```

add_constraints_t7.sql

Note: This script is only run on node 7.

```
--file 10801_to_12600\add_constraints_t.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- Drop any existing partitioning constraints
--

alter table t_warehouse drop constraint cnst_w_id
go

alter table t_district drop constraint cnst_d_w_id
go

alter table t_customer drop constraint cnst_c_w_id
go

alter table t_history drop constraint cnst_h_w_id
go

alter table t_stock drop constraint cnst_s_w_id
go

alter table t_orders drop constraint cnst_o_w_id
go

alter table t_order_line drop constraint cnst_ol_w_id
go

alter table t_new_order drop constraint cnst_no_w_id
go

-- Add partitioning constraints between 10801 and 12600
--

alter table t_warehouse add constraint cnst_w_id check (w_id between
convert(int,10801) and convert(int,12600))
go

alter table t_district add constraint cnst_d_w_id check (d_w_id between
convert(int,10801) and convert(int,12600))
go

alter table t_customer add constraint cnst_c_w_id check (c_w_id between
convert(int,10801) and convert(int,12600))
go

alter table t_history add constraint cnst_h_w_id check (h_w_id between
convert(int,10801) and convert(int,12600))
```

go

```
alter table t_stock add constraint cnst_s_w_id check (s_w_id between
convert(int,10801) and convert(int,12600))
go
```

```
alter table t_orders add constraint cnst_o_w_id check (o_w_id between
convert(int,10801) and convert(int,12600))
go
```

```
alter table t_order_line add constraint cnst_ol_w_id check (ol_w_id between
convert(int,10801) and convert(int,12600))
go
```

```
alter table t_new_order add constraint cnst_no_w_id check (no_w_id between
convert(int,10801) and convert(int,12600))
go
```

add_constraints_t8.sql

Note: This script is only run on node 8.

```
--file 12601_to_14400\add_constraints_t.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- Drop any existing partitioning constraints
--

alter table t_warehouse drop constraint cnst_w_id
go

alter table t_district drop constraint cnst_d_w_id
go

alter table t_customer drop constraint cnst_c_w_id
go

alter table t_history drop constraint cnst_h_w_id
go

alter table t_stock drop constraint cnst_s_w_id
go

alter table t_orders drop constraint cnst_o_w_id
go

alter table t_order_line drop constraint cnst_ol_w_id
go

alter table t_new_order drop constraint cnst_no_w_id
go
```

```

-- Add partitioning constraints between 12601 and 14400
--

alter table t_warehouse add constraint cnst_w_id check (w_id between
convert(int,12601) and convert(int,14400))
go

alter table t_district add constraint cnst_d_w_id check (d_w_id between
convert(int,12601) and convert(int,14400))
go

alter table t_customer add constraint cnst_c_w_id check (c_w_id between
convert(int,12601) and convert(int,14400))
go

alter table t_history add constraint cnst_h_w_id check (h_w_id between
convert(int,12601) and convert(int,14400))
go

alter table t_stock add constraint cnst_s_w_id check (s_w_id between
convert(int,12601) and convert(int,14400))
go

alter table t_orders add constraint cnst_o_w_id check (o_w_id between
convert(int,12601) and convert(int,14400))
go

alter table t_order_line add constraint cnst_ol_w_id check (ol_w_id between
convert(int,12601) and convert(int,14400))
go

alter table t_new_order add constraint cnst_no_w_id check (no_w_id between
convert(int,12601) and convert(int,14400))
go

```

add_constraints_t9.sql

Note: This script is only run on node 9.

```

--file 14401_to_16200\add_constraints_t.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- Drop any existing partitioning constraints
--

alter table t_warehouse drop constraint cnst_w_id
go

alter table t_district drop constraint cnst_d_w_id
go

alter table t_customer drop constraint cnst_c_w_id

```

```

go

alter table t_history drop constraint cnst_h_w_id
go

alter table t_stock drop constraint cnst_s_w_id
go

alter table t_orders drop constraint cnst_o_w_id
go

alter table t_order_line drop constraint cnst_ol_w_id
go

alter table t_new_order drop constraint cnst_no_w_id
go

-- Add partitioning constraints between 14401 and 16200
--

alter table t_warehouse add constraint cnst_w_id check (w_id between
convert(int,14401) and convert(int,16200))
go

alter table t_district add constraint cnst_d_w_id check (d_w_id between
convert(int,14401) and convert(int,16200))
go

alter table t_customer add constraint cnst_c_w_id check (c_w_id between
convert(int,14401) and convert(int,16200))
go

alter table t_history add constraint cnst_h_w_id check (h_w_id between
convert(int,14401) and convert(int,16200))
go

alter table t_stock add constraint cnst_s_w_id check (s_w_id between
convert(int,14401) and convert(int,16200))
go

alter table t_orders add constraint cnst_o_w_id check (o_w_id between
convert(int,14401) and convert(int,16200))
go

alter table t_order_line add constraint cnst_ol_w_id check (ol_w_id between
convert(int,14401) and convert(int,16200))
go

alter table t_new_order add constraint cnst_no_w_id check (no_w_id between
convert(int,14401) and convert(int,16200))
go

```

add_constraints_t10.sql

Note: This script is only run on node 10.

```

--file 16201_to_18000\add_constraints_t.sql

```

```

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- Drop any existing partitioning constraints
--

alter table t_warehouse drop constraint cnst_w_id
go

alter table t_district drop constraint cnst_d_w_id
go

alter table t_customer drop constraint cnst_c_w_id
go

alter table t_history drop constraint cnst_h_w_id
go

alter table t_stock drop constraint cnst_s_w_id
go

alter table t_orders drop constraint cnst_o_w_id
go

alter table t_order_line drop constraint cnst_ol_w_id
go

alter table t_new_order drop constraint cnst_no_w_id
go

-- Add partitioning constraints between 16201 and 18000
--

alter table t_warehouse add constraint cnst_w_id check (w_id between
convert(int,16201) and convert(int,18000))
go

alter table t_district add constraint cnst_d_w_id check (d_w_id between
convert(int,16201) and convert(int,18000))
go

alter table t_customer add constraint cnst_c_w_id check (c_w_id between
convert(int,16201) and convert(int,18000))
go

alter table t_history add constraint cnst_h_w_id check (h_w_id between
convert(int,16201) and convert(int,18000))
go

alter table t_stock add constraint cnst_s_w_id check (s_w_id between
convert(int,16201) and convert(int,18000))
go

alter table t_orders add constraint cnst_o_w_id check (o_w_id between
convert(int,16201) and convert(int,18000))
go

alter table t_order_line add constraint cnst_ol_w_id check (ol_w_id between
convert(int,16201) and convert(int,18000))

```

```

go

alter table t_new_order add constraint cnst_no_w_id check (no_w_id between
convert(int,16201) and convert(int,18000))
go

```

add_constraints_t11.sql

Note: This script is only run on node 11.

```

--file 18001_to_19800\add_constraints_t.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- Drop any existing partitioning constraints
--

alter table t_warehouse drop constraint cnst_w_id
go

alter table t_district drop constraint cnst_d_w_id
go

alter table t_customer drop constraint cnst_c_w_id
go

alter table t_history drop constraint cnst_h_w_id
go

alter table t_stock drop constraint cnst_s_w_id
go

alter table t_orders drop constraint cnst_o_w_id
go

alter table t_order_line drop constraint cnst_ol_w_id
go

alter table t_new_order drop constraint cnst_no_w_id
go

-- Add partitioning constraints between 18001 and 19800
--

alter table t_warehouse add constraint cnst_w_id check (w_id between
convert(int,18001) and convert(int,19800))
go

alter table t_district add constraint cnst_d_w_id check (d_w_id between
convert(int,18001) and convert(int,19800))
go

```

```

alter table t_customer add constraint cnst_c_w_id check (c_w_id between
convert(int,18001) and convert(int,19800))
go

alter table t_history add constraint cnst_h_w_id check (h_w_id between
convert(int,18001) and convert(int,19800))
go

alter table t_stock add constraint cnst_s_w_id check (s_w_id between
convert(int,18001) and convert(int,19800))
go

alter table t_orders add constraint cnst_o_w_id check (o_w_id between
convert(int,18001) and convert(int,19800))
go

alter table t_order_line add constraint cnst_ol_w_id check (ol_w_id between
convert(int,18001) and convert(int,19800))
go

alter table t_new_order add constraint cnst_no_w_id check (no_w_id between
convert(int,18001) and convert(int,19800))
go

```

add_constraints_t12.sql

Note: This script is only run on node 12.

```

--file 19801_to_21600\add_constraints_t.sql

set ansi_warnings on
set ansi_nulls on
go

use tpcc
go

-- Drop any existing partitioning constraints
--

alter table t_warehouse drop constraint cnst_w_id
go

alter table t_district drop constraint cnst_d_w_id
go

alter table t_customer drop constraint cnst_c_w_id
go

alter table t_history drop constraint cnst_h_w_id
go

alter table t_stock drop constraint cnst_s_w_id
go

alter table t_orders drop constraint cnst_o_w_id
go

```

```

alter table t_order_line drop constraint cnst_ol_w_id
go

alter table t_new_order drop constraint cnst_no_w_id
go

-- Add partitioning constraints between 19801 and 21600
--

alter table t_warehouse add constraint cnst_w_id check (w_id >= convert(int,19801))
go

alter table t_district add constraint cnst_d_w_id check (d_w_id >=
convert(int,19801))
go

alter table t_customer add constraint cnst_c_w_id check (c_w_id >=
convert(int,19801))
go

alter table t_history add constraint cnst_h_w_id check (h_w_id >=
convert(int,19801))
go

alter table t_stock add constraint cnst_s_w_id check (s_w_id >= convert(int,19801))
go

alter table t_orders add constraint cnst_o_w_id check (o_w_id >= convert(int,19801))
go

alter table t_order_line add constraint cnst_ol_w_id check (ol_w_id >=
convert(int,19801))
go

alter table t_new_order add constraint cnst_no_w_id check (no_w_id >=
convert(int,19801))
go

```

tpcc.h

```

// File: TPC.C.H
// Microsoft TPC-C Kit Ver. 4.30
// Copyright Microsoft, 1996, 1997, 1998, 1999,
2000
// Purpose: Header file for TPC-C database loader

// Build number of TPC Benchmark Kit
#define TPCKIT_VER "4.30"

// General headers
#include <windows.h>
#include <winbase.h>
#include <stdlib.h>
#include <stdio.h>
#include <process.h>
#include <stddef.h>
#include <stdarg.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>

```

```

#include <sys/types.h>

// ODBC headers
#include <sql.h>
#include <sqlext.h>
#include <odbcss.h>

// General constants
#define MILLI 1000
#define FALSE 0
#define TRUE 1
#define UNDEF -1
#define MINPRINTASCII 32
#define MAXPRINTASCII 126

// Default environment constants
#define SERVER ""
#define DATABASE "tpcc"
#define USER "sa"
#define PASSWORD ""

// Default loader arguments
#define BATCH 10000
#define DEFLOADPACKSIZE 32768
#define LOADER_RES_FILE "logs\\load.out"
#define LOADER_NURAND_C 123
#define DEF_STARTING_WAREHOUSE 1
#define BUILD_INDEX 1 // build both
                        data and indexes
#define INDEX_ORDER 1 // build
                        indexes before load
#define SCALE_DOWN 0 // build a normal
                        scale database
#define INDEX_SCRIPT_PATH "scripts"

typedef struct
{
    char *server;
    char *database;
    char *user;
    char *password;
    BOOL tables_all;
    // 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 *synch_servername;
    long case_sensitivity;
    long starting_warehouse;
    long build_index;
    long index_order;
    long scale_down;
    char *index_script_path;
}

```

```

} TPCCLDR_ARGS;

// String length constants
#define SERVER_NAME_LEN 20
#define DATABASE_NAME_LEN 20
#define USER_NAME_LEN 20
#define PASSWORD_LEN 20
#define TABLE_NAME_LEN 20
#define I_DATA_LEN 50
#define I_NAME_LEN 24
#define BRAND_LEN 1
#define LAST_NAME_LEN 16
#define W_NAME_LEN 10
#define ADDRESS_LEN 20
#define STATE_LEN 2
#define ZIP_LEN 9
#define S_DIST_LEN 24
#define S_DATA_LEN 50
#define D_NAME_LEN 10
#define FIRST_NAME_LEN 16
#define MIDDLE_NAME_LEN 2
#define PHONE_LEN 16
#define CREDIT_LEN 2
#define C_DATA_LEN 500
#define H_DATA_LEN 24
#define DIST_INFO_LEN 24
#define MAX_OL_NEW_ORDER_ITEMS 15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN 25
#define OL_DIST_INFO_LEN 24
#define C_SINCE_LEN 23
#define H_DATE_LEN 23
#define OL_DELIVERY_D_LEN 23
#define O_ENTRY_D_LEN 23

// Functions in random.c
void seed();
long irand();
double drand();
void WUCreate();
short WURand();
long RandomNumber(long lower, long upper);

// Functions in getargs.c;
void GetArgsLoader();
void GetArgsLoaderUsage();

// Functions in time.c
long TimeNow();

// Functions in strings.c
void MakeAddress();
void LastName();
int MakeAlphaString();
int MakeOriginalAlphaString();
int MakeNumberString();
int MakeZipNumberString();
void InitString();
void InitAddress();
void PaddString();

```

tpccldr.c

```
// File: TPCCLDR.C
// Microsoft TPC-C Kit Ver. 4.30
// Copyright Microsoft, 1996, 1997, 1998, 1999,
2000
// Purpose: Source file for TPC-C database loader

// Includes
#include "tpcc.h"
#include "search.h"

// Defines
#define MAXITEMS 100000
#define MAXITEMS_SCALE_DOWN 100
#define CUSTOMERS_PER_DISTRICT 3000
#define CUSTOMERS_SCALE_DOWN 30
#define DISTRICT_PER_WAREHOUSE 10
#define ORDERS_PER_DISTRICT 3000
#define ORDERS_SCALE_DOWN 30
#define MAX_CUSTOMER_THREADS 2
#define MAX_ORDER_THREADS 3
#define MAX_MAIN_THREADS 4

// Functions declarations

void HandleErrorDBC (SQLHDBC hdbc1);

void CheckSQL();
void CheckDataBase();

long NURand();
void LoadItem();
void LoadWarehouse();

void Stock();
void District();

void LoadCustomer();
void CustomerBufInit();
void CustomerBufLoad();
void LoadCustomerTable();
void LoadHistoryTable();

void LoadOrders();
void OrdersBufInit();
void OrdersBufLoad();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void GetPermutation();
void CheckForCommit();
void OpenConnections();
void BuildIndex();
void FormatDate ();

// Shared memory structures

typedef struct
{
    long ol;
```

```
long ol_i_id;
long ol_supply_w_id;
short ol_quantity;
double ol_amount;
char ol_dist_info[DIST_INFO_LEN+1];
    char ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCT;

typedef struct
{
    long o_id;
    short o_d_id;
    long o_w_id;
    long o_c_id;
    short o_carrier_id;
    short o_ol_cnt;
    short o_all_local;
    ORDER_LINE_STRUCT o_ol[15];
} ORDERS_STRUCT;

typedef struct
{
    long c_id;
    short c_d_id;
    long c_w_id;
    char c_first[FIRST_NAME_LEN+1];
    char c_middle[MIDDLE_NAME_LEN+1];
    char c_last[LAST_NAME_LEN+1];
    char c_street_1[ADDRESS_LEN+1];
    char c_street_2[ADDRESS_LEN+1];
    char c_city[ADDRESS_LEN+1];
    char c_state[STATE_LEN+1];
    char c_zip[ZIP_LEN+1];
    char c_phone[PHONE_LEN+1];
    char c_credit[CREDIT_LEN+1];
    double c_credit_lim;
    double c_discount;
    // fix to avoid ODBC float to numeric conversion problem.
    // double c_balance;
    char c_balance[6];

    double c_ytd_payment;
    short c_payment_cnt;
    short c_delivery_cnt;
    char c_data[C_DATA_LEN+1];
    double h_amount;
    char h_data[H_DATA_LEN+1];
} CUSTOMER_STRUCT;

typedef struct
{
    char c_last[LAST_NAME_LEN+1];
    char c_first[FIRST_NAME_LEN+1];
    long c_id;
} CUSTOMER_SORT_STRUCT;

typedef struct
{
    long time_start;
} LOADER_TIME_STRUCT;
```



```

// Global variables

char      szLastError[300];

HENV      henv;

HDBC      v_hdbc;                                // for SQL
Server version verification
HDBC      i_hdbc1;                                // for ITEM table
HDBC      w_hdbc1;                                // for WAREHOUSE,
DISTRICT, STOCK
HDBC      c_hdbc1;                                // for CUSTOMER
HDBC      c_hdbc2;                                // for HISTORY
HDBC      o_hdbc1;                                // for ORDERS
HDBC      o_hdbc2;                                // for NEW-ORDER

HDBC      o_hdbc3;                                // for ORDER-LINE

HSTMT     v_hstmt;                                // for SQL Server
version verification
HSTMT     i_hstmt1;
HSTMT     w_hstmt1;
HSTMT     c_hstmt1, c_hstmt2;
HSTMT     o_hstmt1, o_hstmt2, o_hstmt3;

ORDERS_STRUCT  orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT customer_buf[CUSTOMERS_PER_DISTRICT];
long           orders_rows_loaded;
long           new_order_rows_loaded;
long           order_line_rows_loaded;
long           history_rows_loaded;
long           customer_rows_loaded;
long           stock_rows_loaded;
long           district_rows_loaded;
long           item_rows_loaded;
long           warehouse_rows_loaded;
long           main_time_start;
long           main_time_end;
long           max_items;
long           customers_per_district;
long           orders_per_district;
long           first_new_order;
long           last_new_order;

TPCC_LDR_ARGS *aptr, args;

// support for log directory determined by environment
#define LOGBASE_MAXLEN 128
int      UseAltLogDir = 0;
char     LogBase[LOGBASE_MAXLEN];
char     LogFile[2*LOGBASE_MAXLEN];
#define LOGFILE_CREATE(name) \
        strcpy(LogFile, LogBase); \
        strcat(LogFile, name);

//=====
//
// Function name: main
//
//=====

int main(int argc, char **argv)

```

```

{
    DWORD      dwThreadId[MAX_MAIN_THREADS];
    HANDLE     hThread[MAX_MAIN_THREADS];
    FILE       *fLoader;
    char       buffer[255];
    int        i;

    for (i=0; i<MAX_MAIN_THREADS; i++)
        hThread[i] = NULL;

    printf("\n*****\n");
    printf("\n*                               *");
    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");

    // process command line arguments

    aptr = &args;
    GetArgsLoader(argc, argv, aptr);

    // process environment variables
    if ( GetEnvironmentVariable("LOGBASE", LogBase, LOGBASE_MAXLEN)
        {
            UseAltLogDir = 1;
            strcat(LogBase, "\\logs");
            printf("Will use %s for log files\n", LogBase);
        }
    else
        strcpy(LogBase, "logs");

    // verify database and tables exist before attempting to load

    CheckDataBase();

    printf("Build interface is ODBC.\n");

    if (aptr->build_index == 0)
        printf("Data load only - no index creation.\n");
    else
        printf("Data load and index creation.\n");

    if (aptr->index_order == 0)
        printf("Clustered indexes will be created after bulk load.\n");
    else
        printf("Clustered indexes will be created before bulk load.\n");

    // set database scale values
    if (aptr->scale_down == 1)
    {
        printf("*** Scaled Down Database ***\n");
        max_items = MAXITEMS_SCALE_DOWN;
        customers_per_district = CUSTOMERS_SCALE_DOWN;
        orders_per_district = ORDERS_SCALE_DOWN;
        first_new_order = 0;
        last_new_order = 30;
    }
    else
    {

```

```

        max_items = MAXITEMS;
        customers_per_district = CUSTOMERS_PER_DISTRICT;
        orders_per_district = ORDERS_PER_DISTRICT;
        first_new_order = 2100;
        last_new_order = 3000;
    }

    // open connections to SQL Server
    OpenConnections();

    // open file for loader results
    fLoader = fopen(aptr->loader_res_file, "w");

    if (fLoader == NULL)
    {
        printf("Error, loader result file open failed.");
        exit(-1);
    }

    // start loading data
    sprintf(buffer, "TPC-C load started for %ld warehouses.\n", aptr->num_warehouses);

    printf("%s", buffer);
    fprintf(fLoader, "%s", buffer);

    main_time_start = (TimeNow() / MILLI);

    // start parallel load threads

    if (aptr->tables_all || aptr->table_item)
    {
        fprintf(fLoader, "\nStarting loader threads for: item\n");
        hThread[0] = CreateThread(NULL,
                                0,
                                (LPTHREAD_START_ROUTINE) LoadItem,
                                NULL,
                                0,
                                &dwThreadID[0]);

        if (hThread[0] == NULL)
        {
            printf("Error, failed in creating creating thread =
0.\n");
            exit(-1);
        }
    }

    if (aptr->tables_all || aptr->table_warehouse)
    {
        fprintf(fLoader, "Starting loader threads for: warehouse\n");
        hThread[1] = CreateThread(NULL,
                                0,
                                (LPTHREAD_START_ROUTINE) LoadWarehouse,

```

```

                                NULL,
                                0,
                                &dwThreadID[1]);

        if (hThread[1] == NULL)
        {
            printf("Error, failed in creating creating thread =
1.\n");
            exit(-1);
        }
    }

    if (aptr->tables_all || aptr->table_customer)
    {
        fprintf(fLoader, "Starting loader threads for: customer\n");
        hThread[2] = CreateThread(NULL,
                                0,
                                (LPTHREAD_START_ROUTINE) LoadCustomer,
                                NULL,
                                0,
                                &dwThreadID[2]);

        if (hThread[2] == NULL)
        {
            printf("Error, failed in creating creating main thread
= 2.\n");
            exit(-1);
        }
    }

    if (aptr->tables_all || aptr->table_orders)
    {
        fprintf(fLoader, "Starting loader threads for: orders\n");
        hThread[3] = CreateThread(NULL,
                                0,
                                (LPTHREAD_START_ROUTINE) LoadOrders,
                                NULL,
                                0,
                                &dwThreadID[3]);

        if (hThread[3] == NULL)
        {
            printf("Error, failed in creating creating main thread
= 3.\n");
            exit(-1);
        }
    }

    // Wait for threads to finish...
    for (i=0; i<MAX_MAIN_THREADS; i++)
    {
        if (hThread[i] != NULL)
        {

```

```

        WaitForSingleObject( hThread[i], INFINITE );
        CloseHandle(hThread[i]);
        hThread[i] = NULL;
    }

    main_time_end = (TimeNow() / MILLI);

    sprintf(buffer, "\nTPC-C load completed successfully in %ld minutes.\n",
            (main_time_end - main_time_start)/60);

    printf("%s",buffer);
    fprintf(fLoader, "%s", buffer);

    fclose(fLoader);

    SQLFreeEnv(henv);

    exit(0);

    return 0;
}

//=====
//
// Function name: LoadItem
//
//=====

void LoadItem()
{
    long    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];

    // 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, "t_item");

    LOGFILE_CREATE("\\item.err")
    rc = bcp_init(i_hdbc1, name, NULL, LogFile, 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, CHECK_CONSTRAINTS");
            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("idxitmc1");
}

//=====
//
// 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];

    // Seed with unique number
    seed(aptr->starting_warehouse + 1);

    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, "t_warehouse");

    LOGFILE_CREATE("\\house.err")
    rc = bcp_init(w_hdbc1, name, NULL, LogFile, DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcphint, "tablock, order (w_id), ROWS_PER_BATCH = %d,
CHECK_CONSTRAINTS", aptr->num_warehouses);
        rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
    }
}

```

```

rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0, W_NAME_LEN, NULL, 0, 0, 2);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_street_1, 0, ADDRESS_LEN, NULL, 0, 0,
3);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_street_2, 0, ADDRESS_LEN, NULL, 0, 0,
4);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_city, 0, ADDRESS_LEN, NULL, 0, 0, 5);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_state, 0, STATE_LEN, NULL, 0, 0, 6);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_zip, 0, ZIP_LEN, NULL, 0, 0, 7);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 8);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 9);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

time_start = (TimeNow() / MILLI);

warehouse_rows_loaded = 0;

for (w_id = (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 != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);

    warehouse_rows_loaded++;
    CheckForCommit(w_hdbc1, i_hstmt1, warehouse_rows_loaded,
"warehouse", &time_start);
}

```

```

}

rcint = bcp_done(w_hdbc1);
if (rcint < 0)
    HandleErrorDBC(w_hdbc1);

printf("Finished loading warehouse table.\n");

// if build index after load...
if ((aptr->build_index == 1) && (aptr->index_order == 0))
    BuildIndex("idxwarc1");

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;
    int    w_id;
    RETCODE rc;
    DBINT  rcint;
    char   bcphint[128];

    // Seed with unique number
    seed(aptr->starting_warehouse + 2);

    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, "t_district");

    LOGFILE_CREATE("\\district.err")
    rc = bcp_init(w_hdbc1, name, NULL, LogFile, 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, CHECK_CONSTRAINTS", (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 != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        district_rows_loaded++;
        CheckForCommit(w_hdbc1, w_hstmt1,
district_rows_loaded, "district", &time_start);
    }
}
rcint = bcp_done(w_hdbc1);
if (rcint < 0)
    HandleErrorDBC(w_hdbc1);
printf("Finished loading district table.\n");
// if build index after load...
if ((aptr->build_index == 1) && (aptr->index_order == 0))
    BuildIndex("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];
// Seed with unique number
seed(aptr->starting_warehouse + 3);
// if build index before load...
if ((aptr->build_index == 1) && (aptr->index_order == 1))
    BuildIndex("idxstkcl");
sprintf(name, "%s.%s", aptr->database, "t_stock");
LOGFILE_CREATE("\\stock.err")
rc = bcp_init(w_hdbc1, name, NULL, LogFile, DB_IN);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);
if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    sprintf(bcphint, "tablock, order (s_i_id, s_w_id),
ROWS_PER_BATCH = %u, CHECK CONSTRAINTS", (aptr->num_warehouses * 10000));
    rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
}
rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);
bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
2);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *) &s_quantity, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 3);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01, 0, S_DIST_LEN, NULL, 0, 0, 4);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02, 0, S_DIST_LEN, NULL, 0, 0, 5);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03, 0, S_DIST_LEN, NULL, 0, 0, 6);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

```

```

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04, 0, S_DIST_LEN, NULL, 0, 0, 7);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05, 0, S_DIST_LEN, NULL, 0, 0, 8);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06, 0, S_DIST_LEN, NULL, 0, 0, 9);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07, 0, S_DIST_LEN, NULL, 0, 0, 10);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0, S_DIST_LEN, NULL, 0, 0, 11);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09, 0, S_DIST_LEN, NULL, 0, 0, 12);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0, S_DIST_LEN, NULL, 0, 0, 13);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 14);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_order_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 15);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_remote_cnt, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 16);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0, S_DATA_LEN, NULL, 0, 0, 17);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

s_ytd = s_order_cnt = s_remote_cnt = 0;

time_start = (TimeNow() / 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 != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

stock_rows_loaded++;
CheckForCommit(w_hdbc1, w_hstmt1, stock_rows_loaded,
"stock", &time_start);
    }
}

rcint = bcp_done(w_hdbc1);
if (rcint < 0)
    HandleErrorDBC(w_hdbc1);

printf("Finished loading stock table.\n");

SQLFreeStmt(w_hstmt1, SQL_DROP);
SQLDisconnect(w_hdbc1);
SQLFreeConnect(w_hdbc1);

// if build index after load...
if ((aptr->build_index == 1) && (aptr->index_order == 0))
    BuildIndex("idxstkcl");

return;
}

//=====
//
// Function : LoadCustomer
//
//=====

void LoadCustomer()
{
    LOADER_TIME_STRUCT customer_time_start;
    LOADER_TIME_STRUCT history_time_start;
    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];
    char rc_l;
    // SQLSMALLINT // SQLSMALLINT
    recnum, MsgLen;

```

```

// SQLCHAR                                SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];              NativeError;
// SQLINTEGER

// Seed with unique number
seed(aptr->starting_warehouse + 4);

printf("Loading customer and history tables...\n");

// if build index before load..
if ((aptr->build_index == 1) && (aptr->index_order == 1))
    BuildIndex("idxcuscl");

// Initialize bulk copy
sprintf(name, "%s..%s", aptr->database, "t_customer");

LOGFILE_CREATE("\\customer.err")
rc = bcp_init(c_hdbc1, name, NULL, LogFile, 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, CHECK CONSTRAINTS", (aptr->num_warehouses * 30000));
    rc = bcp_control(c_hdbc1, BCPHINTS, (void*) bcphint);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc1);
}

sprintf(name, "%s..%s", aptr->database, "t_history");

LOGFILE_CREATE("\\history.err")
rc = bcp_init(c_hdbc2, name, NULL, LogFile, DB_IN);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

sprintf(bcphint, "tablock, order (h_w_id, h_d_id, h_c_id, h_date),
ROWS_PER_BATCH = %u, CHECK CONSTRAINTS", (aptr->num_warehouses * 30000));
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");

    // build non-clustered index
    if (aptr->build_index == 1)
        BuildIndex("idxcusnc");

    // Output the NURAND used for the loader into C_FIRST for C_ID = 1,
    // C_W_ID = 1, and C_D_ID = 1
    sprintf(cmd, "isql -S%s -U%s -P%s -d%s -e -Q\"update t_customer set
c_first = 'C_LOAD = %d' where c_id = 1 and c_w_id = 1 and c_d_id = 1\" >
%s\\nurand_load.log",
        aptr->server,
        aptr->user,
        aptr->password,
        aptr->database,
        LOADER_NURAND_C,
        LogBase);

    system(cmd);

    SQLFreeStmt(c_hstmt1, SQL_DROP);
    SQLDisconnect(c_hdbc1);
    SQLFreeConnect(c_hdbc1);

    SQLFreeStmt(c_hstmt2, SQL_DROP);
    SQLDisconnect(c_hdbc2);
    SQLFreeConnect(c_hdbc2);

    return;
}

//=====
//
// Function : CustomerBufInit
//
//=====
void CustomerBufInit()
{
    int i;

    for (i=0;i<customers_per_district;i++)
    {
        customer_buf[i].c_id = 0;

```

```

        customer_buf[i].c_d_id = 0;
        customer_buf[i].c_w_id = 0;

        strcpy(customer_buf[i].c_first,"");
        strcpy(customer_buf[i].c_middle,"");
        strcpy(customer_buf[i].c_last,"");
        strcpy(customer_buf[i].c_street_1,"");
        strcpy(customer_buf[i].c_street_2,"");
        strcpy(customer_buf[i].c_city,"");
        strcpy(customer_buf[i].c_state,"");
        strcpy(customer_buf[i].c_zip,"");
        strcpy(customer_buf[i].c_phone,"");
        strcpy(customer_buf[i].c_credit,"");

        customer_buf[i].c_credit_lim = 0;
        customer_buf[i].c_discount = (float) 0;

        // fix to avoid ODBC float to numeric conversion problem.
        customer_buf[i].c_balance = 0;
        strcpy(customer_buf[i].c_balance,"");

        customer_buf[i].c_ytd_payment = 0;
        customer_buf[i].c_payment_cnt = 0;
        customer_buf[i].c_delivery_cnt = 0;

        strcpy(customer_buf[i].c_data,"");

        customer_buf[i].h_amount = 0;

        strcpy(customer_buf[i].h_data,"");
    }
}

//=====
//
// Function : CustomerBufLoad
//
// Fills shared buffer for HISTORY and CUSTOMER
//=====

void CustomerBufLoad(int d_id, int w_id)
{
    long i;
    CUSTOMER_SORT_STRUCT c[CUSTOMERS_PER_DISTRICT];

    for (i=0;i<customers_per_district;i++)
    {
        if (i < 1000)
            LastName(i, c[i].c_last);
        else
            LastName(NURand(255,0,999,LOADER_NURAND_C),
c[i].c_last);

        MakeAlphaString(8,16,FIRST_NAME_LEN, c[i].c_first);

        c[i].c_id = i+1;
    }
}

```

```

printf("...Loading customer buffer for: d_id = %d, w_id = %d\n",
      d_id, w_id);

for (i=0;i<customers_per_district;i++)
{
    customer_buf[i].c_d_id = d_id;
    customer_buf[i].c_w_id = w_id;
    customer_buf[i].h_amount = 10.0;

    customer_buf[i].c_ytd_payment = 10.0;

    customer_buf[i].c_payment_cnt = 1;
    customer_buf[i].c_delivery_cnt = 0;

    // Generate CUSTOMER and HISTORY data

    customer_buf[i].c_id = c[i].c_id;

    strcpy(customer_buf[i].c_first, c[i].c_first);
    strcpy(customer_buf[i].c_last, c[i].c_last);

    customer_buf[i].c_middle[0] = 'O';
    customer_buf[i].c_middle[1] = 'E';

    MakeAddress(customer_buf[i].c_street_1,
               customer_buf[i].c_street_2,
               customer_buf[i].c_city,
               customer_buf[i].c_state,
               customer_buf[i].c_zip);

    MakeNumberString(16, 16, PHONE_LEN, customer_buf[i].c_phone);

    if (RandomNumber(1L, 100L) > 10)
        customer_buf[i].c_credit[0] = 'G';
    else
        customer_buf[i].c_credit[0] = 'B';
    customer_buf[i].c_credit[1] = 'C';

    customer_buf[i].c_credit_lim = 50000.0;
    customer_buf[i].c_discount = ((float) RandomNumber(0L, 5000L)) /
10000.0;

    // fix to avoid ODBC float to numeric conversion problem.
    // customer_buf[i].c_balance = -10.0;
    strcpy(customer_buf[i].c_balance, "-10.0");

    MakeAlphaString(300, 500, C_DATA_LEN, customer_buf[i].c_data);

    // Generate HISTORY data
    MakeAlphaString(12, 24, H_DATA_LEN, customer_buf[i].h_data);
}

}

//=====
//
// Function : LoadCustomerTable
//
//=====

```

```

void LoadCustomerTable(LOADER_TIME_STRUCT *customer_time_start)
{
    int i;

    long c_id;
    short c_d_id;
    long c_w_id;
    char c_first[FIRST_NAME_LEN+1];
    char c_middle[MIDDLE_NAME_LEN+1];
    char c_last[LAST_NAME_LEN+1];
    char c_street_1[ADDRESS_LEN+1];
    char c_street_2[ADDRESS_LEN+1];
    char c_city[ADDRESS_LEN+1];
    char c_state[STATE_LEN+1];
    char c_zip[ZIP_LEN+1];
    char c_phone[PHONE_LEN+1];
    char c_credit[CREDIT_LEN+1];
    double c_credit_lim;
    double c_discount;

    // fix to avoid ODBC float to numeric conversion problem.
    // double c_balance;
    char c_balance[6];

    double c_ytd_payment;
    short c_payment_cnt;
    short c_delivery_cnt;
    char c_data[C_DATA_LEN+1];
    char c_since[C_SINCE_LEN+1];
    RETCODE rc;

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 3);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0, FIRST_NAME_LEN, NULL, 0, 4);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0, MIDDLE_NAME_LEN, NULL, 0, 5);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN, NULL, 0, 6);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN, NULL, 0, 7);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN, NULL, 0, 8);
    if (rc != SUCCEED)

```

```

        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL, 0, 0, 9);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0, 0, 10);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0, 11);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL, 0, 0, 12);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0, C_SINCE_LEN, NULL, 0,
SQLCHARACTER, 13);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL, 0, 0, 14);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 15);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 16);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

// fix to avoid ODBC float to numeric conversion problem.

// rc = bcp_bind(c_hdbc1, (BYTE *) &c_balance, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 17);
// if (rc != SUCCEED)
//     HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0, SQLCHARACTER, 17);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 18);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 19);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 20);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

```

```

rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0, 500, NULL, 0, 0, 21);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

for (i = 0; i < customers_per_district; i++)
{
    c_id = customer_buf[i].c_id;
    c_d_id = customer_buf[i].c_d_id;
    c_w_id = customer_buf[i].c_w_id;

    strcpy(c_first, customer_buf[i].c_first);
    strcpy(c_middle, customer_buf[i].c_middle);
    strcpy(c_last, customer_buf[i].c_last);
    strcpy(c_street_1, customer_buf[i].c_street_1);
    strcpy(c_street_2, customer_buf[i].c_street_2);
    strcpy(c_city, customer_buf[i].c_city);
    strcpy(c_state, customer_buf[i].c_state);
    strcpy(c_zip, customer_buf[i].c_zip);
    strcpy(c_phone, customer_buf[i].c_phone);
    strcpy(c_credit, customer_buf[i].c_credit);

    FormatDate(&c_since);

    c_credit_lim = customer_buf[i].c_credit_lim;
    c_discount = customer_buf[i].c_discount;

    // fix to avoid ODBC float to numeric conversion problem.

    // c_balance = customer_buf[i].c_balance;
    strcpy(c_balance, customer_buf[i].c_balance);

    c_ytd_payment = customer_buf[i].c_ytd_payment;
    c_payment_cnt = customer_buf[i].c_payment_cnt;
    c_delivery_cnt = customer_buf[i].c_delivery_cnt;

    strcpy(c_data, customer_buf[i].c_data);

    // Send data to server
    rc = bcp_sendrow(c_hdbc1);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc1);

    customer_rows_loaded++;
    CheckForCommit(c_hdbc1, c_hstmt1, customer_rows_loaded,
"customer", &customer_time_start->time_start);
}

//=====
//
// Function : LoadHistoryTable
//
//=====

void LoadHistoryTable(LOADER_TIME_STRUCT *history_time_start)
{
    int i;
    long c_id;
    short c_d_id;

```

```

long      c_w_id;
double   h_amount;
char     h_data[H_DATA_LEN+1];
char     h_date[H_DATE_LEN+1];
RETCODE  rc;

rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
3);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
4);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
5);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

rc = bcp_bind(c_hdbc2, (BYTE *) &h_date, 0, H_DATE_LEN, NULL, 0,
SQLCHARACTER, 6);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8,
7);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0, H_DATA_LEN, NULL, 0, 0, 8);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

for (i = 0; i < customers_per_district; i++)
{
    c_id = customer_buf[i].c_id;
    c_d_id = customer_buf[i].c_d_id;
    c_w_id = customer_buf[i].c_w_id;
    h_amount = customer_buf[i].h_amount;
    strcpy(h_data, customer_buf[i].h_data);

    FormatDate(&h_date);

    // send to server
    rc = bcp_sendrow(c_hdbc2);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    history_rows_loaded++;
    CheckForCommit(c_hdbc2, c_hstmt2, history_rows_loaded,
"history", &history_time_start->time_start);
}

```

```

}

//=====
//
// Function   : LoadOrders
//
//=====

void LoadOrders()
{
    LOADER_TIME_STRUCT  orders_time_start;
    LOADER_TIME_STRUCT  new_order_time_start;
    LOADER_TIME_STRUCT  order_line_time_start;
    long                w_id;
    short               d_id;
    DWORD               dwThreadId[MAX_ORDER_THREADS];
    HANDLE              hThread[MAX_ORDER_THREADS];
    char                name[20];
    RETCODE             rc;
    char                bcphint[128];

    // seed with unique number
    seed(aptr->starting_warehouse + 5);

    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, "t_orders");

    LOGFILE_CREATE("\\orders.err")
    rc = bcp_init(o_hdbc1, name, NULL, LogFile, 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, CHECK CONSTRAINTS", (aptr->num_warehouses * 30000));
        rc = bcp_control(o_hdbc1, BCPHINTS, (void*) bcphint);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc1);
    }

    sprintf(name, "%s..%s", aptr->database, "t_new_order");

    LOGFILE_CREATE("\\neword.err")
    rc = bcp_init(o_hdbc2, name, NULL, LogFile, 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, CHECK_CONSTRAINTS", (aptr->num_warehouses * 9000));
        rc = bcp_control(o_hdbc2, BCPHINTS, (void*) bcphint);
        if (rc != SUCCEEDED)
            HandleErrorDBC(o_hdbc2);
    }

    sprintf(name, "%s..%s", aptr->database, "t_order_line");

    LOGFILE_CREATE("\\ordline.err")
    rc = bcp_init(o_hdbc3, name, NULL, LogFile, DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcphint, "tablock, order (ol_w_id, ol_d_id, ol_o_id,
ol_number), ROWS_PER_BATCH = %u, CHECK_CONSTRAINTS", (aptr->num_warehouses *
300000));
        rc = bcp_control(o_hdbc3, BCPHINTS, (void*) bcphint);
        if (rc != SUCCEEDED)
            HandleErrorDBC(o_hdbc3);
    }

    orders_rows_loaded = 0;
    new_order_rows_loaded = 0;
    order_line_rows_loaded = 0;

    OrdersBufInit();

    orders_time_start.time_start = (TimeNow() / MILLI);
    new_order_time_start.time_start = (TimeNow() / MILLI);
    order_line_time_start.time_start = (TimeNow() / MILLI);

    for (w_id = (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(int d_id, int w_id)
{
    int    cust[ORDERS_PER_DISTRICT+1];
    long   o_id;
    short  ol;

    printf("...Loading Order Buffer for: d_id = %d, w_id = %d\n",
        d_id, w_id);

    GetPermutation(cust, orders_per_district);

    for (o_id=0;o_id<orders_per_district;o_id++)
    {
        // Generate ORDER and NEW-ORDER data

        orders_buf[o_id].o_d_id = d_id;
        orders_buf[o_id].o_w_id = w_id;
        orders_buf[o_id].o_id = o_id+1;
        orders_buf[o_id].o_c_id = cust[o_id+1];
        orders_buf[o_id].o_ol_cnt = (short)RandomNumber(5L, 15L);

        if (o_id < first_new_order)
        {
            orders_buf[o_id].o_carrier_id =
(short)RandomNumber(1L, 10L);
            orders_buf[o_id].o_all_local = 1;
        }
        else
        {
            orders_buf[o_id].o_carrier_id = 0;
            orders_buf[o_id].o_all_local = 1;
        }

        for (ol=0; ol<orders_buf[o_id].o_ol_cnt; ol++)
        {
            orders_buf[o_id].o_ol[ol].ol = ol+1;
            orders_buf[o_id].o_ol[ol].ol_i_id = RandomNumber(1L,
max_items);
            orders_buf[o_id].o_ol[ol].ol_supply_w_id = w_id;
            orders_buf[o_id].o_ol[ol].ol_quantity = 5;
            MakeAlphaString(24, 24, OL_DIST_INFO_LEN,
&orders_buf[o_id].o_ol[ol].ol_dist_info);

            // Generate ORDER-LINE data
            if (o_id < first_new_order)
            {
                orders_buf[o_id].o_ol[ol].ol_amount = 0;
                // Added to insure ol_delivery_d set
                properly during load

                FormatDate(&orders_buf[o_id].o_ol[ol].ol_delivery_d);
            }
            else

```

```

        {
            orders_buf[o_id].o_ol[ol].ol_amount =
RandomNumber(1,999999)/100.0;
            // Added to insure ol_delivery_d set
properly during load

            // odbc datetime format
            strcpy(orders_buf[o_id].o_ol[ol].ol_delivery_d,"1899-12-31 00:00:00.000");
        }
    }
}

//=====
//
// Function   : LoadOrdersTable
//
//=====

void LoadOrdersTable(LOADER_TIME_STRUCT *orders_time_start)
{
    int         i;
    long        o_id;
    short       o_d_id;
    long        o_w_id;
    long        o_c_id;
    short       o_carrier_id;
    short       o_ol_cnt;
    short       o_all_local;
    char        o_entry_d[O_ENTRY_D_LEN+1];
    RETCODE     rc;
    DBINT       rcint;

    // bind ORDER data
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
3);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
4);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_entry_d, 0, O_ENTRY_D_LEN, NULL, 0,
SQLCHARACTER, 5);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 6);

```

```

        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
7);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 8);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    for (i = 0; i < orders_per_district; i++)
    {
        o_id         = orders_buf[i].o_id;
        o_d_id       = orders_buf[i].o_d_id;
        o_w_id       = orders_buf[i].o_w_id;
        o_c_id       = orders_buf[i].o_c_id;
        o_carrier_id = orders_buf[i].o_carrier_id;
        o_ol_cnt     = orders_buf[i].o_ol_cnt;
        o_all_local  = orders_buf[i].o_all_local;

        FormatDate(&o_entry_d);

        // send data to server
        rc = bcp_sendrow(o_hdbc1);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc1);

        orders_rows_loaded++;
        CheckForCommit(o_hdbc1, o_hstmt1, orders_rows_loaded, "orders",
&orders_time_start->time_start);
    }

    // rcint = bcp_batch(o_hdbc1);
    // if (rcint < 0)
    //     HandleErrorDBC(o_hdbc1);

    if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
    {
        rcint = bcp_done(o_hdbc1);
        if (rcint < 0)
            HandleErrorDBC(o_hdbc1);

        SQLFreeStmt(o_hstmt1, SQL_DROP);
        SQLDisconnect(o_hdbc1);
        SQLFreeConnect(o_hdbc1);

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
            BuildIndex("idxordcl");

        // build non-clustered index
        if (aptr->build_index == 1)
            BuildIndex("idxordnc");
    }
}

//=====
//

```

```

// Function : LoadNewOrderTable
//
//=====
void LoadNewOrderTable(LOADER_TIME_STRUCT *new_order_time_start)
{
    int            i;
    long           o_id;
    short         o_d_id;
    long          o_w_id;
    RETCODE       rc;
    DBINT         rcint;

    // Bind NEW-ORDER data

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
3);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

    for (i = first_new_order; i < last_new_order; i++)
    {
        o_id   = orders_buf[i].o_id;
        o_d_id = orders_buf[i].o_d_id;
        o_w_id = orders_buf[i].o_w_id;

        rc = bcp_sendrow(o_hdbc2);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc2);

        new_order_rows_loaded++;
        CheckForCommit(o_hdbc2, o_hstmt2, new_order_rows_loaded,
"new_order", &new_order_time_start->time_start);
    }

    // rcint = bcp_batch(o_hdbc2);
    // if (rcint < 0)
    //     HandleErrorDBC(o_hdbc2);

    if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
    {
        rcint = bcp_done(o_hdbc2);
        if (rcint < 0)
            HandleErrorDBC(o_hdbc2);

        SQLFreeStmt(o_hstmt2, SQL_DROP);
        SQLDisconnect(o_hdbc2);
        SQLFreeConnect(o_hdbc2);

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
            BuildIndex("idxnodcl");
    }
}

```

```

}

//=====
// Function : LoadOrderLineTable
//
//=====
void LoadOrderLineTable(LOADER_TIME_STRUCT *order_line_time_start)
{
    int            i,j;
    long           o_id;
    short         o_d_id;
    long          o_w_id;
    long          ol;
    long          ol_i_id;
    long          ol_supply_w_id;
    short         ol_quantity;
    double        ol_amount;
    char          ol_dist_info[DIST_INFO_LEN+1];
    char          ol_delivery_d[OL_DELIVERY_D_LEN+1];
    RETCODE       rc;
    DBINT         rcint;

    // bind ORDER-LINE data
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
3);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 4);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
5);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 6);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0, OL_DELIVERY_D_LEN,
NULL, 0, SQLCHARACTER, 7);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 8);
    if (rc != SUCCEED)

```



```

        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 9);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0, DIST_INFO_LEN, NULL, 0, 0, 10);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);

    for (i = 0; i < orders_per_district; i++)
    {
        o_id = orders_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 != SUCCEEDED)
                HandleErrorDBC(o_hdbc3);

            order_line_rows_loaded++;
            CheckForCommit(o_hdbc3, o_hstmt3,
order_line_rows_loaded, "order_line", &order_line_time_start->time_start);
        }

    }

    // rcint = bcp_batch(o_hdbc3);
    // if (rcint < 0)
    //     HandleErrorDBC(o_hdbc3);

    if ((o_w_id == 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;
    // DBINT rcint;

    if ( !(rows_loaded % aptr->batch) )
    {
        // rcint = bcp_batch(hdbc);
        // if (rcint < 0)
        //     HandleErrorDBC(hdbc);

        time_end = (TimeNow() / MILLI);
        time_diff = time_end - *time_start;

        printf("-> Loaded %ld rows into %s in %ld sec - Total = %d (%.2f
rps)\n",
                    aptr->batch,
                    table_name,
                    time_diff,
                    rows_loaded,
                    (float) aptr->batch / (time_diff ? time_diff
: 1L));

        *time_start = time_end;
    }
}

```

```

    return;
}

//=====
//
// Function   : OpenConnections
//
//=====

void OpenConnections()
{
    RETCODE      rc;

    char          szDriverString[300];
    char          szDriverStringOut[1024];
    SQLSMALLINT   cbDriverStringOut;

    SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv );

    SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0 );

    SQLAllocHandle(SQL_HANDLE_DBC, henv, &i_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv, &w_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv, &c_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv, &c_hdbc2);
    SQLAllocHandle(SQL_HANDLE_DBC, henv, &o_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv, &o_hdbc2);
    SQLAllocHandle(SQL_HANDLE_DBC, henv, &o_hdbc3);

    SQLSetConnectAttr(i_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(w_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(c_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(c_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );

    // Open connections to SQL Server

    // Connection 1

    sprintf( szDriverString, "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

                                aptr->server,
                                aptr->user,
                                aptr->password,
                                aptr->database );

    rc = SQLSetConnectOption (i_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

```

```

    rc = SQLDriverConnect ( i_hdbc1,
                                NULL,
                                (SQLCHAR*)&szDriverString[0] ,
                                SQL_NTS,
                                (SQLCHAR*)&szDriverStringOut[0],
                                sizeof(szDriverStringOut),
                                &cbDriverStringOut,
                                SQL_DRIVER_NOPROMPT );

    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    // Connection 2

    sprintf( szDriverString, "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

                                aptr->server,
                                aptr->user,
                                aptr->password,
                                aptr->database );

    rc = SQLSetConnectOption (w_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    rc = SQLDriverConnect ( w_hdbc1,
                                NULL,
                                (SQLCHAR*)&szDriverString[0] ,
                                SQL_NTS,
                                (SQLCHAR*)&szDriverStringOut[0],
                                sizeof(szDriverStringOut),
                                &cbDriverStringOut,
                                SQL_DRIVER_NOPROMPT );

    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    // Connection 3

    sprintf( szDriverString, "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

                                aptr->server,
                                aptr->user,
                                aptr->password,
                                aptr->database );

    rc = SQLSetConnectOption (c_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc1);

    rc = SQLDriverConnect ( c_hdbc1,
                                NULL,
                                (SQLCHAR*)&szDriverString[0] ,
                                SQL_NTS,
                                (SQLCHAR*)&szDriverStringOut[0],

```

```

sizeof(szDriverStringOut),
&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

// Connection 4

sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,
aptr->user,
aptr->password,
aptr->database );

rc = SQLSetConnectOption (c_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

rc = SQLDriverConnect ( c_hdbc2,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

// Connection 5

sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,
aptr->user,
aptr->password,
aptr->database );

rc = SQLSetConnectOption (o_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc1);

rc = SQLDriverConnect ( o_hdbc1,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

```

```

SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc1);

// Connection 6

sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,
aptr->user,
aptr->password,
aptr->database );

rc = SQLSetConnectOption (o_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc2);

rc = SQLDriverConnect ( o_hdbc2,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc2);

// Connection 7

sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

aptr->server,
aptr->user,
aptr->password,
aptr->database );

rc = SQLSetConnectOption (o_hdbc3, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc3);

rc = SQLDriverConnect ( o_hdbc3,

NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc3);

```

```

}

//=====
//
// Function name: BuildIndex
//
//=====

void BuildIndex(char          *index_script)
{
    char          cmd[256];

    printf("Starting index creation: %s\n",index_script);

    sprintf(cmd, "isql -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,
            LogBase,
            index_script);

    system(cmd);

    printf("Finished index creation: %s\n",index_script);
}

void HandleErrorDBC (SQLHDBC  hdbc1)
{
    SQLCHAR          SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
    SQLINTEGER  NativeError;
    SQLSMALLINT i, MsgLen;
    SQLRETURN  rc2;
    char          timebuf[128];
    char          datebuf[128];
    FILE          *fp1;

    i = 1;
    while (( rc2 = SQLGetDiagRec(SQL_HANDLE_DBC , hdbc1, i, SqlState ,
    &NativeError,
                                Msg, sizeof(Msg) , &MsgLen )) !=
    SQL_NO_DATA )
    {

        sprintf( szLastError , "%s" , Msg );

        _strtime(timebuf);
        _strdate(datebuf);

        printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

        LOGFILE_CREATE("\\tpccldr.err")
        fp1 = fopen(LogFile,"w");
        if (fp1 == NULL)
            printf("ERROR: Unable to open errorlog file.\n");
        else
        {
            fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf,
            szLastError);

            fclose(fp1);
        }
    }
}

```

```

}

    i++;
}

void HandleErrorSTMT (HSTMT  hstmt1)
{
    SQLCHAR          SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
    SQLINTEGER  NativeError;
    SQLSMALLINT i, MsgLen;
    SQLRETURN  rc2;
    char          timebuf[128];
    char          datebuf[128];
    FILE          *fp1;

    i = 1;
    while (( rc2 = SQLGetDiagRec(SQL_HANDLE_STMT , hstmt1, i, SqlState ,
    &NativeError,
                                Msg, sizeof(Msg) , &MsgLen )) !=
    SQL_NO_DATA )
    {

        sprintf( szLastError , "%s" , Msg );

        _strtime(timebuf);
        _strdate(datebuf);

        printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

        LOGFILE_CREATE("\\tpccldr.err")
        fp1 = fopen(LogFile,"w");
        if (fp1 == NULL)
            printf("ERROR: Unable to open errorlog file.\n");
        else
        {
            fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf,
            szLastError);

            fclose(fp1);
        }
    }
    i++;
}

void FormatDate ( char* szTimeCOutput )
{
    struct tm when;
    time_t now;

    time( &now );
    when = *localtime( &now );

    mktime( &when );

    // odbc datetime format
    strftime( szTimeCOutput , 30 , "%Y-%m-%d %H:%M:%S.000", &when );
}

```

```

        return;
    }

//=====
//
// Function   : CheckDataBase
//
//=====

void CheckDataBase()
{
    RETCODE      rc;

    char          szDriverString[300];
    char          szDriverStringOut[1024];
    char          TablesBitMap[9] = {"000000000"};
    int           i, ExitFlag;

    SQLSMALLINT  cbDriverStringOut;
    SQLCHAR      TabName[10];
    SQLINTEGER    TabNameInd, TabCount, TabCountInd;

    ExitFlag = 0;

    SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv );
    SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0 );
    SQLAllocHandle(SQL_HANDLE_DBC, henv , &v_hdbc);

    SQLSetConnectAttr(v_hdbc, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );

    // Open connection to SQL Server
    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                aptr->server,
                                aptr->user,
                                aptr->password,
                                aptr->database );

    rc = SQLSetConnectAttr( v_hdbc, SQL_ATTR_PACKET_SIZE, (SQLPOINTER)aptr-
>pack_size, SQL_IS_INTEGER );
    if (rc != SQL_SUCCESS)
        HandleErrorDBC(v_hdbc);

    rc = SQLDriverConnect ( v_hdbc,
                                NULL,
                                (SQLCHAR*)&szDriverString[0] ,
                                SQL_NTS,
                                (SQLCHAR*)&szDriverStringOut[0],
                                sizeof(szDriverStringOut),
                                &cbDriverStringOut,
                                SQL_DRIVER_NOPROMPT );

    // if the rc is SQL_ERROR, the the TPCC database probably does not exist

```

```

    if (rc == SQL_ERROR)
    {
        printf("The database TPCC does not appear to exist!\n");
        printf("\nCheck %s\\ directory for database creation errors.\n",
LogBase);

        // cleanup database connections and handles
        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
        SQLDisconnect(v_hdbc);
        SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

        // since there is not a database, exit back to SETUP.CMD
        exit(1);
    }

    if ( SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt) != SQL_SUCCESS )
        HandleErrorDBC(v_hdbc);

    if ( SQLBindCol(v_hstmt, 1, SQL_C_ULONG, &TabCount, 0, &TabCountInd) !=
SQL_SUCCESS )
        HandleErrorSTMT(v_hstmt);

    // count the number of user tables from sysobjects
    rc = SQLExecDirect(v_hstmt, "select count(*) from sysobjects where xtype =
'\U'", SQL_NTS);
    if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
        HandleErrorSTMT(v_hstmt);

    if ( SQLFetch(v_hstmt) != SQL_SUCCESS )
        HandleErrorSTMT(v_hstmt);

    // if the number of tables is less than 9, select all the user tables in
TPCC
    if (TabCount != 9)
    {
        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
        SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt);

        if ( SQLBindCol(v_hstmt, 1, SQL_C_CHAR, &TabName,
sizeof(TabName), &TabNameInd) != SQL_SUCCESS )
            HandleErrorSTMT(v_hstmt);

        // select the list of user tables into a result set
        rc = SQLExecDirect(v_hstmt, "select * from sysobjects where
xtype = '\U'", SQL_NTS);
        if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
            HandleErrorSTMT(v_hstmt);

        // go through the result set and set the bitmap for each found
table
        // set the bitmap to '1' if the table name is found
        while ((rc = SQLFetch(v_hstmt)) != SQL_NO_DATA)
        {
            switch( TabName[0] )
            {
                case 'w':
                    TablesBitMap[0] = '1';
                    break;
                case 'd':
                    TablesBitMap[1] = '1';
                    break;
            }
        }
    }

```

```

case 'c':
    TablesBitMap[2] = '1';
    break;
case 'h':
    TablesBitMap[3] = '1';
    break;
case 'n':
    TablesBitMap[4] = '1';
    break;
case 'o':
    if (TabName[5] = 's')
        TablesBitMap[5] = '1';
    if (TabName[5] = '_')
        TablesBitMap[6] = '1';
    break;
case 'i':
    TablesBitMap[7] = '1';
    break;
case 's':
    TablesBitMap[8] = '1';
    break;
}

// a '0' ExitFlag means do NOT exit the loader early, a '1'
means exit the loader early
ExitFlag = 0;

// iterate through the bitmap to display which table(s) is
actually missing
for (i = 0; i <= 8; i++)
{
    switch(i)
    {
    case 0:
        if (TablesBitMap[i] == '0')
        {
            printf("The Warehouse table is
missing or damaged.\n");
            ExitFlag = 1;
        }
        break;
    case 1:
        if (TablesBitMap[i] == '0')
        {
            printf("The District table is
missing or damaged.\n");
            ExitFlag = 1;
        }
        break;
    case 2:
        if (TablesBitMap[i] == '0')
        {
            printf("The Customer table is
missing or damaged.\n");
            ExitFlag = 1;
        }
        break;
    case 3:
        if (TablesBitMap[i] == '0')
        {
            printf("The History table is
missing or damaged.\n");

```

```

        ExitFlag = 1;
    }
    break;
case 4:
    if (TablesBitMap[i] == '0')
    {
        printf("The New_Order table is
missing or damaged.\n");
        ExitFlag = 1;
    }
    break;
case 5:
    if (TablesBitMap[i] == '0')
    {
        printf("The Orders table is
missing or damaged.\n");
        ExitFlag = 1;
    }
    break;
case 6:
    if (TablesBitMap[i] == '0')
    {
        printf("The Order_Line table is
missing or damaged.\n");
        ExitFlag = 1;
    }
    break;
case 7:
    if (TablesBitMap[i] == '0')
    {
        printf("The Item table is missing
or damaged.\n");
        ExitFlag = 1;
    }
    break;
case 8:
    if (TablesBitMap[i] == '0')
    {
        printf("The Stock table is missing
or damaged.\n");
        ExitFlag = 1;
    }
    break;
}

// if one or more tables are missing, display message and exit
the loader
if (ExitFlag = 1)
{
    printf("\nExiting TPC-C Loader!\n");
    printf("\nCheck %s\ directory for database\n",
LogBase);
    printf("or table creation errors.\n");

    // cleanup database connections and handles
    SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
    SQLDisconnect(v_hdbc);
    SQLFreeHandle(SQL_HANDLE_DEC, v_hdbc);

    exit(1);
}
}

```

```

// cleanup database connections and handles
SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
SQLDisconnect(v_hdbc);
SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

```

```
return;
```

time.c

```

// File: TIME.C
// Microsoft TPC-C Kit Ver. 4.30
// Copyright Microsoft, 1996, 1997, 1998,
1999,2000
// 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;
}

```

strings.c

```

// File: STRINGS.C
// Microsoft TPC-C Kit Ver. 4.30
// Copyright Microsoft, 1996, 1997, 1998, 1999,
2000
// Purpose: Source file for database loader string functions

// Includes
#include "tpcc.h"
#include <string.h>

```

```
#include <ctype.h>
```

```

//=====
// Function name: MakeAddress
//=====

```

```

void MakeAddress(char *street_1,
                 char *street_2,
                 char *city,
                 char *state,
                 char *zip)
{
#ifdef DEBUG
    printf("[%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[] =
"0123456789ABCDEFGHIJKLMNQPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
    static int chArrayMax = 61;

#ifdef DEBUG
    printf("[%ld]DBG: Entering MakeAlphaString()\n", (int) GetCurrentThreadId());
#endif

    len= RandomNumber(x, y);
    for (i=0; i<len; i++)
    {
        cc = chArray[RandomNumber(0, chArrayMax)];
        str[i] = cc;
    }
}

```

```

    }
    if ( len < z )
        memset(str+len, ' ', z - len);
    str[z] = 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 prcentage is valid
    if ((percent < 0) || (percent > 100))
    {
        printf("MakeOriginalAlphaString: Invalid percentage: %d\n",
            percent);
        exit(-1);
    }

    // verify string is at least 8 chars in length
    if ((x + y) <= 8)
    {
        printf("MakeOriginalAlphaString: string length must be >= 8\n");
        exit(-1);
    }

    // Make Alpha String
    len = MakeAlphaString(x,y, z, str);

    val = RandomNumber(1,100);
    if (val <= percent)
    {
        start = RandomNumber(0, len - 8);
        strncpy(str + start, "ORIGINAL", 8);
    }

#ifdef DEBUG
    printf("[%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;
}



---


random.c


---


// File: RANDOM.C
// Microsoft TPC-C Kit Ver. 4.30
// Copyright Microsoft, 1996, 1997, 1998, 1999,
2000
// Purpose: Random number generation routines for database loader

// Includes
#include "tpcc.h"
#include "math.h"

// Defines
#define A 16807
#define M 2147483647
#define Q 127773 /* M div A */
#define R 2836 /* M mod A */

```

```

#define Thread      __declspec(thread)

// Globals
long      Thread Seed = 0;      /* thread local seed */

/*****
 *
 * random -
 * Implements a GOOD pseudo random number generator. This generator
 * will/should? run the complete period before repeating.
 *
 * Copied from:
 * Random Numbers Generators: Good Ones Are Hard to Find.
 * Communications of the ACM - October 1988 Volume 31 Number 10
 *
 * Machine Dependencies:
 * long must be 2 ^ 31 - 1 or greater.
 *****/

/*****
 * seed - load the Seed value used in irand and drand. Should be used before
 * first call to irand or drand.
 *****/

void seed(long val)
{
#ifdef DEBUG
    printf("[%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
//Original 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;
}

```

```

}

```

Appendix C: Tunable Parameters

Microsoft SQL Server 2000 Installation Procedures

Microsoft SQL Server 2000 Installation
Procedures

Type of installation: custom

During the custom installation, use the default
settings for all except the following two areas:

Services accounts:

SQL Server - local system account

SQL Server Agent - local system account

Set the sort order/collation as:

binary sort order/Latin_1_General

Microsoft SQL Server 8.0 Startup Parameters

C:\SQL80\MSSQL\BINN\SQLSERVR

-eC:\MSSQL7\LOG\ERRORLOG -x -c -t3502
-g576

Where:

-c Start SQL Server independently of the
Windows NT Service Control Manager

-x Disables the keeping of CPU time and
cache-hit ratio statistics
-t3502 Prints a message to the SQL Server log
at the start and end of each checkpoint
-g576 Specify the amount of virtual address
space in MB, SQL Server will leave available
for memory allocations, excluding the buffer
pool and threads stack, such as dynamically-
loaded DLLs, extended procedure calls, etc.
Incorrect use of this option can lead to
conditions under which SQL Server may not
start or may encounter runtime errors.

Once SQL Server was booted the following
SQL script was run on each node:

```
dbcc traceon(-1)
go
dbcc traceon(9270)
go
```

This trace flag affects multinode locking and
will not be needed in the shipping product.

Boot.ini Parameters

8GB Tuning was enabled for Windows 2000
Advanced Server by setting the "/8GB /PAE"
switch in the boot.ini .

Compaq Device Drivers for Windows NT

The following Microsoft Windows NT device
drivers were replaced with Compaq-specific
device drivers:

- The Microsoft SMART-5300 Array
Controller default device driver
(CPQARRY2.SYS) was replaced with the
Compaq SMART-5300 Array Controller
Performance Drivers for Microsoft
Windows 2000 (cpqcissb.sys and
cpqcissd.sys).

Microsoft COM Component Configuration Parameters

The component services tool in Windows 2000
was used to change the settings for the four
TPCC components. The components are single
queues. All these components were set to
enable object pooling, object construction, just
in time activation, and component supports
events and statistics. The min and max pool
size for the components on each client are set
in the registry at 70 with 8 delivery queues.
The construction string was "Server =
myserver;UID=sa;pwd=;DATABASE=tpcc;"

Microsoft SQL Server 2000 Version

8.00.174 Configuration

1> 2> 3> DBCC execution completed. If DBCC printed error messages, contact your system administrator. Configuration option 'show advanced options' changed from 1 to 1. Run the RECONFIGURE statement to install.

```
sp_configure 'show advanced',1
1> 2> reconfigure with override
1> 2> sp_configure
```

name	config_value	run_value	minimum
affinity mask			0
2147483647	255	255	
allow updates			0
1	0	0	
awe enabled			0
1	1	1	
c2 audit mode			0
1	0	0	
cost threshold for parallelism			0
32767	5	5	
cursor threshold			-1
2147483647	-1	-1	
default full-text language			0
2147483647	1033	1033	
default language			0
9999	0	0	
fill factor (%)			0
100	0	0	
index create memory (KB)			704
2147483647	0	0	
lightweight pooling			0
1	0	0	
locks			5000
2147483647	0	0	
max degree of parallelism			0
32	0	0	
max server memory (MB)			4
2147483647	2147483647	2147483647	
max text repl size (B)			0
2147483647	65536	65536	
max worker threads			32
32767	400	400	
media retention			0
365	0	0	
min memory per query (KB)			512
2147483647	1024	1024	
min server memory (MB)			0
2147483647	0	0	
nested triggers			0
1	1	1	
network packet size (B)			512
65536	4096	4096	
open objects			0
2147483647	0	0	

```
priority boost 0
1 1 1
query governor cost limit 0
2147483647 0 0
query wait (s) -1 -1 -1
2147483647
recovery interval (min) 0
32767 40 40
remote access 0
1 1 1
remote login timeout (s) 0
2147483647 0 0
remote proc trans 0
1 0 0
remote query timeout (s) 0
2147483647 0 0
scan for startup procs 0
1 0 0
set working set size 0
1 0 0
show advanced options 0
1 1 1
two digit year cutoff 1753
9999 2049 2049
user connections 0
32767 0 0
user options 0
32767 0 0
```

1>

Benchcraft Profile

```
Profile: 12-nodesx1800
File Path: C:\BenchCraft\12-nodesx1800.pro
Version: 2
```

Number of Engines: 36

```
Name: FE01
Description:
Directory: c:\temp\FE01.log
Machine: N1
Parameter Set: 90%
Index: 0
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER185943500
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: FE02
Description:
Directory: c:\temp\FE02.log
Machine: N1
```

```
Parameter Set: 90%
Index: 50000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER286005718
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 2
```

```
Name: FE03
Description:
Directory: c:\temp\FE03.log
Machine: N1
Parameter Set: 90%
Index: 100000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER486111687
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 3
```

```
Name: FE04
Description:
Directory: c:\temp\FE04.log
Machine: N2
Parameter Set: 90%
Index: 150000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER61351046
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 1
```

```
Name: FE05
Description:
Directory: c:\temp\FE05.log
Machine: N2
Parameter Set: 90%
Index: 200000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER51445656
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 2
```

```
Name: FE06
Description:
Directory: c:\temp\FE06.log
```

Machine: N2
Parameter Set: 90%
Index: 250000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER61470359
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 3

Name: FE07
Description:
Directory: c:\temp\FE07.log
Machine: N3
Parameter Set: 90%
Index: 300000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER71530812
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: FE08
Description:
Directory: c:\temp\FE08.log
Machine: N3
Parameter Set: 90%
Index: 350000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER81559609
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: FE09
Description:
Directory: c:\temp\FE09.log
Machine: N3
Parameter Set: 90%
Index: 400000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER91581734
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 3

Name: FE10
Description:

Directory: c:\temp\FE10.log
Machine: N4
Parameter Set: 90%
Index: 450000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER101796593
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: FE11
Description:
Directory: c:\temp\FE11.log
Machine: N4
Parameter Set: 90%
Index: 500000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER111815125
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: FE12
Description:
Directory: c:\temp\FE12.log
Machine: N4
Parameter Set: 90%
Index: 550000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER121847578
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 3

Name: FE13
Description:
Directory: c:\temp\FE13.log
Machine: N5
Parameter Set: 90%
Index: 600000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER131948703
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: FE14

Description:
Directory: c:\temp\FE14.log
Machine: N5
Parameter Set: 90%
Index: 650000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER141987734
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: FE15
Description:
Directory: c:\temp\FE15.log
Machine: N5
Parameter Set: 90%
Index: 700000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER152008625
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 3

Name: FE16
Description:
Directory: c:\temp\FE16.log
Machine: N6
Parameter Set: 90%
Index: 750000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER162045765
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: FE17
Description:
Directory: c:\temp\FE17.log
Machine: N6
Parameter Set: 90%
Index: 800000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER172074875
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: FE18
Description:
Directory: c:\temp\fe18.log
Machine: N6
Parameter Set: 90%
Index: 850000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER182096734
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 3

Name: FE33
Description:
Directory: c:\temp\fe33.log
Machine: N11
Parameter Set: 90%
Index: 1600000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER194163625
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 3

Name: FE19
Description:
Directory: c:\temp\fe19.log
Machine: N7
Parameter Set: 90%
Index: 900000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER208165921
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: FE20
Description:
Directory: c:\temp\fe20.log
Machine: N7
Parameter Set: 90%
Index: 950000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER218309125
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: FE21
Description:
Directory: c:\temp\fe21.log
Machine: N7
Parameter Set: 90%
Index: 1000000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER228369562
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 3

Name: FE22
Description:
Directory: c:\temp\fe22.log
Machine: N8
Parameter Set: 90%
Index: 1050000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER238477187
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: FE23
Description:
Directory: c:\temp\fe23.log
Machine: N8
Parameter Set: 90%
Index: 1100000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER248583203
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: FE24
Description:
Directory: c:\temp\fe24.log
Machine: N8
Parameter Set: 90%
Index: 1150000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER258632218
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233

CPU: 3
Name: FE25
Description:
Directory: c:\temp\fe25.log
Machine: N9
Parameter Set: 90%
Index: 1200000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER268712812
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: FE26
Description:
Directory: c:\temp\fe26.log
Machine: N9
Parameter Set: 90%
Index: 1250000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER278790828
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: FE27
Description:
Directory: c:\temp\fe27.log
Machine: N9
Parameter Set: 90%
Index: 1300000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER288846796
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 3

Name: FE28
Description:
Directory: c:\temp\fe28.log
Machine: N10
Parameter Set: 90%
Index: 1350000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER298908937
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0

CLIENT_NURAND: 233
CPU: 1

Name: FE29
Description:
Directory: c:\temp\fe29.log
Machine: N10
Parameter Set: 90%
Index: 1400000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER308985281
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: FE30
Description:
Directory: c:\temp\fe30.log
Machine: N10
Parameter Set: 90%
Index: 1450000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER319079562
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 3

Name: FE31
Description:
Directory: c:\temp\fe31.log
Machine: N11
Parameter Set: 90%
Index: 1500000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER329129281
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: FE32
Description:
Directory: c:\temp\fe32.log
Machine: N11
Parameter Set: 90%
Index: 1550000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER339195046
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0

Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: FE34
Description:
Directory: c:\temp\fe34.log
Machine: N12
Parameter Set: 90%
Index: 1650000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER349230843
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 1

Name: FE35
Description:
Directory: c:\temp\fe35.log
Machine: N12
Parameter Set: 90%
Index: 1700000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER359267828
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 2

Name: FE36
Description:
Directory: c:\temp\fe36.log
Machine: N12
Parameter Set: 90%
Index: 1750000000
Seed: 18546
Configured Users: 6000
Pipe Name: DRIVER369315421
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 233
CPU: 3

Number of User groups: 36

Driver Engine: FE01
IIS Server: FE1C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1 - 600
w_id Min Warehouse: 1
w_id Max Warehouse: 21600

Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE02
IIS Server: FE2C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 601 - 1200
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE03
IIS Server: FE3C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1201 - 1800
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE04
IIS Server: FE4C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1801 - 2400
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE05
IIS Server: FE5C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2401 - 3000
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE06
IIS Server: FE6C

SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3001 - 3600
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE07
IIS Server: FE7C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3601 - 4200
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE08
IIS Server: FE8C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4201 - 4800
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE09
IIS Server: FE9C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4801 - 5400
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE10
IIS Server: FE10C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5401 - 6000
w_id Min Warehouse: 1
w_id Max Warehouse: 21600

Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE11
IIS Server: FE11C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 6001 - 6600
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE12
IIS Server: FE12C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 6601 - 7200
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE13
IIS Server: FE13C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 7201 - 7800
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE14
IIS Server: FE14C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 7801 - 8400
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE15
IIS Server: FE15C

SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 8401 - 9000
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE16
IIS Server: FE16C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 9001 - 9600
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE17
IIS Server: FE17C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 9601 - 10200
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE18
IIS Server: FE18C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 10201 - 10800
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE19
IIS Server: FE19C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 10801 - 11400
w_id Min Warehouse: 1
w_id Max Warehouse: 21600

Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE20
IIS Server: FE20C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 11401 - 12000
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE21
IIS Server: FE21C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 12001 - 12600
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE22
IIS Server: FE22C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 12601 - 13200
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE23
IIS Server: FE23C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 13201 - 13800
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE24
IIS Server: FE24C

SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 13801 - 14400
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE25
IIS Server: FE25C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 14401 - 15000
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE26
IIS Server: FE26C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 15001 - 15600
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE27
IIS Server: FE27C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 15601 - 16200
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE28
IIS Server: FE28C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 16201 - 16800
w_id Min Warehouse: 1
w_id Max Warehouse: 21600

Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE29
IIS Server: FE29C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 16801 - 17400
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE30
IIS Server: FE30C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 17401 - 18000
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE31
IIS Server: FE31C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 18001 - 18600
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE32
IIS Server: FE32C
SQL Server: bigfoot_ip
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 18601 - 19200
w_id Min Warehouse: 1
w_id Max Warehouse: 21600
Scale: Normal
User Count: 6000
District id: 1
Scale Down: No

Driver Engine: FE33
IIS Server: FE33C

SQL Server: bigfoot_ip
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 19201 - 19800
 w_id Min Warehouse: 1
 w_id Max Warehouse: 21600
 Scale: Normal
 User Count: 6000
 District id: 1
 Scale Down: No

Driver Engine: FE34
 IIS Server: FE34C
 SQL Server: bigfoot_ip
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 19801 - 20400
 w_id Min Warehouse: 1
 w_id Max Warehouse: 21600
 Scale: Normal
 User Count: 6000
 District id: 1
 Scale Down: No

Driver Engine: FE35
 IIS Server: FE35C
 SQL Server: bigfoot_ip
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 20401 - 21000
 w_id Min Warehouse: 1
 w_id Max Warehouse: 21600
 Scale: Normal
 User Count: 6000
 District id: 1
 Scale Down: No

Driver Engine: FE36
 IIS Server: FE36C
 SQL Server: bigfoot_ip
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 21001 - 21600
 w_id Min Warehouse: 1
 w_id Max Warehouse: 21600
 Scale: Normal
 User Count: 6000
 District id: 1
 Scale Down: No

Number of Parameter Sets: 40

~Default
 Default Parameter Set

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time

12.05	18.01		New Order	10.00	
			0.10	5.00	0.10
12.05	3.01		Payment	10.00	
			0.10	5.00	0.10
5.05	2.01		Delivery	1.00	
			0.10	5.00	0.10
5.05	2.01		Stock Level	1.00	
			0.10	20.00	0.10
10.05	2.01		Order Status	1.00	
			0.10	5.00	0.10

Tuned Distribution

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.05	18.01		New Order	44.75	
			0.10	5.00	0.10
12.05	3.01		Payment	43.10	
			0.10	5.00	0.10
5.05	2.01		Delivery	4.05	
			0.10	5.00	0.10
5.05	2.01		Stock Level	4.05	
			0.10	20.00	0.10
10.05	2.01		Order Status	4.05	
			0.10	5.00	0.10

No Think

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
0.00	0.00		New Order	10.00	
			0.00	5.00	0.00
0.00	0.00		Payment	10.00	
			0.00	5.00	0.00
0.00	0.00		Delivery	1.00	
			0.00	5.00	0.00
0.00	0.00		Stock Level	1.00	
			0.00	20.00	0.00
0.00	0.00		Order Status	1.00	
			0.00	5.00	0.00

95%

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
13.00	18.01		New Order	44.75	
			0.10	5.00	0.10
13.00	3.01		Payment	43.10	
			0.10	5.00	0.10
6.00	2.01		Delivery	4.05	
			0.10	5.00	0.10
6.00	2.01		Stock Level	4.05	
			0.10	20.00	0.10
11.00	2.01		Order Status	4.05	
			0.10	5.00	0.10

90%

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
16.00	18.01		New Order	44.75	
			0.10	5.00	0.10
16.00	3.01		Payment	43.10	
			0.10	5.00	0.10
9.00	2.01		Delivery	4.05	
			0.10	5.00	0.10
9.00	2.01		Stock Level	4.05	
			0.10	20.00	0.10
14.00	2.01		Order Status	4.05	
			0.10	5.00	0.10

3.0

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
36.15	0.00		New Order	44.75	
			0.10	5.00	0.10
36.15	0.00		Payment	43.10	
			0.10	5.00	0.10
15.15	0.00		Delivery	4.05	
			0.10	5.00	0.10
15.15	0.00		Stock Level	4.05	
			0.10	20.00	0.10
30.15	0.00		Order Status	4.05	
			0.10	5.00	0.10

4.0

4.0 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
48.20	18.01		New Order	44.75	
			0.10	5.00	0.10
48.20	3.01		Payment	43.10	
			0.10	5.00	0.10
20.20	2.01		Delivery	4.05	
			0.10	5.00	0.10
20.20	2.01		Stock Level	4.05	
			0.10	20.00	0.10
40.20	2.01		Order Status	4.05	
			0.10	5.00	0.10

3.8

3.8 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
45.70	18.01		New Order	44.75	
			0.10	5.00	0.10
45.70	3.01		Payment	43.10	
			0.10	5.00	0.10
19.10	2.01		Delivery	4.05	
			0.10	5.00	0.10

19.10	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
38.10	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			3.6			
			3.6 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
43.30	18.01		New Order	44.75		
			0.10	5.00	0.10	
43.30	3.01		Payment	43.10		
			0.10	5.00	0.10	
18.10	2.01		Delivery	4.05		
			0.10	5.00	0.10	
18.10	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
36.18	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			3.4			
			3.4 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
40.90	18.01		New Order	44.75		
			0.10	5.00	0.10	
40.90	3.01		Payment	43.10		
			0.10	5.00	0.10	
17.10	2.01		Delivery	4.05		
			0.10	5.00	0.10	
17.10	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
17.10	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			3.2			
			3.2 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
38.50	18.01		New Order	44.75		
			0.10	5.00	0.10	
38.50	3.01		Payment	43.10		
			0.10	5.00	0.10	
16.10	2.01		Delivery	4.05		
			0.10	5.00	0.10	
16.10	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
32.10	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			2.8			
			2.8 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	

33.74	18.01		New Order	44.75		
			0.10	5.00	0.10	
33.74	3.01		Payment	43.10		
			0.10	5.00	0.10	
14.14	2.01		Delivery	4.05		
			0.10	5.00	0.10	
14.14	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
28.14	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			2.6			
			2.6 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
31.30	18.01		New Order	44.75		
			0.10	5.00	0.10	
31.30	3.01		Payment	43.10		
			0.10	5.00	0.10	
13.10	2.01		Delivery	4.05		
			0.10	5.00	0.10	
13.10	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
26.10	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			2.4			
			2.4 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
28.90	18.01		New Order	44.75		
			0.10	5.00	0.10	
28.90	3.01		Payment	43.10		
			0.10	5.00	0.10	
12.10	2.01		Delivery	4.05		
			0.10	5.00	0.10	
12.10	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
24.10	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			2.2			
			2.2 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
28.90	18.01		New Order	44.75		
			0.10	5.00	0.10	
28.90	3.01		Payment	43.10		
			0.10	5.00	0.10	
12.10	2.01		Delivery	4.05		
			0.10	5.00	0.10	
12.10	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
24.12	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			2.0			

			2.0 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
24.10	18.01		New Order	44.75		
			0.10	5.00	0.10	
24.10	3.01		Payment	43.10		
			0.10	5.00	0.10	
10.10	2.01		Delivery	4.05		
			0.10	5.00	0.10	
10.10	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
20.10	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			5.0			
			5.0 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
60.25	18.01		New Order	44.75		
			0.10	5.00	0.10	
60.25	3.01		Payment	43.10		
			0.10	5.00	0.10	
25.25	2.01		Delivery	4.05		
			0.10	5.00	0.10	
25.25	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
50.25	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			4.5			
			4.5 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
54.20	18.01		New Order	44.75		
			0.10	5.00	0.10	
54.20	3.01		Payment	43.10		
			0.10	5.00	0.10	
22.70	2.01		Delivery	4.05		
			0.10	5.00	0.10	
22.70	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
45.20	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			3.5			
			3.5 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
42.10	18.01		New Order	44.75		
			0.10	5.00	0.10	
42.10	3.01		Payment	43.10		
			0.10	5.00	0.10	
17.60	2.01		Delivery	4.05		
			0.10	5.00	0.10	

17.60	2.01		Stock Level	4.05				
			0.10	20.00	0.10			
35.10	2.01		Order Status	4.05				
			0.10	5.00	0.10			
			1.8					
			1.8 tt					
Key	RT	RT	Menu	Txn	Think			
Time	Delay	Fence	Delay	Weight	Time			
21.60	18.01		New Order	44.75				
			0.10	5.00	0.10			
21.60	3.01		Payment	43.10				
			0.10	5.00	0.10			
9.09	2.01		Delivery	4.05				
			0.10	5.00	0.10			
9.09	2.01		Stock Level	4.05				
			0.10	20.00	0.10			
18.09	2.01		Order Status	4.05				
			0.10	5.00	0.10			
			4.2					
			4.2 tt					
Key	RT	RT	Menu	Txn	Think			
Time	Delay	Fence	Delay	Weight	Time			
54.20	18.01		New Order	44.75				
			0.10	5.00	0.10			
54.20	3.01		Payment	43.10				
			0.10	5.00	0.10			
22.70	2.01		Delivery	4.05				
			0.10	5.00	0.10			
22.70	2.01		Stock Level	4.05				
			0.10	20.00	0.10			
45.20	2.01		Order Status	4.05				
			0.10	5.00	0.10			
			1.6					
			1.6 tt					
Key	RT	RT	Menu	Txn	Think			
Time	Delay	Fence	Delay	Weight	Time			
19.20	18.01		New Order	44.75				
			0.10	5.00	0.10			
19.20	3.01		Payment	43.10				
			0.10	5.00	0.10			
8.08	2.01		Delivery	4.05				
			0.10	5.00	0.10			
8.08	2.01		Stock Level	4.05				
			0.10	20.00	0.10			
16.08	2.01		Order Status	4.05				
			0.10	5.00	0.10			
			1.4					
			1.4 tt					
Key	RT	RT	Menu	Txn	Think			
Time	Delay	Fence	Delay	Weight	Time			

16.87	18.01		New Order	44.75				
			0.10	5.00	0.10			
16.87	3.01		Payment	43.10				
			0.10	5.00	0.10			
7.07	2.01		Delivery	4.05				
			0.10	5.00	0.10			
7.07	2.01		Stock Level	4.05				
			0.10	20.00	0.10			
14.07	2.01		Order Status	4.05				
			0.10	5.00	0.10			
			1.2					
			1.2 tt					
Key	RT	RT	Menu	Txn	Think			
Time	Delay	Fence	Delay	Weight	Time			
14.46	18.01		New Order	44.75				
			0.10	5.00	0.10			
14.46	3.01		Payment	43.10				
			0.10	5.00	0.10			
6.06	2.01		Delivery	4.05				
			0.10	5.00	0.10			
6.06	2.01		Stock Level	4.05				
			0.10	20.00	0.10			
12.06	2.01		Order Status	4.05				
			0.10	5.00	0.10			
			3.5					
			3.5 tt					
Key	RT	RT	Menu	Txn	Think			
Time	Delay	Fence	Delay	Weight	Time			
42.10	18.01		New Order	44.75				
			0.10	5.00	0.10			
42.10	3.01		Payment	43.10				
			0.10	5.00	0.10			
17.60	2.01		Delivery	4.05				
			0.10	5.00	0.10			
17.60	2.01		Stock Level	4.05				
			0.10	20.00	0.10			
35.10	2.01		Order Status	4.05				
			0.10	5.00	0.10			
			1.9					
			1.9 tt					
Key	RT	RT	Menu	Txn	Think			
Time	Delay	Fence	Delay	Weight	Time			
22.89	18.01		New Order	44.75				
			0.10	5.00	0.10			
22.89	3.01		Payment	43.10				
			0.10	5.00	0.10			
9.59	2.01		Delivery	4.05				
			0.10	5.00	0.10			
9.59	2.01		Stock Level	4.05				
			0.10	20.00	0.10			
19.09	2.01		Order Status	4.05				
			0.10	5.00	0.10			
			1.1					

			1.1 tt					
Key	RT	RT	Menu	Txn	Think			
Time	Delay	Fence	Delay	Weight	Time			
13.25	18.01		New Order	44.75				
			0.10	5.00	0.10			
13.25	3.01		Payment	43.10				
			0.10	5.00	0.10			
5.55	2.01		Delivery	4.05				
			0.10	5.00	0.10			
5.55	2.01		Stock Level	4.05				
			0.10	20.00	0.10			
11.05	2.01		Order Status	4.05				
			0.10	5.00	0.10			
			1.05					
			1.05 tt					
Key	RT	RT	Menu	Txn	Think			
Time	Delay	Fence	Delay	Weight	Time			
12.65	18.01		New Order	44.75				
			0.10	5.00	0.10			
12.65	3.01		Payment	43.10				
			0.10	5.00	0.10			
5.30	2.01		Delivery	4.05				
			0.10	5.00	0.10			
5.30	2.01		Stock Level	4.05				
			0.10	20.00	0.10			
10.55	2.01		Order Status	4.05				
			0.10	5.00	0.10			
			1.3					
			1.3 tt					
Key	RT	RT	Menu	Txn	Think			
Time	Delay	Fence	Delay	Weight	Time			
15.60	18.01		New Order	44.75				
			0.10	5.00	0.10			
15.60	3.01		Payment	43.10				
			0.10	5.00	0.10			
6.50	2.01		Delivery	4.05				
			0.10	5.00	0.10			
6.50	2.01		Stock Level	4.05				
			0.10	20.00	0.10			
13.06	2.01		Order Status	4.05				
			0.10	5.00	0.10			
			1.09					
			1.09 tt					
Key	RT	RT	Menu	Txn	Think			
Time	Delay	Fence	Delay	Weight	Time			
13.13	18.01		New Order	44.75				
			0.10	5.00	0.10			
13.13	3.01		Payment	43.10				
			0.10	5.00	0.10			
5.50	2.01		Delivery	4.05				
			0.10	5.00	0.10			

5.50	2.01		Stock Level	4.05					
			0.10	20.00	0.10				
10.95	2.01		Order Status	4.05					
			0.10	5.00	0.10				
			1.08						
			1.08 tt						
Key	RT	RT	Menu	Txn	Think				
				Weight	Time				
Time	Delay	Fence	Delay						
13.01	18.01		New Order	44.75					
			0.10	5.00	0.10				
13.01	3.01		Payment	43.10					
			0.10	5.00	0.10				
5.45	2.01		Delivery	4.05					
			0.10	5.00	0.10				
5.45	2.01		Stock Level	4.05					
			0.10	20.00	0.10				
10.85	2.01		Order Status	4.05					
			0.10	5.00	0.10				
			1.07						
			1.07 tt						
Key	RT	RT	Menu	Txn	Think				
				Weight	Time				
Time	Delay	Fence	Delay						
12.89	18.01		New Order	44.75					
			0.10	5.00	0.10				
12.89	3.01		Payment	43.10					
			0.10	5.00	0.10				
5.40	2.01		Delivery	4.05					
			0.10	5.00	0.10				
5.40	2.01		Stock Level	4.05					
			0.10	20.00	0.10				
10.75	2.01		Order Status	4.05					
			0.10	5.00	0.10				
			1.06						
			1.06 tt						
Key	RT	RT	Menu	Txn	Think				
				Weight	Time				
Time	Delay	Fence	Delay						
12.77	18.01		New Order	44.75					
			0.10	5.00	0.10				
12.77	3.01		Payment	43.10					
			0.10	5.00	0.10				
5.35	2.01		Delivery	4.05					
			0.10	5.00	0.10				
5.35	2.01		Stock Level	4.05					
			0.10	20.00	0.10				
10.65	2.01		Order Status	4.05					
			0.10	5.00	0.10				
			1.15						
			1.15 tt						
Key	RT	RT	Menu	Txn	Think				
				Weight	Time				
Time	Delay	Fence	Delay						

13.85	18.01		New Order	44.75					
			0.10	5.00	0.10				
13.85	3.01		Payment	43.10					
			0.10	5.00	0.10				
5.80	2.01		Delivery	4.05					
			0.10	5.00	0.10				
5.80	2.01		Stock Level	4.05					
			0.10	20.00	0.10				
11.55	2.01		Order Status	4.05					
			0.10	5.00	0.10				
			1.25						
			1.25 tt						
Key	RT	RT	Menu	Txn	Think				
				Weight	Time				
Time	Delay	Fence	Delay						
15.06	18.01		New Order	44.75					
			0.10	5.00	0.10				
15.06	3.01		Payment	43.10					
			0.10	5.00	0.10				
6.31	2.01		Delivery	4.05					
			0.10	5.00	0.10				
6.31	2.01		Stock Level	4.05					
			0.10	20.00	0.10				
12.56	2.01		Order Status	4.05					
			0.10	5.00	0.10				
			1.3						
			1.3 tt						
Key	RT	RT	Menu	Txn	Think				
				Weight	Time				
Time	Delay	Fence	Delay						
15.66	18.01		New Order	44.75					
			0.10	5.00	0.10				
15.66	3.01		Payment	43.10					
			0.10	5.00	0.10				
6.56	2.01		Delivery	4.05					
			0.10	5.00	0.10				
6.56	2.01		Stock Level	4.05					
			0.10	20.00	0.10				
13.06	2.01		Order Status	4.05					
			0.10	5.00	0.10				
			1.12						
			1.12 tt						
Key	RT	RT	Menu	Txn	Think				
				Weight	Time				
Time	Delay	Fence	Delay						
13.49	18.01		New Order	44.75					
			0.10	5.00	0.10				
13.49	3.01		Payment	43.10					
			0.10	5.00	0.10				
5.65	2.01		Delivery	4.05					
			0.10	5.00	0.10				
5.65	2.01		Stock Level	4.05					
			0.10	20.00	0.10				
11.25	2.01		Order Status	4.05					
			0.10	5.00	0.10				
			1.18						

				1.18 tt					
Key	RT	RT	Menu	Txn	Think				
				Weight	Time				
Time	Delay	Fence	Delay						
14.21	18.01		New Order	44.75					
			0.10	5.00	0.10				
14.21	3.01		Payment	43.10					
			0.10	5.00	0.10				
5.95	2.01		Delivery	4.05					
			0.10	5.00	0.10				
5.95	2.01		Stock Level	4.05					
			0.10	20.00	0.10				
11.85	2.01		Order Status	4.05					
			0.10	5.00	0.10				
			1.22						
			1.22 tt						
Key	RT	RT	Menu	Txn	Think				
				Weight	Time				
Time	Delay	Fence	Delay						
14.70	18.01		New Order	44.75					
			0.10	5.00	0.10				
14.70	3.01		Payment	43.10					
			0.10	5.00	0.10				
6.16	2.01		Delivery	4.05					
			0.10	5.00	0.10				
6.16	2.01		Stock Level	4.05					
			0.10	20.00	0.10				
12.26	2.01		Order Status	4.05					
			0.10	5.00	0.10				
			1.28						
			1.28 tt						
Key	RT	RT	Menu	Txn	Think				
				Weight	Time				
Time	Delay	Fence	Delay						
15.42	18.01		New Order	44.75					
			0.10	5.00	0.10				
15.42	3.01		Payment	43.10					
			0.10	5.00	0.10				
6.46	2.01		Delivery	4.05					
			0.10	5.00	0.10				
6.46	2.01		Stock Level	4.05					
			0.10	20.00	0.10				
12.86	2.01		Order Status	4.05					
			0.10	5.00	0.10				

Internet Information Server Registry Parameters

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\InetInfo]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\InetInfo\Parameters]
"ListenBackLog"=dword:00002710
"DispatchEntries"=hex(7):4c,00,44,00,41,00,50,00,53,0
0,56,00,43,00,00,00,00,00
"PoolThreadLimit"=dword:00000258
"ThreadTimeout"=dword:00015180

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\InetInfo\Performance]
"Library"="inofctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000842
"Last Help"=dword:00000843
"First Counter"=dword:00000802
"First Help"=dword:00000803
"Library Validation
Code"=hex:a6,d3,a6,6e,3b,6f,bf,01,10,25,00,00,00,0,0
0,00
"WbemAdapFileTime"=hex:00,a8,84,48,24,af,bf,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000
```

World Wide Web Service Registry Parameters

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,00,3a,00,5c,00,57,00,49,00,4e,0
0,4e,00,54,00,5c,00,53,00,\
79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00
,6e,00,65,00,74,00,73,\
00,72,00,76,00,5c,00,69,00,69,00,6e,00,65,00,74,00,69,00,6e
,00,66,00,6f,00,2e,00,\
65,00,78,00,65,00,00,00
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):49,00,49,00,53,00,41,00,44,0
0,4d,00,49,00,4e,00,00,00,\
00,00
"DependOnGroup"=hex(7):00,00
"ObjectName"="LocalSystem"
```

```
"Description"="Provides Web connectivity and
administration through the Internet Information
Services snap-in."
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\ASP]
"NOTE"="This is for backward compatibility only."
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\ASP\Parameters]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\Parameters]
"MajorVersion"=dword:00000005
"MinorVersion"=dword:00000000
"InstallPath"="C:\\WINNT\\System32\\inetsrv"
"CertMapList"="C:\\WINNT\\System32\\inetsrv\\iiscremap
.dll"
"AccessDeniedMessage"="Error: Access is Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\\WINNT\\System32\\LogFiles"
"AcceptExOutstanding"=dword:00000028
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\Parameters\ADCLaunch]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\Parameters\ADCLaunch\RDSSErver.DataFactory]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\Parameters\Script Map]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\Parameters\Virtual Roots]
"/"="c:\\inetpub\\wwwroot,,207"
"/Scripts"="c:\\inetpub\\scripts,,204"
"/IISHelp"="c:\\winnt\\help\\iishelp,,201"
"/IISAdmin"="C:\\WINNT\\System32\\inetsrv\\iisadmin,,
201"
"/IISSamples"="c:\\inetpub\\iissamples,,201"
"/MSADC"="c:\\program files\\common
files\\system\\msadc,,205"
"/Printers"="C:\\WINNT\\web\\printers,,201"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\Performance]
"Library"="w3ctr.s.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"Last Counter"=dword:000008e6
"Last Help"=dword:000008e7
"First Counter"=dword:00000844
"First Help"=dword:00000845
"Library Validation
Code"=hex:3c,9a,d8,71,3b,6f,bf,01,10,3d,00,00,00,0,0
0,00
"WbemAdapFileTime"=hex:00,a5,d6,1b,a2,a5,bf,01
"WbemAdapFileSize"=dword:00003d10
"WbemAdapStatus"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14
,00,00,00,30,00,00,00,02,\
00,1c,00,01,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\
00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00
,01,01,00,00,00,00,00,\
05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01
,02,00,00,00,00,00,05,\
20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01
,02,00,01,01,00,00,00,\
00,00,05,0b,00,00,00,20,02,00,00,00,1c,00,fd,01,02
,00,01,02,00,00,00,00,\
00,05,20,00,00,00,23,02,00,00,72,00,73,00,01,01,00,00
,00,00,00,05,12,00,00,\
00,01,01,00,00,00,00,00,05,12,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\Enum]
"0"="Root\\LEGACY_W3SVC\\00000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

Server Registry Parameters

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cpqcissb\Parameters]
"CompletionMode"=dword:00000002
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cpqcissb\Parameters\Controller2]
"CompletionMode"=dword:00000001
```

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cpqcissd]
"Type"=dword:00000001
"Start"=dword:00000000
"ErrorControl"=dword:00000001
"Tag"=dword:00000102
"ImagePath"=hex(2):53,00,79,00,73,00,74,00,65,00,6d,0
0,33,00,32,00,5c,00,44,00,\
```

```
52,00,49,00,56,00,45,00,52,00,53,00,5c,00,63,00,70,00
,71,00,63,00,69,00,73,\
00,73,00,64,00,2e,00,73,00,79,00,73,00,00,00
"DisplayName"="Compaq CISS Controllers Disk Driver"
"Group"="Primary Disk"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissd\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
```

```
00,1c,00,01,01,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00,00,\
```

```
00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
```

```
05,12,00,00,00,50,00,5f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
```

```
20,00,00,00,20,02,00,00,32,00,65,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
```

```
00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
```

```
00,05,20,00,00,00,23,02,00,00,32,00,65,00,01,01,00,00,00,00,05,12,00,00,\
00,01,01,00,00,00,00,00,05,12,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissd\Enum]
```

```
"0"="CPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME&REV_1.10\4&1e595b43&0&0000000400000000"
```

```
"Count"=dword:0000000e
"NextInstance"=dword:0000000e
```

```
"1"="CPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME&REV_1.10\4&1e595b43&0&0100004000000000"
```

```
"2"="CPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME&REV_1.10\4&1e595b43&0&0200004000000000"
```

```
"3"="CPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME&REV_1.10\4&3aeb2299&0&0000000400000000"
```

```
"4"="CPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME&REV_1.10\4&3aeb2299&0&0100004000000000"
```

```
"5"="CPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME&REV_1.10\4&3aeb2299&0&0200004000000000"
```

```
"6"="CPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME&REV_1.10\4&1f479888&0&0000000400000000"
```

```
"7"="CPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME&REV_1.10\4&1f479888&0&0100004000000000"
```

```
"8"="CPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME&REV_1.10\4&102d450b&0&0000000400000000"
```

```
"9"="CPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME&REV_1.10\4&102d450b&0&0100004000000000"
```

```
"10"="CPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME&REV_1.10\4&102d450b&0&0200004000000000"
```

```
"11"="CPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME&REV_1.10\4&2f3647bc&0&0000000400000000"
```

```
"12"="CPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME&REV_1.10\4&2f3647bc&0&0100004000000000"
```

```
"13"="CPQCISS\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME&REV_1.10\4&2f3647bc&0&0200004000000000"
```

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SnetNic0\Parameters]
```

```
"maxCQ"=dword:00000ffc
"maxCQEntries"=dword:000003ff
"maxRegisterRegions"=dword:00001000
"maxVi"=dword:00000ffb
"maxSegmentsPerDesc"=dword:00000006
"maxDescriptorsPerQueue"=dword:0000001e
"cntEnable"=dword:00000000
```

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client]
"SharedMemoryOn"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\ConnectTo]
"DSQUERY"="DBNETLIB"
"APR_IP"="DBMSSOCN,130.170.1.4,1433"
"AUG_IP"="DBMSSOCN,130.170.1.8,1433"
"DEC_IP"="DBMSSOCN,130.170.1.12,1433"
"FEB_IP"="DBMSSOCN,130.170.1.2,1433"
"JAN_IP"="DBMSSOCN,130.170.1.1,1433"
"JUL_IP"="DBMSSOCN,130.170.1.7,1433"
"JUN_IP"="DBMSSOCN,130.170.1.6,1433"
"MAR_IP"="DBMSSOCN,130.170.1.3,1433"
"MAY_IP"="DBMSSOCN,130.170.1.5,1433"
"NOV_IP"="DBMSSOCN,130.170.1.11,1433"
"OCT_IP"="DBMSSOCN,130.170.1.10,1433"
"SEP_IP"="DBMSSOCN,130.170.1.9,1433"
"JAN"="DBNETLIB, via: JAN, 1433, 0"
"FEB"="DBNETLIB, via: FEB, 1433, 0"
"MAR"="DBNETLIB, via: MAR, 1433, 0"
"APR"="DBNETLIB, via: APR, 1433, 0"
"MAY"="DBNETLIB, via: MAY, 1433, 0"
"JUN"="DBNETLIB, via: JUN, 1433, 0"
"JUL"="DBNETLIB, via: JUL, 1433, 0"
"AUG"="DBNETLIB, via: AUG, 1433, 0"
"SEP"="DBNETLIB, via: SEP, 1433, 0"
"OCT"="DBNETLIB, via: OCT, 1433, 0"
"NOV"="DBNETLIB, via: NOV, 1433, 0"
"DEC"="DBNETLIB, via: DEC, 1433, 0"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\DB-Lib]
"AutoAnsiToOem"="ON"
"UseIntlSettings"="ON"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib]
"ProtocolOrder"=hex(7):74,00,63,00,70,00,00,00,6e,00,70,00,00,00,00
"Encrypt"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib>LastConnect]
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib\Np]
"DefaultPipe"="sql\query"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib\Tcp]
"DefaultPort"=dword:00000599
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib\Via]
"Vendor"="ServerNet II"
"DefaultServerPort"="0:1433"
"DefaultClientNIC"="0"
"RecognizedVendors"="Giganet, ServerNet II"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\TDS]
"JAN"="7.0"
"@="7.0"
"FEB"="7.0"
"(local)"="7.0"
"jan_ip"="7.0"
```

TPCC Application Registry Parameters

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPCC]
"Path"="c:\inetpub\wwwroot\"
"NumberOfDeliveryThreads"=dword:00000008
"MaxConnections"=dword:00002710
"MaxPendingDeliveries"=dword:000003e8
"DB_Protocol"="ODBC"
"TxnMonitor"="COM"
"DbServer"="JAN"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"=""
"COM_SinglePool"="YES"
```

Client Systems Configuration

```
Date . . . . . 07/18/2000
Time . . . . . 12:06:46

Product . . . . . ProLiant 1850R

Machine ID
From System Board . . . . . CPQ0679

Processor . . . . . Pentium III (R) at
550 MHz
Slot . . . . . 2
Secondary Cache . . . . . 512K
CPU ID . . . . . 0673
```



```

Processor . . . . . Pentium III(R) at
550 MHz
  Slot . . . . . 1
  Secondary Cache . . . . . 512K
  CPU ID . . . . . 0673

Processor(s) Mapped Out . . . . . None

Numeric Coprocessor . . . . . Integrated 387-
Compatible

Expansion Bus . . . . . ISA, PCI

System Identification Number . . . . . D827BZQ30041

CPU Mode . . . . . Real Mode

System ROM
  Revision . . . . . 07/14/1999
  Family . . . . . P7
  Flashable . . . . . Yes
  Supports P10 partition . . . . . Yes

Video Controller ROM
  Revision . . . . . 3.95

Option ROMs
  Address Range . . . . . C0000 - C7FFF
  Data Dump . . . . . (1998/08/26 10:10)

  Address Range . . . . . E8000 - EDFFF
  Data Dump . . . . . ( CPQSCSI d)

Bootblock ROM . . . . . 05/18/1998

Standby Recovery Server
  Status . . . . . Disabled
  COM Port . . . . . COM1
  Server Configuration . . . . . Recovery
  Timeout Value . . . . . 1 minutes

Memory Boards Identified:
  System Board
    DIMM Slot 1 . . . . . 0 Megabytes
    DIMM Slot 2 (SDRAM) . . . . . 128 Megabytes
    DIMM Slot 3 (SDRAM) . . . . . 128 Megabytes
    DIMM Slot 4 (SDRAM) . . . . . 256 Megabytes
  Total Compaq Memory . . . . . 512 Megabytes

Keyboard . . . . . Enhanced

LPT Ports . . . . . LPT1 (Address 3BC)

COM Ports . . . . . COM1 (Address 3F8)
  . . . . . COM2 (Address 2F8)

Netelligent 10/100 TX Embedded UTP Controller
  Device Type . . . . . Other Network
  Controller
    PCI Bus Number . . . . . 0
    Device Number . . . . . 7
    Function Number . . . . . 00h
    Slot Number . . . . . 0

```

```

Vendor ID . . . . . 0E11h
Device ID . . . . . AE43h
Revision ID . . . . . 10h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFFF0000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
IO Address Base . . . . . 2C20h
IO Address Length . . . . . 10h
Memory Address Base . . . . . C6FCFC0h
Memory Address Length . . . . . 10h

Compaq NC3120 Fast Ethernet NIC
  Device Type . . . . . Ethernet Controller
  PCI Bus Number . . . . . 0
  Device Number . . . . . 15
  Function Number . . . . . 00h
  Slot Number . . . . . 2
  Vendor ID . . . . . 0E11h
  Device ID . . . . . 1229h
  Subsystem Vendor ID . . . . . 0E11h
  Subsystem ID . . . . . B01Eh
  Revision ID . . . . . 05h
  Programming Interface . . . . . 00h
  Expansion ROM Base Address . . . . . FFF00000h
  IRQ Line . . . . . 11
  IRQ Pin . . . . . INTA#
  Memory Address Base . . . . . C4FFF000h
  Memory Address Length . . . . . 1000h
  IO Address Base . . . . . 2C00h
  IO Address Length . . . . . 20h
  Memory Address Base . . . . . C6D00000h
  Memory Address Length . . . . . 100000h

Compaq NC3122 Fast Ethernet NIC
  Device Type . . . . . Ethernet Controller
  PCI Bus Number . . . . . 1
  Device Number . . . . . 4
  Function Number . . . . . 00h
  Slot Number . . . . . 1
  Vendor ID . . . . . 0E11h
  Device ID . . . . . 1229h
  Subsystem Vendor ID . . . . . 0E11h
  Subsystem ID . . . . . B01Fh
  Revision ID . . . . . 05h
  Programming Interface . . . . . 00h
  Expansion ROM Base Address . . . . . 0h
  IRQ Line . . . . . 15
  IRQ Pin . . . . . INTA#
  Memory Address Base . . . . . C6BFF000h
  Memory Address Length . . . . . 1000h
  IO Address Base . . . . . 3000h
  IO Address Length . . . . . 20h
  Memory Address Base . . . . . C6F00000h
  Memory Address Length . . . . . 100000h

Compaq NC3122 Fast Ethernet NIC
  Device Type . . . . . Ethernet Controller
  PCI Bus Number . . . . . 1
  Device Number . . . . . 5

```

```

Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B01Fh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . C6BFE000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 3020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . C6E00000h
Memory Address Length . . . . . 100000h

Diskette Drive A . . . . . 1.44 Megabyte (3.5
inch)

Graphics Mode . . . . . 03 (80-Column Text)

Primary Monitor attached to . . . . . ATI RAGE IIC PCI
Graphics Controller
with Video Graphics Color Monitor

Base Memory
  System Total . . . . . 640 Kbytes
  Amount Free . . . . . 597 Kbytes
(612144 Bytes)

Extended Memory
  System Total . . . . . 523264 Kbytes

Expanded Memory
  LIM Driver Support . . . . . LIM driver not
loaded

Operating System . . . . . MS-DOS version 7.00
(from diskette)

Environment variables
  PATH=
  PROMPT=$PSG
  COMSPEC=A:\COMMAND.COM
  CMDLINE=inspect /u
End of environment

Revisions Table

Previous Revisions

Current Revisions
  I/O Board Revision . . . . . 08
  Assembly Version . . . . . 1
  Functional Revision Level . . . . . H
Date . . . . . 07/18/2000
Time . . . . . 12:18:28

```

```

Product . . . . . ProLiant DL380
Machine ID
  From System Board . . . . . CPQ0692
Processor . . . . . Pentium III(R) at
733 MHz
  Slot . . . . . 2
  Secondary Cache . . . . . 256K
  CPU ID . . . . . 0681
Processor . . . . . Pentium III(R) at
733 MHz
  Slot . . . . . 1
  Secondary Cache . . . . . 256K
  CPU ID . . . . . 0681
Processor(s) Mapped Out . . . . . None
Numeric Coprocessor . . . . . Integrated 387-
Compatible
Expansion Bus . . . . . ISA, PCI
System Identification Number . . . . . D005DKN1K117
CPU Mode . . . . . Real Mode
System ROM
  Revision . . . . . 12/05/1999
  Family . . . . . P17
  Flashable . . . . . Yes
  Supports F10 partition . . . . . Yes
Video Controller ROM
  Revision . . . . . 3.96
Option ROMs
  Address Range . . . . . C0000 - C7FFF
  Data Dump . . . . . (1999/03/24 23:56)
  Address Range . . . . . C8000 - CBFFF
  Data Dump . . . . . (04/22/98 ROC
ROC-ICR Option ROM/BIOS (C)Copyri...)
  Address Range . . . . . E8000 - EDFFF
  Data Dump . . . . . ( CPQSCSI d)
Bootblock ROM . . . . . 12/05/1999
Standby Recovery Server
  Status . . . . . Disabled
  COM Port . . . . . COM1
  Server Configuration . . . . . Recovery
  Timeout Value . . . . . 1 minutes
Memory Boards Identified:
System Board
  DIMM Slot 1 (SDRAM) . . . . . 256 Megabytes
  DIMM Slot 2 (SDRAM) . . . . . 256 Megabytes
  DIMM Slot 3 . . . . . 0 Megabytes

```

```

DIMM Slot 4 . . . . . 0 Megabytes
Total Compaq Memory . . . . . 512 Megabytes
Keyboard . . . . . Standard 11-Bit
LPT Ports . . . . . LPT1 (Address 3BC)
COM Ports . . . . . COM1 (Address 3F8)
COM2 (Address 2F8)
Compaq NC3163 Fast Ethernet NIC
  Device Type . . . . . Ethernet Controller
  PCI Bus Number . . . . . 0
  Device Number . . . . . 2
  Function Number . . . . . 00h
  Slot Number . . . . . 0
  Vendor ID . . . . . 0E11h
  Device ID . . . . . 1229h
  Subsystem Vendor ID . . . . . 0E11h
  Subsystem ID . . . . . B134h
  Revision ID . . . . . 08h
  Programming Interface . . . . . 00h
  Expansion ROM Base Address . . . . . FFF00000h
  IRQ Line . . . . . 11
  IRQ Pin . . . . . INTA#
  Memory Address Base . . . . . C3FFF000h
  Memory Address Length . . . . . 1000h
  IO Address Base . . . . . 2400h
  IO Address Length . . . . . 40h
  Memory Address Base . . . . . C3E00000h
  Memory Address Length . . . . . 100000h
Compaq NC3131 Fast Ethernet NIC
  Device Type . . . . . Ethernet Controller
  PCI Bus Number . . . . . 4
  Device Number . . . . . 4
  Function Number . . . . . 00h
  Slot Number . . . . . 3
  Vendor ID . . . . . 0E11h
  Device ID . . . . . 1229h
  Subsystem Vendor ID . . . . . 0E11h
  Subsystem ID . . . . . B0DDh
  Revision ID . . . . . 05h
  Programming Interface . . . . . 00h
  Expansion ROM Base Address . . . . . 0h
  IRQ Line . . . . . 5
  IRQ Pin . . . . . INTA#
  Memory Address Base . . . . . C6DFF000h
  Memory Address Length . . . . . 1000h
  IO Address Base . . . . . 3000h
  IO Address Length . . . . . 20h
  Memory Address Base . . . . . C6F00000h
  Memory Address Length . . . . . 100000h
Compaq NC3131 Fast Ethernet NIC
  Device Type . . . . . Ethernet Controller
  PCI Bus Number . . . . . 4
  Device Number . . . . . 5
  Function Number . . . . . 00h
  Slot Number . . . . . 3
  Vendor ID . . . . . 0E11h
  Device ID . . . . . 1229h

```

```

Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . C6DFE000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 3020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . C6E00000h
Memory Address Length . . . . . 100000h
Diskette Drive A . . . . . 1.44 Megabyte (3.5
inch)
Drive Controller 1, Compaq Integrated Smart Array
Controller
  IDA Firmware Revision . . . . . 1.34
  Array Accelerator Memory . . . . . 8188 Kbytes
  Accelerator Status . . . . . Not Configured
  Battery count . . . . . 0
  Batteries charged . . . . . 0
  Batteries failed . . . . . 0
  Internal ProLiant . . . . . Bus 2, Rev. JB21
Logical Drive 1 . . . . . 9095 Megabyte
Fault Tolerance . . . . . Mirroring
OS Format . . . . . Multi-Sector
Distribution
  Drive geometry (Cyl, Hds, Sec) 2177, 255, 32
  Array Accelerator . . . . . Disabled
  Logical drive in interim recovery mode.
Hard Drive 1
  SCSI Bus . . . . . 2
  SCSI ID . . . . . 0
  Serial Number . . . . . B3134145
  Firmware Revision 1 . . . . . B016
  Model Number . . . . . COMPAQ BD009122C6
  Initialized for Monitoring . . . . . Yes
  Reference time . . . . . 272806
  Sectors read . . . . . *732081181
  Hard read errors . . . . . 0
  Read errors retry . . . . . 0
  ECC read errors . . . . . 0
  Sectors written . . . . . 214083679
  Hard write errors . . . . . 0
  Write errors retry . . . . . 0
  Seek count . . . . . 866976
  Seek errors . . . . . 0
  Spin cycles . . . . . 4
  Spin up time . . . . . 0
  Seek time track . . . . . 36%
  Seek time third . . . . . 73%
  Seek time full . . . . . 73%
  Reallocated sectors . . . . . 30
  Recovers read failed . . . . . 0
  Bus faults . . . . . 0

```

```

Hard Drive 2
  SCSI Bus . . . . . 2
  SCSI ID . . . . . 1
  Serial Number . . . . . Undetermined
  Model Number . . . . . Undetermined
  Initialized for Monitoring . No

Graphics Mode . . . . . 03 (80-Column Text)

Primary Monitor attached to . . ATI RAGE IIC PCI
Graphics Controller
with Video Graphics Color Monitor

Base Memory
  System Total . . . . . 638 Kbytes
  Amount Free . . . . . 596 Kbytes
(611136 Bytes)

Extended Memory
  System Total . . . . . 523264 Kbytes

Expanded Memory
  LIM Driver Support . . . . . LIM driver not
loaded

Operating System . . . . . MS-DOS version 7.00
(from diskette)

Environment variables
  PATH=
  PROMPT=$PSG
  COMSPEC=A:\COMMAND.COM
  CMDLINE=inspect /u
End of environment

Revisions Table

Previous Revisions

Current Revisions

System serial number . . . . . D005DKN1K117
Date . . . . . 07/18/2000
Time . . . . . 11:41:56

Product . . . . . ProLiant 1600

Machine ID
  From System Board . . . . . CPQ0689

Processor . . . . . Pentium III(R) at
600 MHz
  Slot . . . . . 2
  Secondary Cache . . . . . 512K
  CPU ID . . . . . 0673

Processor . . . . . Pentium III(R) at
600 MHz
  Slot . . . . . 1
  Secondary Cache . . . . . 512K
  CPU ID . . . . . 0673

```

```

Processor(s) Mapped Out . . . . . None

Numeric Coprocessor . . . . . Integrated 387-
Compatible

Expansion Bus . . . . . ISA, PCI

System Identification Number . . D917CNK10304

CPU Mode . . . . . Real Mode

System ROM
  Revision . . . . . 12/07/1999
  Family . . . . . P8
  Flashable . . . . . Yes
  Supports F10 partition . . . . . Yes

Video Controller ROM
  Revision . . . . . 1.31 (Cirrus)

Option ROMs
  Address Range . . . . . C0000 - C7FFF
  Data Dump . . . . . (CL-GD5446 PCI VGA
BIOS Version 1.31 Copyright...)

  Address Range . . . . . E8000 - EDFFF
  Data Dump . . . . . ( CPQSCSI d)

Bootblock ROM . . . . . 04/07/1998

Standby Recovery Server
  Status . . . . . Disabled
  COM Port . . . . . COM1
  Server Configuration . . . . . Recovery
  Timeout Value . . . . . 1 minutes

Memory Boards Identified:
System Board
  DIMM Slot 1 (SDRAM) . . . . . 128 Megabytes
  DIMM Slot 2 (SDRAM) . . . . . 128 Megabytes
  DIMM Slot 3 (SDRAM) . . . . . 128 Megabytes
  DIMM Slot 4 (SDRAM) . . . . . 128 Megabytes
Total Compaq Memory . . . . . 512 Megabytes

Keyboard . . . . . Enhanced

LPT Ports . . . . . LPT1 (Address 3BC)

COM Ports . . . . . COM1 (Address 3F8)
COM2 (Address 2F8)

Netelligent 10/100 TX Embedded UTP Controller
  Device Type . . . . . Other Network
Controller
  PCI Bus Number . . . . . 1
  Device Number . . . . . 7
  Function Number . . . . . 00h
  Slot Number . . . . . 0
  Vendor ID . . . . . 0E11h
  Device ID . . . . . AE43h
  Revision ID . . . . . 10h
  Programming Interface . . . . . 00h
  Expansion ROM Base Address . . . . . FFFF0000h
  IRQ Line . . . . . 5

```

```

IRQ Pin . . . . . INTA#
IO Address Base . . . . . 2800h
IO Address Length . . . . . 10h
Memory Address Base . . . . . C6FFDDF0h
Memory Address Length . . . . . 10h

Netelligent 10/100TX PCI UTP Controller
  Device Type . . . . . Other Network
Controller
  PCI Bus Number . . . . . 1
  Device Number . . . . . 10
  Function Number . . . . . 00h
  Slot Number . . . . . 0
  Vendor ID . . . . . 0E11h
  Device ID . . . . . AE32h
  Revision ID . . . . . 10h
  Programming Interface . . . . . 00h
  Expansion ROM Base Address . . . . . FFFF0000h
  IRQ Line . . . . . 11
  IRQ Pin . . . . . INTA#
  IO Address Base . . . . . 2810h
  IO Address Length . . . . . 10h
  Memory Address Base . . . . . C6FFDDE0h
  Memory Address Length . . . . . 10h

Diskette Drive A . . . . . 1.44 Megabyte (3.5
inch)

Graphics Mode . . . . . 03 (80-Column Text)

Primary Monitor attached to . . Cirrus CL-GD5446
Graphics Controller
with Video Graphics

Color Monitor

Total Video Memory . . . . . 1024 Kbytes

Base Memory
  System Total . . . . . 640 Kbytes
  Amount Free . . . . . 597 Kbytes
(612144 Bytes)

Extended Memory
  System Total . . . . . 523264 Kbytes

Expanded Memory
  LIM Driver Support . . . . . LIM driver not
loaded

Operating System . . . . . MS-DOS version 7.00
(from diskette)

Environment variables
  PATH=
  PROMPT=$PSG
  COMSPEC=A:\COMMAND.COM
  CMDLINE=inspect /u
End of environment

```

Revisions Table

Previous Revisions
 I/O Board Revision 04
 Assembly Version 1
 Functional Revision Level D

Current Revisions
 I/O Board Revision 04
 Assembly Version 1
 Functional Revision Level D

SUT System Configuration

Date 07/19/2000
 Time 14:10:02

Product ProLiant

Machine ID
 From System Board CPQ1608

Processor Pentium III(R) Xeon
 at 700 MHz
 Slot 8
 Secondary Cache 2048K
 CPU ID 06A0

Processor Pentium III(R) Xeon
 at 700 MHz
 Slot 7
 Secondary Cache 2048K
 CPU ID 06A0

Processor Pentium III(R) Xeon
 at 700 MHz
 Slot 6
 Secondary Cache 2048K
 CPU ID 06A0

Processor Pentium III(R) Xeon
 at 700 MHz
 Slot 5
 Secondary Cache 2048K
 CPU ID 06A0

Processor Pentium III(R) Xeon
 at 700 MHz
 Slot 4
 Secondary Cache 2048K
 CPU ID 06A0

Processor Pentium III(R) Xeon
 at 700 MHz
 Slot 3
 Secondary Cache 2048K
 CPU ID 06A0

Processor Pentium III(R) Xeon
 at 700 MHz
 Slot 2
 Secondary Cache 2048K
 CPU ID 06A0

Processor Pentium III(R) Xeon
 at 700 MHz
 Slot 1
 Secondary Cache 2048K
 CPU ID 06A0

Processor(s) Mapped Out None

Numeric Coprocessor Integrated 387-
 Compatible

Expansion Bus ISA, PCI

System Identification Number D948BX71K013

CPU Mode Real Mode

System ROM
 Revision 06/16/2000
 Family P42
 Flashable Yes
 Supports F10 partition Yes

Video Controller ROM
 Revision 3.96

Option ROMs
 Address Range C0000 - C7FFF
 Data Dump (1998/10/08 15:41)

Address Range C8000 - CBFFF
 Data Dump (04/22/98 Maxwell
 Smart Array Option ROM/BIOS (C)Co...)

Address Range CC000 - CFFFF
 Data Dump (04/22/98 ROC
 ROC-LCR Option ROM/BIOS (C)Copyri...)

Address Range E8000 - EDFFF
 Data Dump (CPQSCSI d)

Bootblock ROM 04/17/2000

Standby Recovery Server
 Status Disabled
 COM Port COM1
 Server Configuration Recovery
 Timeout Value 1 minutes

Memory Boards Identified:
 System Board
 DIMM Slot 1 (SDRAM) 1024 Megabytes
 DIMM Slot 2 (SDRAM) 1024 Megabytes
 DIMM Slot 3 (SDRAM) 1024 Megabytes
 DIMM Slot 4 (SDRAM) 1024 Megabytes
 DIMM Slot 5 (SDRAM) 1024 Megabytes
 DIMM Slot 6 (SDRAM) 1024 Megabytes

DIMM Slot 7 (SDRAM) 1024 Megabytes
 DIMM Slot 8 (SDRAM) 1024 Megabytes
 DIMM Slot 9 0 Megabytes
 DIMM Slot 10 0 Megabytes
 DIMM Slot 11 0 Megabytes
 DIMM Slot 12 0 Megabytes
 DIMM Slot 13 0 Megabytes
 DIMM Slot 14 0 Megabytes
 DIMM Slot 15 0 Megabytes
 DIMM Slot 16 0 Megabytes
 Total Compaq Memory 8192 Megabytes

Keyboard Enhanced

LPT Ports LPT1 (Address 3BC)

COM Ports COM1 (Address 3F8)
 COM2 (Address 2F8)

Compaq NC3131 Fast Ethernet NIC
 Device Type Ethernet Controller
 PCI Bus Number 6
 Device Number 4
 Function Number 00h
 Slot Number 1
 Vendor ID 0E11h
 Device ID 1229h
 Subsystem Vendor ID 0E11h
 Subsystem ID B0DDh
 Revision ID 05h
 Programming Interface 00h
 Expansion ROM Base Address 0h
 IRQ Line 15
 IRQ Pin INTA#
 Memory Address Base EF8F0000h
 Memory Address Length 1000h
 IO Address Base 8000h
 IO Address Length 20h
 Memory Address Base F1F00000h
 Memory Address Length 100000h

Compaq NC3131 Fast Ethernet NIC
 Device Type Ethernet Controller
 PCI Bus Number 6
 Device Number 5
 Function Number 00h
 Slot Number 1
 Vendor ID 0E11h
 Device ID 1229h
 Subsystem Vendor ID 0E11h
 Subsystem ID B0DDh
 Revision ID 05h
 Programming Interface 00h
 Expansion ROM Base Address 0h
 IRQ Line 15
 IRQ Pin INTA#
 Memory Address Base EF8E0000h
 Memory Address Length 1000h
 IO Address Base 8020h
 IO Address Length 20h
 Memory Address Base F1E00000h
 Memory Address Length 100000h

```

Other Network Controller
PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Subsystem Vendor ID . . . . . 10E4h
Subsystem ID . . . . . 0000h
Revision ID . . . . . 00h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

```

```

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

```

```

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h

```

```

IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

Diskette Drive A . . . . . 1.44 Megabyte (3.5
inch)

Drive Controller 1, Compaq Integrated Smart Array
Controller
IDA Firmware Revision . . . . . 1.34
Array Accelerator Memory . . . . . 8188 Kbytes
Accelerator Status . . . . . Not Configured
Battery count . . . . . 0
Batteries charged . . . . . 0
Batteries failed . . . . . 0
Internal ProLiant . . . . . Bus 2, Rev. JB21

Logical Drive 1 . . . . . 9095 Megabyte
Fault Tolerance . . . . . Mirroring
OS Format . . . . . Multi-Sector
Distribution
Drive geometry (Cyl, Hds, Sec) 2177, 255, 32
Array Accelerator . . . . . Disabled

Hard Drive 1
SCSI Bus . . . . . 2
SCSI ID . . . . . 0
Serial Number . . . . .
LS633438000010171VXV
Firmware Revision 1 . . . . . 3B07
Model Number . . . . . COMPAQ BD009122BA
Initialized for Monitoring . . . . . Yes
Reference time . . . . . 94992
Sectors read . . . . . *641095218
Hard read errors . . . . . 0
Read errors retry . . . . . 0
ECC read errors . . . . . 0
Sectors written . . . . . 134727121
Hard write errors . . . . . 0
Write errors retry . . . . . 0
Seek count . . . . . 330000
Seek errors . . . . . 0
Spin cycles . . . . . 2
Spin up time . . . . . 0
Seek time track . . . . . 0%
Seek time third . . . . . 0%
Seek time full . . . . . 0%
Reallocated sectors . . . . . 2008
Recovers read failed . . . . . 0
Bus faults . . . . . 0

Hard Drive 2
SCSI Bus . . . . . 2
SCSI ID . . . . . 1
Serial Number . . . . .
LS8111980000W021090G
Firmware Revision 1 . . . . . 3B07
Model Number . . . . . COMPAQ BD009122BA
Initialized for Monitoring . . . . . Yes

```

```

Graphics Mode . . . . . 03 (80-Column Text)

Primary Monitor attached to . . ATI RAGE IIC PCI
Graphics Controller
with Video Graphics Color Monitor

Base Memory
System Total . . . . . 636 Kbytes
Amount Free . . . . . 594 Kbytes
(609088 Bytes)

Extended Memory
System Total . . . . . 8387584 Kbytes

Expanded Memory
LIM Driver Support . . . . . LIM driver not
loaded

Operating System . . . . . MS-DOS version 7.00
(from diskette)

Environment variables
PATH=
PROMPT=$PSG
COMSPEC=A:\COMMAND.COM
CMDLINE=inspect /u
End of environment

Revisions Table

Previous Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . . . . A
Memory Board 1 Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . . . . A

Current Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . . . . A
Memory Board 1 Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . . . . A

System serial number . . . . . D948BX71K013

Memory Allocation (including INSPECT)
PSP SIZE NAME TRAPPED INTERRUPTS
-----
0887 007200 COMMAND.COM 2Fh 2Eh 24h 23h 22h
0A52 218144 INSPECT.EXE FFh F9h F3h F2h F1h
F0h EFh EEh EBh E3h 3Fh 00h

System Configuration Memory
00 - 0F : 14 00 10 00 14 00 03 19 07 00 26
82 50 80 00 00
10 - 1F : 40 00 00 00 03 80 02 00 3C 00 00
00 00 00 00 02

```

```

20 - 2F : 00 00 00 00 7F 20 00 40 00 92 00
00 00 18 02 8C
30 - 3F : 00 3C 20 80 00 00 XX XX XX XX XX
XX XX XX XX XX

```

BIOS Data Area

```

40:0000 : F8 03 F8 02 00 00 00 00 BC 03 00
00 00 00 00 9F
40:0010 : 27 44 00 7C 02 00 00 00 00 00 1E
00 1E 00 00 00
40:0020 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0030 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 01 01
40:0040 : 24 00 04 00 00 2B 01 0F 02 03 50
00 00 10 00 00
40:0050 : 00 18 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0060 : 0E 0D 00 D4 03 29 30 C2 11 85 76
00 40 2B 0E 00
40:0070 : 00 00 00 12 00 01 00 00 14 14 14
14 01 01 01 01
40:0080 : 1E 00 3E 00 18 10 00 60 F9 11 0B
01 00 00 00 05
40:0090 : 17 00 00 00 2B 00 10 00 00 00 00
00 00 00 00 00
40:00A0 : 00 00 00 00 00 00 00 00 5C 13 00
C0 00 00 00 00
40:00B0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00C0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00D0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00E0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00F0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00

```

Interrupt Vector Table (including INSPECT)

```

00 - 03 : 0A62:0555 0070:0465
07BE:0016 0070:0465
04 - 07 : 0070:0465 F000:FF54
F000:4D64 F000:9BD0
08 - 0B : 07BE:001F 07BE:0028
F000:9BD0 F000:9BD0
0C - 0F : F000:9BD0 F000:9BD0
07BE:009A 0070:0465
10 - 13 : C000:12DE F000:F84D
F000:F841 0070:03EE
14 - 17 : F000:DDBF 0204:0240
0070:042D F000:EF83
18 - 1B : F000:B0BB 087F:002F
F000:FE6E 0070:045F
1C - 1F : F000:FF53 F000:0000
0000:0522 C000:2019
20 - 23 : 00C9:0FA8 00C9:0FB2
0887:0314 0887:016D
24 - 27 : 0887:0178 00C9:0FBC
00C9:0FC6 00C9:0FD0
28 - 2B : 00C9:106F 0070:0466
00C9:106F 00C9:106F

```

```

2C - 2F : 00C9:106F 00C9:106F
0887:0162 0888:01CC
30 - 33 : C90F:E4EA F000:9B00
00C9:106F 00C9:106F
34 - 37 : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
38 - 3B : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
3C - 3F : 00C9:106F 00C9:106F
00C9:106F 1B1E:04F3
40 - 43 : F000:EC59 CC1F:01C6
F000:F065 C000:242C
44 - 47 : F000:9BD0 F000:9BD0
0000:0000 F000:9BD0
48 - 4B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
4C - 4F : F000:9BD0 F000:9BD0
F000:9BD0 0070:04FC
50 - 53 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
54 - 57 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
58 - 5B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
5C - 5F : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
60 - 63 : 0000:0000 0000:0000
0000:0000 0000:0000
64 - 67 : 0000:0000 0000:0000
0000:0000 0000:0000
68 - 6B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
6C - 6F : F000:9BD0 C000:12DE
F000:9BD0 F000:9BD0
70 - 73 : 07BE:0035 F000:9C1F
F000:9BD0 07BE:00CA
74 - 77 : 07BE:00E2 F000:9C21
07BE:00FA F000:9BD0
78 - 7B : 0000:0000 0000:0000
0000:0000 0000:0000
7C - 7F : 0000:0000 0000:0000
0000:0000 0000:0000
80 - 83 : 0000:0000 0000:0000
0000:0000 0000:0000
84 - 87 : 0000:0000 0000:0000
0000:0000 0000:0000
88 - 8B : 0000:0000 0000:0000
0000:0000 0000:0000
8C - 8F : 0000:0000 0000:0000
0000:0000 0000:0000
90 - 93 : 0000:0000 0000:0000
0000:0000 0000:0000
94 - 97 : 0000:0000 0000:0000
0000:0000 0000:0000
98 - 9B : 0000:0000 0000:0000
0000:0000 0000:0000
9C - 9F : 0000:0000 0000:0000
0000:0000 0000:0000
A0 - A3 : 0000:0000 0000:0000
0000:0000 0000:0000
A4 - A7 : 0000:0000 0000:0000
0000:0000 0000:0000

```

```

A8 - AB : 0000:0000 0000:0000
0000:0000 0000:0000
AC - AF : 0000:0000 0000:0000
0000:0000 0000:0000
B0 - B3 : 0000:0000 0000:0000
0000:0000 0000:0000
B4 - B7 : 0000:0000 0000:0000
0000:0000 0000:0000
B8 - BB : 0000:0000 0000:0000
0000:0000 0000:0000
BC - BF : 0000:0000 0000:0000
0000:0000 0000:0000
C0 - C3 : 0000:0000 0000:0000
0000:0000 0000:0000
C4 - C7 : 0000:0000 0000:0000
0000:0000 0000:0000
C8 - CB : 0000:0000 0000:0000
0000:0000 0000:0000
CC - CF : 0000:0000 0000:0000
0000:0000 0000:0000
D0 - D3 : 0000:0000 0000:0000
0000:0000 0000:0000
D4 - D7 : 0000:0000 0000:0000
0000:0000 0000:0000
D8 - DB : 0000:0000 0000:0000
0000:0000 0000:0000
DC - DF : 0000:0000 0000:0000
0000:0000 0000:0000
E0 - E3 : 0000:0000 0000:0000
0008:0008 0E11:0CFC
E4 - E7 : C0ED:0000 0000:0809
0000:0400 0000:6E51
E8 - EB : 0008:0000 00C8:00C8
0078:0CFD 0CFC:0078
EC - EF : 0000:1F07 BA00:0006
1F85:0046 0087:BA00
F0 - F3 : 13C1:0046 1CE4:0013
13C1:D527 1400:1CE4
F4 - F7 : F000:4D30 03CE:0007
0000:0720 0000:5FFE
F8 - FB : 5FFE:0020 1497:6304
006A:0003 0000:0923
FC - FF : 514B:00FC 0000:00F6
EFE2:0000 0246:F000

```

PCI Devices Information

```

Signature . . . . . PCI
Config Mechanism #1 . . . . . Supported
Config Mechanism #2 . . . . . Not Supported
Spec Cycle for Config #1 . . . . . Supported
Spec Cycle for Config #2 . . . . . Not Supported
BIOS Interface Version . . . . . 2.10
Last PCI Bus Number . . . . . 142
Number of PCI Devices . . . . . 12

PCI Bus Number . . . . . 0
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 7
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller

```

```

Programming Interface . . . . . 00h
Expansion ROM Base Address . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E77C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7600000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 2
Function Number . . . . . 00h
Slot Number . . . . . 8
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E75C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7400000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 9
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . FFF00000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E73C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7200000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2800h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 13
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 1002h
Device ID . . . . . 4756h
Revision ID . . . . . 7Ah
Device Type . . . . . VGA Compatible
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . FFFE0000h
IRQ Line . . . . . 255
IRQ Pin . . . . . Not Used
Memory Address Base . . . . . E3000000h

```

```

Memory Address Length . . . . . 1000000h
IO Address Base . . . . . 2C00h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E71D0000h
Memory Address Length . . . . . 1000h

PCI Bus Number . . . . . 0
Device Number . . . . . 14
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 0E11h
Device ID . . . . . 0010h
Revision ID . . . . . 01h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . FFF80000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
IO Address Base . . . . . 3000h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E6000000h
Memory Address Length . . . . . 1000000h
Memory Address Base . . . . . E5000000h
Memory Address Length . . . . . 1000000h

PCI Bus Number . . . . . 5
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 3
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1DC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1C00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 5
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 4
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1BC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1A00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4400h
IO Address Length . . . . . 100h

```

```

PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . 00h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Revision ID . . . . . 00h
Device Type . . . . . Other Network
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h

```

```

Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

```

ProLiant is a trademark of Compaq Computer Corporation.

```

Date . . . . . 07/19/2000
Time . . . . . 14:24:28

```

```

Product . . . . . ProLiant

```

```

Machine ID
From System Board . . . . . CPQ1608

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 8
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 7
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 6
Secondary Cache . . . . . 2048K

```

```

CPU ID . . . . . 06A0
Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 5
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 4
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 3
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 2
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 1
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor(s) Mapped Out . . . . . None

```

```

Numeric Coprocessor . . . . . Integrated 387-
Compatible

```

```

Expansion Bus . . . . . ISA, PCI

```

```

System Identification Number . . . . . D948BX71K039

```

```

CPU Mode . . . . . Real Mode

```

```

System ROM
Revision . . . . . 06/16/2000
Family . . . . . P42
Flashable . . . . . Yes
Supports F10 partition . . . . . Yes

```

```

Video Controller ROM
Revision . . . . . 3.96

```

```

Option ROMs
Address Range . . . . . C0000 - C7FFF
Data Dump . . . . . (1998/10/08 15:41)

```

```

Address Range . . . . . C8000 - CBFFF
Data Dump . . . . . (04/22/98 Maxwell
Smart Array Option ROM/BIOS (C)Co...)

```

```

Address Range . . . . . CC000 - CFFFF
Data Dump . . . . . (04/22/98 ROC
ROC-LCR Option ROM/BIOS (C)Copyri...)

```

```

Address Range . . . . . E8000 - EDFFF
Data Dump . . . . . ( CPQSCSI d)

```

```

Bootblock ROM . . . . . 04/17/2000

```

```

Standby Recovery Server
Status . . . . . Disabled
COM Port . . . . . COM1
Server Configuration . . . . . Recovery
Timeout Value . . . . . 1 minutes

```

```

Memory Boards Identified:
System Board
DIMM Slot 1 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 2 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 3 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 4 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 5 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 6 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 7 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 8 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 9 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 10 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 11 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 12 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 13 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 14 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 15 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 16 (SDRAM) . . . . . 512 Megabytes
Total Compaq Memory . . . . . 8192 Megabytes

```

```

Keyboard . . . . . Enhanced

```

```

LPT Ports . . . . . LPT1 (Address 3BC)

```

```

COM Ports . . . . . COM1 (Address 3F8)
COM2 (Address 2F8)

```

```

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h

```

```

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller

```



```

PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h

Other Network Controller
PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Subsystem Vendor ID . . . . . 10E4h
Subsystem ID . . . . . 0000h
Revision ID . . . . . 00h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h

```

```

Memory Address Length . . . . . 100000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

Diskette Drive A . . . . . 1.44 Megabyte (3.5
inch)

Drive Controller 1, Compaq Integrated Smart Array
Controller
IDA Firmware Revision . . . . . 1.34
Array Accelerator Memory . . . . . 8188 Kbytes
Accelerator Status . . . . . Not Configured
Battery count . . . . . 0
Batteries charged . . . . . 0
Batteries failed . . . . . 0
Internal ProLiant . . . . . Bus 2, Rev. JB21

Logical Drive 1 . . . . . 9095 Megabyte
Fault Tolerance . . . . . Mirroring
OS Format . . . . . Multi-Sector
Distribution
Drive geometry (Cyl, Hds, Sec) 2177, 255, 32
Array Accelerator . . . . . Disabled

Hard Drive 1
SCSI Bus . . . . . 2
SCSI ID . . . . . 0
Serial Number . . . . .
LJL641710000191704QR
Firmware Revision 1 . . . . . 3B02
Model Number . . . . . COMPAQ BD00911934
Initialized for Monitoring . . . . . Yes
Reference time . . . . . 72266
Sectors read . . . . . *3065620485
Hard read errors . . . . . 0
Read errors retry . . . . . 0
ECC read errors . . . . . 0
Sectors written . . . . . 132640912
Hard write errors . . . . . 0
Write errors retry . . . . . 0

```

```

Seek count . . . . . 254608
Seek errors . . . . . 0
Spin cycles . . . . . 5
Spin up time . . . . . 0
Seek time track . . . . . 22%
Seek time third . . . . . 69%
Seek time full . . . . . 71%
Reallocated sectors . . . . . 41
Recovers read failed . . . . . 0
Bus faults . . . . . 0

Hard Drive 2
SCSI Bus . . . . . 2
SCSI ID . . . . . 1
Serial Number . . . . .
LS686971000010181RYF
Firmware Revision 1 . . . . . 3B07
Model Number . . . . . COMPAQ BD009122BA
Initialized for Monitoring . . . . . Yes
Reference time . . . . . 202709
Sectors read . . . . . *170244284
Hard read errors . . . . . 0
Read errors retry . . . . . 0
ECC read errors . . . . . 0
Sectors written . . . . . 396111246
Hard write errors . . . . . 0
Write errors retry . . . . . 0
Seek count . . . . . 700728
Seek errors . . . . . 0
Spin cycles . . . . . 5
Spin up time . . . . . 0
Seek time track . . . . . 47%
Seek time third . . . . . 69%
Seek time full . . . . . 72%
Reallocated sectors . . . . . 1052
Recovers read failed . . . . . 0
Bus faults . . . . . 4

Graphics Mode . . . . . 03 (80-Column Text)

Primary Monitor attached to . . . . . ATI RAGE IIC PCI
Graphics Controller
with Video Graphics Color Monitor

Base Memory
System Total . . . . . 636 Kbytes
Amount Free . . . . . 594 Kbytes
(609088 Bytes)

Extended Memory
System Total . . . . . 8387584 Kbytes

Expanded Memory
LIM Driver Support . . . . . LIM driver not
loaded

Operating System . . . . . MS-DOS version 7.00
(from diskette)

Environment variables
PATH=
PROMPT=$P$G

```

```

COMSPEC=A:\COMMAND.COM
CMDLINE=inspect /u
End of environment

Revisions Table

Previous Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A
Memory Board 1 Revision . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A

```

```

Current Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A
Memory Board 1 Revision . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A

```

System serial number D948BX71K039

```

Memory Allocation (including INSPECT)
PSP  SIZE  NAME  TRAPPED INTERRUPTS
-----
0887 007200  COMMAND.COM  Efh 2Fh 2Eh 24h 23h
22h
0A52 218144  INSPECT.EXE  EDh FFh F9h F3h F2h
Flh F0h EEh
Ebh E3h 3Fh 00h

```

```

System Configuration Memory
00 - 0F : 41 00 24 00 14 00 03 19 07 00 26
82 50 80 00 00
10 - 1F : 40 00 00 00 03 80 02 00 3C 00 00
00 00 00 00 02
20 - 2F : 00 00 00 00 7F 20 00 40 00 92 00
00 00 18 02 8C
30 - 3F : 00 3C 20 80 00 00 XX XX XX XX XX XX
XX XX XX XX XX

```

```

BIOS Data Area
40:0000 : F8 03 F8 02 00 00 00 00 BC 03 00
00 00 00 00 9F
40:0010 : 27 44 00 7C 02 81 00 00 00 00 26
00 26 00 1B 01
40:0020 : 1B 01 1B 01 1B 01 00 00 00 00 00
00 00 00 00 00
40:0030 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 01 01
40:0040 : 25 00 04 00 00 2B 01 0F 02 03 50
00 00 10 00 00
40:0050 : 00 18 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0060 : 0E 0D 00 D4 03 29 30 C2 11 85 76
00 D7 68 0E 00
40:0070 : 00 00 00 12 00 01 00 00 14 14 14
14 01 01 01 01
40:0080 : 1E 00 3E 00 18 10 00 60 F9 11 0B
01 00 00 00 05

```

```

40:0090 : 17 00 00 00 2B 00 10 00 00 00 00
00 00 00 00 00
40:00A0 : 00 00 00 00 00 00 00 00 5C 13 00
C0 00 00 00 00
40:00B0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00C0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00D0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00E0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00F0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00

```

Interrupt Vector Table (including INSPECT)

```

00 - 03 : 0A62:0555 0070:0465
07BE:0016 0070:0465
04 - 07 : 0070:0465 F000:FF54
F000:4D64 F000:9BD0
08 - 0B : 07BE:001F 07BE:0028
F000:9BD0 F000:9BD0
0C - 0F : F000:9BD0 F000:9BD0
07BE:009A 0070:0465
10 - 13 : C000:12DE F000:F84D
F000:F841 0070:03EE
14 - 17 : F000:DDBF 0204:0240
0070:042D F000:EF83
18 - 1B : F000:BOBB 087F:002F
F000:FE6E 0070:045F
1C - 1F : F000:FF53 F000:0000
0000:0522 C000:2019
20 - 23 : 00C9:0FA8 00C9:0FB2
0887:0314 0887:016D
24 - 27 : 0887:0178 00C9:0FBC
00C9:0FC6 00C9:0FD0
28 - 2B : 00C9:106F 0070:0466
00C9:106F 00C9:106F
2C - 2F : 00C9:106F 00C9:106F
0887:0162 0888:01CC
30 - 33 : C90F:E4EA F000:9B00
00C9:106F 00C9:106F
34 - 37 : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
38 - 3B : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
3C - 3F : 00C9:106F 00C9:106F
00C9:106F 1B1E:04F3
40 - 43 : F000:EC59 CC1F:01C6
F000:F065 C000:242C
44 - 47 : F000:9BD0 F000:9BD0
0000:0000 F000:9BD0
48 - 4B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
4C - 4F : F000:9BD0 F000:9BD0
F000:9BD0 0070:04FC
50 - 53 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
54 - 57 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
58 - 5B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0

```

```

5C - 5F : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
60 - 63 : 0000:0000 0000:0000
0000:0000 0000:0000
64 - 67 : 0000:0000 0000:0000
0000:0000 0000:0000
68 - 6B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
6C - 6F : F000:9BD0 C000:12DE
F000:9BD0 F000:9BD0
70 - 73 : 07BE:0035 F000:9C1F
F000:9BD0 07BE:00CA
74 - 77 : 07BE:00E2 F000:9C21
07BE:00FA F000:9BD0
78 - 7B : 0000:0000 0000:0000
0000:0000 0000:0000
7C - 7F : 0000:0000 0000:0000
0000:0000 0000:0000
80 - 83 : 0000:0000 0000:0000
0000:0000 0000:0000
84 - 87 : 0000:0000 0000:0000
0000:0000 0000:0000
88 - 8B : 0000:0000 0000:0000
0000:0000 0000:0000
8C - 8F : 0000:0000 0000:0000
0000:0000 0000:0000
90 - 93 : 0000:0000 0000:0000
0000:0000 0000:0000
94 - 97 : 0000:0000 0000:0000
0000:0000 0000:0000
98 - 9B : 0000:0000 0000:0000
0000:0000 0000:0000
9C - 9F : 0000:0000 0000:0000
0000:0000 0000:0000
A0 - A3 : 0000:0000 0000:0000
0000:0000 0000:0000
A4 - A7 : 0000:0000 0000:0000
0000:0000 0000:0000
A8 - AB : 0000:0000 0000:0000
0000:0000 0000:0000
AC - AF : 0000:0000 0000:0000
0000:0000 0000:0000
B0 - B3 : 0000:0000 0000:0000
0000:0000 0000:0000
B4 - B7 : 0000:0000 0000:0000
0000:0000 0000:0000
B8 - BB : 0000:0000 0000:0000
0000:0000 0000:0000
BC - BF : 0000:0000 0000:0000
0000:0000 0000:0000
C0 - C3 : 0000:0000 0000:0000
0000:0000 0000:0000
C4 - C7 : 0000:0000 0000:0000
0000:0000 0000:0000
C8 - CB : 0000:0000 0000:0000
0000:0000 0000:0000
CC - CF : 0000:0000 0000:0000
0000:0000 0000:0000
D0 - D3 : 0000:0000 0000:0000
0000:0000 0000:0000
D4 - D7 : 0000:0000 0000:0000
0000:0000 0000:0000

```

```

D8 - DB : 0000:0000 0000:0000
0000:0000 0000:0000
DC - DF : 0000:0000 0000:0000
0000:0000 0000:0000
E0 - E3 : 0000:0000 0000:0000
0008:0008 0E11:0CFC
E4 - E7 : COED:0000 0000:0809
0000:0400 0000:6E51
E8 - EB : 0008:0000 00C8:00C8
0078:0CFD 0CFC:0078
EC - EF : 0000:1F07 9100:0006
1F85:0046 0087:9100
F0 - F3 : 13C1:0046 1CE4:0013
13C1:D527 1400:1CE4
F4 - F7 : F000:4D30 03DA:0007
0000:0000 0000:7289
F8 - FB : 5FFE:0020 1497:6304
0024:0003 0000:0923
FC - FF : 514B:00FC 0000:00F6
EFE2:0000 0246:F000

```

PCI Devices Information

```

Signature . . . . . PCI
Config Mechanism #1 . . . . . Supported
Config Mechanism #2 . . . . . Not Supported
Spec Cycle for Config #1 . . . . . Supported
Spec Cycle for Config #2 . . . . . Not Supported
BIOS Interface Version . . . . . 2.10
Last PCI Bus Number . . . . . 142
Number of PCI Devices . . . . . 12

PCI Bus Number . . . . . 0
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 7
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E77C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7600000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 2
Function Number . . . . . 00h
Slot Number . . . . . 8
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E75C0000h

```

```

Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7400000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 9
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E73C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7200000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2800h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 13
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 1002h
Device ID . . . . . 4756h
Revision ID . . . . . 7Ah
Device Type . . . . . VGA Compatible
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFFE0000h
IRQ Line . . . . . 255
IRQ Pin . . . . . Not Used
Memory Address Base . . . . . E3000000h
Memory Address Length . . . . . 1000000h
IO Address Base . . . . . 2C00h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E71D0000h
Memory Address Length . . . . . 1000h

PCI Bus Number . . . . . 0
Device Number . . . . . 14
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 0E11h
Device ID . . . . . 0010h
Revision ID . . . . . 01h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF80000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
IO Address Base . . . . . 3000h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E6000000h
Memory Address Length . . . . . 1000000h
Memory Address Base . . . . . E5000000h
Memory Address Length . . . . . 1000000h

```

```

PCI Bus Number . . . . . 5
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 3
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1DC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1C00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4000h
IO Address Length . . . . . 100h

```

```

PCI Bus Number . . . . . 5
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 4
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1BC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1A00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4400h
IO Address Length . . . . . 100h

```

```

PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h

```

```

PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h

```

```

Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Revision ID . . . . . 00h
Device Type . . . . . Other Network
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h

```

```

IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

ProLiant is a trademark of Compaq Computer
Corporation.

Date . . . . . 07/19/2000
Time . . . . . 14:56:18

Product . . . . . ProLiant

Machine ID
From System Board . . . . . CPQ1608

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 8
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 7
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 6
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 5
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 4
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 3
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 2
Secondary Cache . . . . . 2048K

```

```

CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 1
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor(s) Mapped Out . . . . . None

Numeric Coprocessor . . . . . Integrated 387-
Compatible

Expansion Bus . . . . . ISA, PCI

System Identification Number . . . . . 0927BX710018

CPU Mode . . . . . Real Mode

System ROM
Revision . . . . . 06/16/2000
Family . . . . . P42
Flashable . . . . . Yes
Supports F10 partition . . . . . Yes

Video Controller ROM
Revision . . . . . 3.96

Option ROMs
Address Range . . . . . C0000 - C7FFF
Data Dump . . . . . (1998/10/08 15:41)

Address Range . . . . . C8000 - CBFFF
Data Dump . . . . . (04/22/98 Maxwell
Smart Array Option ROM/BIOS (C)Co...)

Address Range . . . . . CC000 - CFFFF
Data Dump . . . . . (04/22/98 ROC
ROC-LCR Option ROM/BIOS (C)Copyri...)

Address Range . . . . . E8000 - EDFFF
Data Dump . . . . . ( CPQSCSI d)

Bootblock ROM . . . . . 04/17/2000

Standby Recovery Server
Status . . . . . Disabled
COM Port . . . . . COM1
Server Configuration . . . . . Recovery
Timeout Value . . . . . 1 minutes

Memory Boards Identified:
System Board
DIMM Slot 1 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 2 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 3 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 4 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 5 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 6 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 7 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 8 (SDRAM) . . . . . 1024 Megabytes

```

```

DIMM Slot 9 . . . . . 0 Megabytes
DIMM Slot 10 . . . . . 0 Megabytes
DIMM Slot 11 . . . . . 0 Megabytes
DIMM Slot 12 . . . . . 0 Megabytes
DIMM Slot 13 . . . . . 0 Megabytes
DIMM Slot 14 . . . . . 0 Megabytes
DIMM Slot 15 . . . . . 0 Megabytes
DIMM Slot 16 . . . . . 0 Megabytes
Total Compaq Memory . . . . . 8192 Megabytes

Keyboard . . . . . Enhanced

LPT Ports . . . . . LPT1 (Address 378)

COM Ports . . . . . COM1 (Address 3F8)
                  COM2 (Address 2F8)

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F38F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 5000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F3F00000h
Memory Address Length . . . . . 100000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F38E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 5020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F3E00000h
Memory Address Length . . . . . 100000h

Other Network Controller

```

```

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Subsystem Vendor ID . . . . . 10E4h
Subsystem ID . . . . . 0000h
Revision ID . . . . . 00h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7CF0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 6000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7CE0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 6020h
IO Address Length . . . . . 20h

```

```

Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

Diskette Drive A . . . . . 1.44 Megabyte (3.5
inch)

Drive Controller 1, Compaq Integrated Smart Array
Controller
IDA Firmware Revision . . . . . 1.34
Array Accelerator Memory . . . . . 8188 Kbytes
Accelerator Status . . . . . Not Configured
Battery count . . . . . 0
Batteries charged . . . . . 0
Batteries failed . . . . . 0
Internal ProLiant . . . . . Bus 2, Rev. JB21

Logical Drive 1 . . . . . 9095 Megabyte
Fault Tolerance . . . . . Mirroring
OS Format . . . . . Multi-Sector
Distribution
Drive geometry (Cyl, Hds, Sec) 2177, 255, 32
Array Accelerator . . . . . Disabled

Hard Drive 1
SCSI Bus . . . . . 2
SCSI ID . . . . . 0
Serial Number . . . . .
LS73036000010191RAP
Firmware Revision 1 . . . . . 3B07
Model Number . . . . . COMPAQ BD009122BA
Initialized for Monitoring . . . . . Yes
Reference time . . . . . 95946
Sectors read . . . . . *3295577002
Hard read errors . . . . . 0
Read errors retry . . . . . 0
ECC read errors . . . . . 0
Sectors written . . . . . 114855468
Hard write errors . . . . . 0
Write errors retry . . . . . 0
Seek count . . . . . 350592
Seek errors . . . . . 0
Spin cycles . . . . . 4
Spin up time . . . . . 0
Seek time track . . . . . 47%
Seek time third . . . . . 70%
Seek time full . . . . . 72%
Reallocated sectors . . . . . 668
Recovers read failed . . . . . 0
Bus faults . . . . . 0

Hard Drive 2
SCSI Bus . . . . . 2
SCSI ID . . . . . 1
Serial Number . . . . .
LS76349200010071H3L
Firmware Revision 1 . . . . . 3B07
Model Number . . . . . COMPAQ BD009122BA
Initialized for Monitoring . . . . . Yes
Reference time . . . . . 49071
Sectors read . . . . . *4291839352
Hard read errors . . . . . 0

```

```

Read errors retry . . . . . 0
ECC read errors . . . . . 0
Sectors written . . . . . 146462573
Hard write errors . . . . . 0
Write errors retry . . . . . 0
Seek count . . . . . 200112
Seek errors . . . . . 0
Spin cycles . . . . . 4
Spin up time . . . . . 0
Seek time track . . . . . 52%
Seek time third . . . . . 67%
Seek time full . . . . . 70%
Reallocated sectors . . . . 1296
Recovers read failed . . . . 0
Bus faults . . . . . 0

Graphics Mode . . . . . 03 (80-Column Text)

Primary Monitor attached to . . ATI RAGE IIC PCI
Graphics Controller
with Video Graphics Color Monitor

Base Memory
System Total . . . . . 636 Kbytes
Amount Free . . . . . 594 Kbytes
(609088 Bytes)

Extended Memory
System Total . . . . . 8387584 Kbytes

Expanded Memory
LIM Driver Support . . . . . LIM driver not
loaded

Operating System . . . . . MS-DOS version 7.00
(from diskette)

Environment variables
PATH=
PROMPT=$PSG
COMSPEC=A:\COMMAND.COM
CMDLINE=inspect /u
End of environment

Revisions Table

Previous Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A
Memory Board 1 Revision . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A

Current Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A
Memory Board 1 Revision . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A

```

```

System serial number . . . . . 0927BX710018

Memory Allocation (including INSPECT)
PSP SIZE NAME TRAPPED INTERRUPTS
-----
0887 007200 COMMAND.COM 2Fh 2Eh 24h 23h 22h
0A52 218144 INSPECT.EXE FFh F9h F4h F3h F2h
Flh F0h EEh
Ebh 3Fh 00h

System Configuration Memory
00 - 0F : 31 00 56 00 14 00 03 19 07 00 26
82 50 80 00 00
10 - 1F : 40 00 00 00 03 80 02 00 3C 00 00
00 00 00 00 02
20 - 2F : 00 00 00 00 7F 20 00 40 00 92 00
00 00 18 02 8C
30 - 3F : 00 3C 20 80 00 00 XX XX XX XX XX
XX XX XX XX XX

BIOS Data Area
40:0000 : F8 03 F8 02 00 00 00 00 78 03 00
00 00 00 00 9F
40:0010 : 27 44 00 7C 02 00 00 00 00 00 1E
00 1E 00 00 00
40:0020 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0030 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 01 01
40:0040 : 25 00 04 00 00 2B 01 0F 02 03 50
00 00 10 00 00
40:0050 : 00 18 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0060 : 0E 0D 00 D4 03 29 30 C2 11 85 76
00 AA F0 0E 00
40:0070 : 00 00 00 12 00 01 00 00 14 14 14
14 01 01 01 01
40:0080 : 1E 00 3E 00 18 10 00 60 F9 11 0B
01 00 00 00 05
40:0090 : 17 00 00 00 2B 00 10 00 00 00 00
00 00 00 00 00
40:00A0 : 00 00 00 00 00 00 00 00 5C 13 00
C0 00 00 00 00
40:00B0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00C0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00D0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00E0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00F0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00

Interrupt Vector Table (including INSPECT)
00 - 03 : 0A62:0555 0070:0465
07BE:0016 0070:0465
04 - 07 : 0070:0465 F000:FF54
F000:4D64 F000:9BD0
08 - 0B : 07BE:001F 07BE:0028
F000:9BD0 F000:9BD0

```

```

0C - 0F : F000:9BD0 07BE:0082
07BE:009A 0070:0465
10 - 13 : C000:12DE F000:F84D
F000:F841 0070:03EE
14 - 17 : F000:DDBF 0204:0240
0070:042D F000:EF83
18 - 1B : F000:BOBB 087F:002F
F000:FE6E 0070:045F
1C - 1F : F000:FF53 F000:0000
0000:0522 C000:2019
20 - 23 : 00C9:0FA8 00C9:0FB2
0887:0314 0887:016D
24 - 27 : 0887:0178 00C9:0FBC
00C9:0FC6 00C9:0FD0
28 - 2B : 00C9:106F 0070:0466
00C9:106F 00C9:106F
2C - 2F : 00C9:106F 00C9:106F
0887:0162 0888:01CC
30 - 33 : C90F:E4EA F000:9B00
00C9:106F 00C9:106F
34 - 37 : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
38 - 3B : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
3C - 3F : 00C9:106F 00C9:106F
00C9:106F 1B1E:04F3
40 - 43 : F000:EC59 CC1F:01C6
F000:F065 C000:242C
44 - 47 : F000:9BD0 F000:9BD0
0000:0000 F000:9BD0
48 - 4B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
4C - 4F : F000:9BD0 F000:9BD0
F000:9BD0 0070:04FC
50 - 53 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
54 - 57 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
58 - 5B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
5C - 5F : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
60 - 63 : 0000:0000 0000:0000
0000:0000 0000:0000
64 - 67 : 0000:0000 0000:0000
0000:0000 0000:0000
68 - 6B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
6C - 6F : F000:9BD0 C000:12DE
F000:9BD0 F000:9BD0
70 - 73 : 07BE:0035 F000:9C1F
F000:9BD0 F000:9BD0
74 - 77 : 07BE:00E2 F000:9C21
07BE:00FA F000:9BD0
78 - 7B : 0000:0000 0000:0000
0000:0000 0000:0000
7C - 7F : 0000:0000 0000:0000
0000:0000 0000:0000
80 - 83 : 0000:0000 0000:0000
0000:0000 0000:0000
84 - 87 : 0000:0000 0000:0000
0000:0000 0000:0000

```

```

88 - 8B : 0000:0000 0000:0000
0000:0000 0000:0000
8C - 8F : 0000:0000 0000:0000
0000:0000 0000:0000
90 - 93 : 0000:0000 0000:0000
0000:0000 0000:0000
94 - 97 : 0000:0000 0000:0000
0000:0000 0000:0000
98 - 9B : 0000:0000 0000:0000
0000:0000 0000:0000
9C - 9F : 0000:0000 0000:0000
0000:0000 0000:0000
A0 - A3 : 0000:0000 0000:0000
0000:0000 0000:0000
A4 - A7 : 0000:0000 0000:0000
0000:0000 0000:0000
A8 - AB : 0000:0000 0000:0000
0000:0000 0000:0000
AC - AF : 0000:0000 0000:0000
0000:0000 0000:0000
B0 - B3 : 0000:0000 0000:0000
0000:0000 0000:0000
B4 - B7 : 0000:0000 0000:0000
0000:0000 0000:0000
B8 - BB : 0000:0000 0000:0000
0000:0000 0000:0000
BC - BF : 0000:0000 0000:0000
0000:0000 0000:0000
C0 - C3 : 0000:0000 0000:0000
0000:0000 0000:0000
C4 - C7 : 0000:0000 0000:0000
0000:0000 0000:0000
C8 - CB : 0000:0000 0000:0000
0000:0000 0000:0000
CC - CF : 0000:0000 0000:0000
0000:0000 0000:0000
D0 - D3 : 0000:0000 0000:0000
0000:0000 0000:0000
D4 - D7 : 0000:0000 0000:0000
0000:0000 0000:0000
D8 - DB : 0000:0000 0000:0000
0000:0000 0000:0000
DC - DF : 0000:0000 0000:0000
0000:0000 0000:0000
E0 - E3 : 0000:0000 0000:0000
0000:0000 0000:0000
E4 - E7 : 0000:0000 0000:0000
0000:0000 00D8:00D8
E8 - EB : 000D:0CFC 00C8:00C8
0078:0CFD 0CFC:0078
EC - EF : 0000:1F07 6F00:0006
1F85:0046 0087:6F00
F0 - F3 : 156B:0046 1C96:0013
156B:D527 1400:1C96
F4 - F7 : 1B00:0246 0101:7248
0000:0000 0000:5FFE
F8 - FB : 5FFE:0020 1497:6304
0084:0003 0000:072B
FC - FF : 514B:00FC 0000:00F6
EFE2:0000 0246:F000

PCI Devices Information
Signature . . . . . PCI

```

```

Config Mechanism #1 . . . . . Supported
Config Mechanism #2 . . . . . Not Supported
Spec Cycle for Config #1 . . . . . Supported
Spec Cycle for Config #2 . . . . . Not Supported
BIOS Interface Version . . . . . 2.10
Last PCI Bus Number . . . . . 16
Number of PCI Devices . . . . . 12

PCI Bus Number . . . . . 0
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 7
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F37C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F3600000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 2
Function Number . . . . . 00h
Slot Number . . . . . 8
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F35C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F3400000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 9
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F33C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F3200000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2800h

```

```

IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 13
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 1002h
Device ID . . . . . 4756h
Revision ID . . . . . 7Ah
Device Type . . . . . VGA Compatible
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFFE0000h
IRQ Line . . . . . 255
IRQ Pin . . . . . Not Used
Memory Address Base . . . . . F0000000h
Memory Address Length . . . . . 1000000h
IO Address Base . . . . . 2C00h
IO Address Length . . . . . 100h
Memory Address Base . . . . . F31D0000h
Memory Address Length . . . . . 1000h

PCI Bus Number . . . . . 0
Device Number . . . . . 14
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 0E11h
Device ID . . . . . 0010h
Revision ID . . . . . 01h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF80000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
IO Address Base . . . . . 3000h
IO Address Length . . . . . 100h
Memory Address Base . . . . . F2000000h
Memory Address Length . . . . . 1000000h
Memory Address Base . . . . . F1000000h
Memory Address Length . . . . . 1000000h

PCI Bus Number . . . . . 5
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 3
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F3DC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F3C00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 5
Device Number . . . . . 4
Function Number . . . . . 00h

```

```

Slot Number . . . . . 4
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F3BC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F3A00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F38F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 5000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F3F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F38E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 5020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F3E00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Revision ID . . . . . 00h
Device Type . . . . . Other Network
Controller

```

```

Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7CF0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 6000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7CE0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 6020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

ProLiant is a trademark of Compaq Computer
Corporation.

Date . . . . . 07/19/2000
Time . . . . . 13:59:47

Product . . . . . ProLiant

Machine ID
From System Board . . . . . CPQ1608

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 8
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 7
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 6
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 5
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 4
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 3
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 2
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 1
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor(s) Mapped Out . . . . . None

Numeric Coprocessor . . . . . Integrated 387-
Compatible

Expansion Bus . . . . . ISA, PCI

System Identification Number . . . . . D935BX71A011

CPU Mode . . . . . Real Mode

System ROM
Revision . . . . . 06/16/2000
Family . . . . . P42
Flashable . . . . . Yes

```



```

Supports F10 partition . . . . Yes
Video Controller ROM
Revision . . . . . 3.96
Option ROMs
Address Range . . . . . C0000 - C7FFF
Data Dump . . . . . (1998/10/08 15:41)
Address Range . . . . . C8000 - CBFFF
Data Dump . . . . . (04/22/98 Maxwell)
Smart Array Option ROM/BIOS (C)Co...
Address Range . . . . . CC000 - CFFFF
Data Dump . . . . . (04/22/98 ROC
ROC-LCR Option ROM/BIOS (C)Copyri...
Address Range . . . . . E8000 - EDFFF
Data Dump . . . . . ( CPQSCSI d)
Bootblock ROM . . . . . 04/17/2000
Standby Recovery Server
Status . . . . . Disabled
COM Port . . . . . COM1
Server Configuration . . . . . Recovery
Timeout Value . . . . . 1 minutes
Memory Boards Identified:
System Board
DIMM Slot 1 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 2 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 3 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 4 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 5 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 6 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 7 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 8 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 9 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 10 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 11 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 12 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 13 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 14 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 15 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 16 (SDRAM) . . . . . 512 Megabytes
Total Compaq Memory . . . . . 8192 Megabytes
Keyboard . . . . . Standard 11-Bit
LPT Ports . . . . . LPT1 (Address 378)
COM Ports . . . . . COM1 (Address 3F8)
COM2 (Address 2F8)
Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h

```

```

Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h
Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h
Other Network Controller
PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Subsystem Vendor ID . . . . . 10E4h
Subsystem ID . . . . . 0000h
Revision ID . . . . . 00h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h
Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller

```

```

PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h
Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h
Diskette Drive A . . . . . 1.44 Megabyte (3.5
inch)
Drive Controller 1, Compaq Integrated Smart Array
Controller
IDA Firmware Revision . . . . . 1.34
Array Accelerator Memory . . . . . 8188 Kbytes
Accelerator Status . . . . . Not Configured
Battery count . . . . . 0
Batteries charged . . . . . 0
Batteries failed . . . . . 0
Internal ProLiant . . . . . Bus 2, Rev. JB21
Logical Drive 1 . . . . . 9095 Megabyte
Fault Tolerance . . . . . Mirroring
OS Format . . . . . Multi-Sector
Distribution
Drive geometry (Cyl, Hds, Sec) 2177, 255, 32

```

```

Array Accelerator . . . . . Disabled

Hard Drive 1
SCSI Bus . . . . . 2
SCSI ID . . . . . 0
Serial Number . . . . .
LS784065000W0210793
Firmware Revision 1 . . . . 3B07
Model Number . . . . . COMPAQ BD009122BA
Initialized for Monitoring . Yes

Hard Drive 2
SCSI Bus . . . . . 2
SCSI ID . . . . . 1
Serial Number . . . . .
LS729834000010191S8V
Firmware Revision 1 . . . . 3B07
Model Number . . . . . COMPAQ BD009122BA
Initialized for Monitoring . Yes
Reference time . . . . . 78986
Sectors read . . . . . *838327270
Hard read errors . . . . . 0
Read errors retry . . . . . 0
ECC read errors . . . . . 0
Sectors written . . . . . 138549931
Hard write errors . . . . . 0
Write errors retry . . . . . 0
Seek count . . . . . 301312
Seek errors . . . . . 0
Spin cycles . . . . . 4
Spin up time . . . . . 0
Seek time track . . . . . 47%
Seek time third . . . . . 69%
Seek time full . . . . . 71%
Reallocated sectors . . . . 1008
Recovers read failed . . . . 0
Bus faults . . . . . 0

Graphics Mode . . . . . 03 (80-Column Text)

Primary Monitor attached to . . ATI RAGE IIC PCI
Graphics Controller
with Video Graphics Color Monitor

Base Memory
System Total . . . . . 636 Kbytes
Amount Free . . . . . 594 Kbytes
(609088 Bytes)

Extended Memory
System Total . . . . . 8387584 Kbytes

Expanded Memory
LIM Driver Support . . . . . LIM driver not
loaded

Operating System . . . . . MS-DOS version 7.00
(from diskette)

Environment variables
PATH=
PROMPT=$P$G

```

```

COMSPEC=A:\COMMAND.COM
CMDLINE=inspect /u
End of environment

Revisions Table

Previous Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . A
Memory Board 1 Revision . . . 01
Assembly Version . . . . . 1
Functional Revision Level . A

Current Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . A
Memory Board 1 Revision . . . 01
Assembly Version . . . . . 1
Functional Revision Level . A

System serial number . . . . . D935BX71A011

Memory Allocation (including INSPECT)
PSP SIZE NAME TRAPPED INTERRUPTS
-----
0887 007200 COMMAND.COM 2Fh 2Eh 24h 23h 22h
0A52 218144 INSPECT.EXE FFh F9h F0h EFh EEh
EBh E4h 3Fh

System Configuration Memory
00 - 0F : 00 00 00 00 14 00 03 19 07 00 26
82 50 80 00 00
10 - 1F : 40 00 00 00 03 80 02 00 3C 00 00
00 00 00 00 02
20 - 2F : 00 00 00 00 7F 20 00 40 00 92 00
00 00 18 02 8C
30 - 3F : 00 3C 20 80 00 00 XX XX XX XX XX XX
XX XX XX XX XX

BIOS Data Area
40:0000 : F8 03 F8 02 00 00 00 00 78 03 00
00 00 00 00 9F
40:0010 : 27 44 00 7C 02 00 00 00 00 00 1E
00 1E 00 00 00
40:0020 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0030 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 01 01
40:0040 : 25 00 04 00 00 2B 01 0F 02 03 50
00 00 10 00 00
40:0050 : 00 18 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0060 : 0E 0D 00 D4 03 29 30 C2 11 85 76
00 81 FF 0D 00
40:0070 : 00 00 00 12 00 01 00 00 14 14 14
14 01 01 01 01
40:0080 : 1E 00 3E 00 18 10 00 60 F9 11 0B
01 00 00 00 05

```

```

40:0090 : 17 00 00 00 2B 00 00 00 00 00 00
00 00 00 00 00
40:00A0 : 00 00 00 00 00 00 00 00 5C 13 00
C0 00 00 00 00
40:00B0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00C0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00D0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00E0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00F0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00

Interrupt Vector Table (including INSPECT)
00 - 03 : 0A62:0555 0070:0465
07BE:0016 0070:0465
04 - 07 : 0070:0465 F000:FF54
F000:4D64 F000:9BD0
08 - 0B : 07BE:001F 07BE:0028
F000:9BD0 F000:9BD0
0C - 0F : F000:9BD0 07BE:0082
07BE:009A 0070:0465
10 - 13 : C000:12DE F000:F84D
F000:F841 0070:03EE
14 - 17 : F000:DBDF 0204:0240
0070:042D F000:EF83
18 - 1B : F000:BOBB 087F:002F
F000:FE6E 0070:045F
1C - 1F : F000:FF53 F000:0000
0000:0522 C000:2019
20 - 23 : 00C9:0FA8 00C9:0FB2
0887:0314 0887:016D
24 - 27 : 0887:0178 00C9:0FBC
00C9:0FC6 00C9:0FD0
28 - 2B : 00C9:106F 0070:0466
00C9:106F 00C9:106F
2C - 2F : 00C9:106F 00C9:106F
0887:0162 0888:01CC
30 - 33 : C90F:E4EA F000:9B00
00C9:106F 00C9:106F
34 - 37 : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
38 - 3B : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
3C - 3F : 00C9:106F 00C9:106F
00C9:106F 1B1E:04F3
40 - 43 : F000:EC59 CC1F:01C6
F000:F065 C000:242C
44 - 47 : F000:9BD0 F000:9BD0
0000:0000 F000:9BD0
48 - 4B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
4C - 4F : F000:9BD0 F000:9BD0
F000:9BD0 0070:04FC
50 - 53 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
54 - 57 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
58 - 5B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0

```

```

5C - 5F : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
60 - 63 : 0000:0000 0000:0000
0000:0000 0000:0000
64 - 67 : 0000:0000 0000:0000
0000:0000 0000:0000
68 - 6B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
6C - 6F : F000:9BD0 C000:12DE
F000:9BD0 F000:9BD0
70 - 73 : 07BE:0035 F000:9C1F
F000:9BD0 F000:9BD0
74 - 77 : 07BE:00E2 F000:9C21
07BE:00FA F000:9BD0
78 - 7B : 0000:0000 0000:0000
0000:0000 0000:0000
7C - 7F : 0000:0000 0000:0000
0000:0000 0000:0000
80 - 83 : 0000:0000 0000:0000
0000:0000 0000:0000
84 - 87 : 0000:0000 0000:0000
0000:0000 0000:0000
88 - 8B : 0000:0000 0000:0000
0000:0000 0000:0000
8C - 8F : 0000:0000 0000:0000
0000:0000 0000:0000
90 - 93 : 0000:0000 0000:0000
0000:0000 0000:0000
94 - 97 : 0000:0000 0000:0000
0000:0000 0000:0000
98 - 9B : 0000:0000 0000:0000
0000:0000 0000:0000
9C - 9F : 0000:0000 0000:0000
0000:0000 0000:0000
A0 - A3 : 0000:0000 0000:0000
0000:0000 0000:0000
A4 - A7 : 0000:0000 0000:0000
0000:0000 0000:0000
A8 - AB : 0000:0000 0000:0000
0000:0000 0000:0000
AC - AF : 0000:0000 0000:0000
0000:0000 0000:0000
B0 - B3 : 0000:0000 0000:0000
0000:0000 0000:0000
B4 - B7 : 0000:0000 0000:0000
0000:0000 0000:0000
B8 - BB : 0000:0000 0000:0000
0000:0000 0000:0000
BC - BF : 0000:0000 0000:0000
0000:0000 0000:0000
C0 - C3 : 0000:0000 0000:0000
0000:0000 0000:0000
C4 - C7 : 0000:0000 0000:0000
0000:0000 0000:0000
C8 - CB : 0000:0000 0000:0000
0000:0000 0000:0000
CC - CF : 0000:0000 0000:0000
0000:0000 0000:0000
D0 - D3 : 0000:0000 0000:0000
0000:0000 0000:0000
D4 - D7 : 0000:0000 0000:0000
0000:0000 0000:0000

```

```

D8 - DB : 0000:0000 0000:0000
0000:0000 0000:0000
DC - DF : 0000:0000 0000:0000
0000:0000 0000:0000
E0 - E3 : 0000:0000 0000:0000
0000:0000 00D8:0000
E4 - E7 : 0CFC:00D8 0000:000D
0020:0020 FFFF:0CFC
E8 - EB : C0ED:0000 00C8:00C8
0078:0CFD 0CFC:0078
EC - EF : 0000:1F07 A700:0006
1F85:0046 0087:A700
F0 - F3 : 13C1:0046 F000:4D30
0000:0007 C000:0000
F4 - F7 : C000:74E7 0101:0246
0000:0000 0000:5FFE 1497:6304
F8 - FB : 5FFE:0020
00FA:0003 0000:08FA
FC - FF : 514B:00FC 0000:00F6
EFE2:0000 0246:F000

```

PCI Devices Information

```

Signature . . . . . PCI
Config Mechanism #1 . . . . . Supported
Config Mechanism #2 . . . . . Not Supported
Spec Cycle for Config #1 . . . . . Supported
Spec Cycle for Config #2 . . . . . Not Supported
BIOS Interface Version . . . . . 2.10
Last PCI Bus Number . . . . . 142
Number of PCI Devices . . . . . 12

PCI Bus Number . . . . . 0
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 7
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E77C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7600000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 2
Function Number . . . . . 00h
Slot Number . . . . . 8
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E75C0000h

```

```

Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7400000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 9
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E73C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7200000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2800h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 13
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 1002h
Device ID . . . . . 4756h
Revision ID . . . . . 7Ah
Device Type . . . . . VGA Compatible
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFFE0000h
IRQ Line . . . . . 255
IRQ Pin . . . . . Not Used
Memory Address Base . . . . . E3000000h
Memory Address Length . . . . . 1000000h
IO Address Base . . . . . 2C00h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E71D0000h
Memory Address Length . . . . . 1000h

PCI Bus Number . . . . . 0
Device Number . . . . . 14
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 0E11h
Device ID . . . . . 0010h
Revision ID . . . . . 01h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF80000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
IO Address Base . . . . . 3000h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E6000000h
Memory Address Length . . . . . 1000000h
Memory Address Base . . . . . E5000000h
Memory Address Length . . . . . 1000000h

```

```

PCI Bus Number . . . . . 5
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 3
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1DC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1C00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 5
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 4
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1BC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1A00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h

```

```

Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Revision ID . . . . . 00h
Device Type . . . . . Other Network
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h

```

```

IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

ProLiant is a trademark of Compaq Computer
Corporation.

Date . . . . . 07/19/2000
Time . . . . . 13:58:57

Product . . . . . ProLiant

Machine ID
From System Board . . . . . CPQ1608

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 8
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 7
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 6
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 5
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 4
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 3
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 2
Secondary Cache . . . . . 2048K

```

```

CPU ID . . . . . 06A0
Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 1
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor(s) Mapped Out . . . . . None

Numeric Coprocessor . . . . . Integrated 387-
Compatible

Expansion Bus . . . . . ISA, PCI

System Identification Number . . . . . D021DR51K004

CPU Mode . . . . . Real Mode

System ROM
Revision . . . . . 06/16/2000
Family . . . . . P42
Flashable . . . . . Yes
Supports F10 partition . . . . . Yes

Video Controller ROM
Revision . . . . . 3.96

Option ROMs
Address Range . . . . . C0000 - C7FFF
Data Dump . . . . . (1998/10/08 15:41)

Address Range . . . . . C8000 - CBFFF
Data Dump . . . . . (04/22/98 Maxwell)

Smart Array Option ROM/BIOS (C)Co...
Address Range . . . . . CC000 - CFFFF
Data Dump . . . . . (04/22/98 ROC

Smart Array Option ROM/BIOS (C)Co...
Address Range . . . . . E8000 - EDFFF
Data Dump . . . . . ( CPQSCSI d)

Bootblock ROM . . . . . 04/17/2000

Standby Recovery Server
Status . . . . . Disabled
COM Port . . . . . COM1
Server Configuration . . . . . Recovery
Timeout Value . . . . . 1 minutes

Memory Boards Identified:
System Board
DIMM Slot 1 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 2 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 3 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 4 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 5 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 6 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 7 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 8 (SDRAM) . . . . . 1024 Megabytes

```

```

DIMM Slot 9 . . . . . 0 Megabytes
DIMM Slot 10 . . . . . 0 Megabytes
DIMM Slot 11 . . . . . 0 Megabytes
DIMM Slot 12 . . . . . 0 Megabytes
DIMM Slot 13 . . . . . 0 Megabytes
DIMM Slot 14 . . . . . 0 Megabytes
DIMM Slot 15 . . . . . 0 Megabytes
DIMM Slot 16 . . . . . 0 Megabytes
Total Compaq Memory . . . . . 8192 Megabytes

Keyboard . . . . . Standard 11-Bit

LPT Ports . . . . . LPT1 (Address 378)

COM Ports . . . . . COM1 (Address 3F8)
COM2 (Address 2F8)

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h

Other Network Controller

```

```

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Subsystem Vendor ID . . . . . 10E4h
Subsystem ID . . . . . 0000h
Revision ID . . . . . 00h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h

```

```

Memory Address Base . . . . . F7E0000h
Memory Address Length . . . . . 100000h

Diskette Drive A . . . . . 1.44 Megabyte (3.5
inch)

Drive Controller 1, Compaq Integrated Smart Array
Controller
  IDA Firmware Revision . . . . . 1.40
  Array Accelerator Memory . . . . . 8188 Kbytes
  Accelerator Status . . . . . Not Configured
  Battery count . . . . . 0
  Batteries charged . . . . . 0
  Batteries failed . . . . . 0
  Internal ProLiant . . . . . Bus 2, Rev. JB21

  Logical Drive 1 . . . . . 9095 Megabyte
  Fault Tolerance . . . . . Mirroring
  OS Format . . . . . Multi-Sector
  Distribution
  Drive geometry (Cyl, Hds, Sec) 2177, 255, 32
  Array Accelerator . . . . . Disabled

  Hard Drive 1
  SCSI Bus . . . . . 2
  SCSI ID . . . . . 0
  Serial Number . . . . .
LS766756000102010PH
  Firmware Revision 1 . . . . . 3B07
  Model Number . . . . . COMPAQ BD009122BA
  Initialized for Monitoring . Yes
  Reference time . . . . . 100152
  Sectors read . . . . . *3133155901
  Hard read errors . . . . . 0
  Read errors retry . . . . . 0
  ECC read errors . . . . . 0
  Sectors written . . . . . 280820225
  Hard write errors . . . . . 0
  Write errors retry . . . . . 0
  Seek count . . . . . 364448
  Seek errors . . . . . 0
  Spin cycles . . . . . 4
  Spin up time . . . . . 0
  Seek time track . . . . . 47%
  Seek time third . . . . . 70%
  Seek time full . . . . . 71%
  Reallocated sectors . . . . . 100
  Recovers read failed . . . . . 0
  Bus faults . . . . . 0

  Hard Drive 2
  SCSI Bus . . . . . 2
  SCSI ID . . . . . 1
  Serial Number . . . . .
LS81129200070211G3D
  Firmware Revision 1 . . . . . 3B07
  Model Number . . . . . COMPAQ BD009122BA
  Initialized for Monitoring . Yes

Graphics Mode . . . . . 03 (80-Column Text)

```

```

Primary Monitor attached to . . ATI RAGE IIC PCI
Graphics Controller
with Video Graphics Color Monitor

Base Memory
  System Total . . . . . 636 Kbytes
  Amount Free . . . . . 594 Kbytes
(609088 Bytes)

Extended Memory
  System Total . . . . . 8387584 Kbytes

Expanded Memory
  LIM Driver Support . . . . . LIM driver not
loaded

Operating System . . . . . MS-DOS version 7.00
(from diskette)

Environment variables
  PATH=
  PROMPT=$P$G
  COMSPEC=A:\COMMAND.COM
  CMDLINE=inspect /u
End of environment

Revisions Table

Previous Revisions
  I/O Board Revision . . . . . 01
  Assembly Version . . . . . 1
  Functional Revision Level . A
  Memory Board 1 Revision . . . . . 01
  Assembly Version . . . . . 1
  Functional Revision Level . A

Current Revisions
  I/O Board Revision . . . . . 01
  Assembly Version . . . . . 1
  Functional Revision Level . A
  Memory Board 1 Revision . . . . . 01
  Assembly Version . . . . . 1
  Functional Revision Level . A

System serial number . . . . . D021DR51K004

Memory Allocation (including INSPECT)
  PSP   SIZE   NAME           TRAPPED INTERRUPTS
  ----  -
  0887 007200  COMMAND.COM    D8h 2Fh 2Eh 24h 23h
22h
  0A52 218144  INSPECT.EXE    FFh F9h F4h F3h F2h
F1h F0h EEh
  DBh DAh D9h
  EBh E8h E6h E2h E0h
  3Fh 00h

System Configuration Memory
  00 - 0F : 09 00 59 00 13 00 03 19 07 00 26
82 50 80 00 00

```

```

10 - 1F : 40 00 00 00 03 80 02 00 3C 00 00
00 00 00 00 02
20 - 2F : 00 00 00 00 7F 20 20 40 00 92 00
00 00 18 02 AC
30 - 3F : 00 3C 20 80 00 00 XX XX XX XX XX
XX XX XX XX

BIOS Data Area
40:0000 : F8 03 F8 02 00 00 00 00 78 03 00
00 00 00 00 9F
40:0010 : 27 44 00 7C 02 81 00 00 00 00 28
00 28 00 1B 01
40:0020 : 1B 01 1B 01 1B 01 1B 01 00 00 00
00 00 00 00 00
40:0030 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 01 01
40:0040 : 25 00 04 00 00 2B 01 0F 02 03 50
00 00 10 00 00
40:0050 : 00 18 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0060 : 0E 0D 00 D4 03 29 30 C2 11 85 76
00 F4 FB 0D 00
40:0070 : 00 00 00 12 00 01 00 00 14 14 14
14 01 01 01 01
40:0080 : 1E 00 3E 00 18 10 00 60 F9 11 0B
01 00 00 00 05
40:0090 : 17 00 00 00 2B 00 10 00 00 00 00
00 00 00 00 00
40:00A0 : 00 00 00 00 00 00 00 00 5C 13 00
C0 00 00 00 00
40:00B0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00C0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00D0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00E0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00F0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00

Interrupt Vector Table (including INSPECT)
00 - 03 : 0A62:0555 0070:0465
07BE:0016 0070:0465
04 - 07 : 0070:0465 F000:FF54
F000:4D64 F000:9BD0
08 - 0B : 07BE:001F 07BE:0028
F000:9BD0 F000:9BD0
0C - 0F : F000:9BD0 07BE:0082
07BE:009A 0070:0465
10 - 13 : C000:12DE F000:F84D
F000:F841 0070:03EE
14 - 17 : F000:DDBF 0204:0240
0070:042D F000:EF83
18 - 1B : F000:BOBB 087F:002F
F000:FE6E 0070:045F
1C - 1F : F000:FF53 F000:0000
0000:0522 C000:2019
20 - 23 : 00C9:0FA8 00C9:0FB2
0887:0314 0887:016D
24 - 27 : 0887:0178 00C9:0FBC
00C9:0FC6 00C9:0FD0

```

```

28 - 2B : 00C9:106F 0070:0466
00C9:106F 00C9:106F
2C - 2F : 00C9:106F 00C9:106F
0887:0162 0888:01CC
30 - 33 : C90F:E4EA F000:9B00
00C9:106F 00C9:106F
34 - 37 : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
38 - 3B : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
3C - 3F : 00C9:106F 00C9:106F
00C9:106F 1B1E:04F3
40 - 43 : F000:EC59 CC1F:01C6
F000:F065 C000:242C
44 - 47 : F000:9BD0 F000:9BD0
0000:0000 F000:9BD0
48 - 4B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
4C - 4F : F000:9BD0 F000:9BD0
F000:9BD0 0070:04FC
50 - 53 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
54 - 57 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
58 - 5B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
5C - 5F : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
60 - 63 : 0000:0000 0000:0000
0000:0000 0000:0000
64 - 67 : 0000:0000 0000:0000
0000:0000 0000:0000
68 - 6B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
6C - 6F : F000:9BD0 C000:12DE
F000:9BD0 F000:9BD0
70 - 73 : 07BE:0035 F000:9C1F
F000:9BD0 F000:9BD0
74 - 77 : 07BE:00E2 F000:9C21
07BE:00FA F000:9BD0
78 - 7B : 0000:0000 0000:0000
0000:0000 0000:0000
7C - 7F : 0000:0000 0000:0000
0000:0000 0000:0000
80 - 83 : 0000:0000 0000:0000
0000:0000 0000:0000
84 - 87 : 0000:0000 0000:0000
0000:0000 0000:0000
88 - 8B : 0000:0000 0000:0000
0000:0000 0000:0000
8C - 8F : 0000:0000 0000:0000
0000:0000 0000:0000
90 - 93 : 0000:0000 0000:0000
0000:0000 0000:0000
94 - 97 : 0000:0000 0000:0000
0000:0000 0000:0000
98 - 9B : 0000:0000 0000:0000
0000:0000 0000:0000
9C - 9F : 0000:0000 0000:0000
0000:0000 0000:0000
A0 - A3 : 0000:0000 0000:0000
0000:0000 0000:0000

```

```

A4 - A7 : 0000:0000 0000:0000
0000:0000 0000:0000
A8 - AB : 0000:0000 0000:0000
0000:0000 0000:0000
AC - AF : 0000:0000 0000:0000
0000:0000 0000:0000
B0 - B3 : 0000:0000 0000:0000
0000:0000 0000:0000
B4 - B7 : 0000:0000 0000:0000
0000:0000 0000:0000
B8 - BB : 0000:0000 0000:0000
0000:0000 0000:0000
BC - BF : 0000:0000 0000:0000
0000:0000 0000:0000
C0 - C3 : 0000:0000 0000:0000
0000:0000 0000:0000
C4 - C7 : 0000:0000 0000:0000
0000:0000 0000:0000
C8 - CB : 0000:0000 0000:0000
0000:0000 0000:0000
CC - CF : 0000:0000 0000:0000
0000:0000 0000:0000
D0 - D3 : 0000:0000 0000:0000
0000:0000 0000:0000
D4 - D7 : 54EE:0000 0000:4046
4B0C:F2EE 9804:BD0B
D8 - DB : 0200:6B3A 0001:F03A
2F00:0000 0001:F00B
DC - DF : F018:3400 0000:1E01
BB15:B1A0 0000:0118
E0 - E3 : 0118:C730 0000:0100
111A:8630 0000:0000
E4 - E7 : 0000:0030 0100:0000
2200:0000 0300:0003
E8 - EB : 1200:0000 00C8:00C8
0078:0CFD 0CFD:0078
EC - EF : 0000:1F07 0400:0006
1F85:0046 0087:0400
F0 - F3 : 13C1:0046 1D52:0013
13C1:D527 1400:1D52
F4 - F7 : 1BB0:0246 0101:7248
0000:0000 0000:5FFE
F8 - FB : 5FFE:0020 1497:6304
00DA:0003 0000:0991
FC - FF : 514B:00FC 0000:00F6
EFE2:0000 0246:F000

```

PCI Devices Information

```

Signature . . . . . PCI
Config Mechanism #1 . . . . . Supported
Config Mechanism #2 . . . . . Not Supported
Spec Cycle for Config #1 . . . . . Supported
Spec Cycle for Config #2 . . . . . Not Supported
BIOS Interface Version . . . . . 2.10
Last PCI Bus Number . . . . . 142
Number of PCI Devices . . . . . 12

PCI Bus Number . . . . . 0
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 7
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h

```

```

Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E77C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7600000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2000h
IO Address Length . . . . . 100h

```

```

PCI Bus Number . . . . . 0
Device Number . . . . . 2
Function Number . . . . . 00h
Slot Number . . . . . 8
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E75C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7400000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2400h
IO Address Length . . . . . 100h

```

```

PCI Bus Number . . . . . 0
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 9
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E73C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7200000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2800h
IO Address Length . . . . . 100h

```

```

PCI Bus Number . . . . . 0
Device Number . . . . . 13
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 1002h
Device ID . . . . . 4756h
Revision ID . . . . . 7Ah
Device Type . . . . . VGA Compatible
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFFE0000h
IRQ Line . . . . . 255

```

```

IRQ Pin . . . . . Not Used
Memory Address Base . . . . . E300000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2C00h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E71D0000h
Memory Address Length . . . . . 1000h

PCI Bus Number . . . . . 0
Device Number . . . . . 14
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 0E11h
Device ID . . . . . 0010h
Revision ID . . . . . 01h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF80000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
IO Address Base . . . . . 3000h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E6000000h
Memory Address Length . . . . . 1000000h
Memory Address Base . . . . . E5000000h
Memory Address Length . . . . . 1000000h

PCI Bus Number . . . . . 5
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 3
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1DC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1C00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 5
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 4
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1BC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1A00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4400h

```

```

IO Address Length . . . . . 100h

PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Revision ID . . . . . 00h
Device Type . . . . . Other Network
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h

```

```

Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

```

```

ProLiant is a trademark of Compaq Computer
Corporation.

Date . . . . . 07/19/2000
Time . . . . . 14:24:49

Product . . . . . ProLiant

Machine ID
From System Board . . . . . CPQ1608

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 8
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 7
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz

```



```

Slot . . . . . 6
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 5
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 4
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 3
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 2
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 1
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor(s) Mapped Out . . . . . None

Numeric Coprocessor . . . . . Integrated 387-
Compatible

Expansion Bus . . . . . ISA, PCI

System Identification Number . . D947BX71K002

CPU Mode . . . . . Real Mode

System ROM
Revision . . . . . 06/16/2000
Family . . . . . P42
Flashable . . . . . Yes
Supports F10 partition . . . . . Yes

Video Controller ROM
Revision . . . . . 3.96

Option ROMs
Address Range . . . . . C0000 - C7FFF
Data Dump . . . . . (1998/10/08 15:41)

Address Range . . . . . C8000 - CBFFF
Data Dump . . . . . (04/22/98 Maxwell
Smart Array Option ROM/BIOS (C)Co...

Address Range . . . . . CC000 - CFFFF

```

```

Data Dump . . . . . (04/22/98 ROC
ROC-LCR Option ROM/BIOS (C)Copyri...)

Address Range . . . . . E8000 - EDFFF
Data Dump . . . . . ( CPQSCSI d)

Bootblock ROM . . . . . 04/17/2000

Standby Recovery Server
Status . . . . . Disabled
COM Port . . . . . COM1
Server Configuration . . . . . Recovery
Timeout Value . . . . . 1 minutes

Memory Boards Identified:
System Board
DIMM Slot 1 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 2 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 3 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 4 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 5 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 6 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 7 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 8 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 9 . . . . . 0 Megabytes
DIMM Slot 10 . . . . . 0 Megabytes
DIMM Slot 11 . . . . . 0 Megabytes
DIMM Slot 12 . . . . . 0 Megabytes
DIMM Slot 13 . . . . . 0 Megabytes
DIMM Slot 14 . . . . . 0 Megabytes
DIMM Slot 15 . . . . . 0 Megabytes
DIMM Slot 16 . . . . . 0 Megabytes
Total Compaq Memory . . . . . 8192 Megabytes

Keyboard . . . . . Standard 11-Bit

LPT Ports . . . . . LPT1 (Address 378)

COM Ports . . . . . COM1 (Address 3F8)
COM2 (Address 2F8)

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F0000h
Memory Address Length . . . . . 100000h

```

```

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E0000h
Memory Address Length . . . . . 100000h

Other Network Controller
PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Subsystem Vendor ID . . . . . 10E4h
Subsystem ID . . . . . 0000h
Revision ID . . . . . 00h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F400000h
Memory Address Length . . . . . 200000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F000h

```

```

Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

Diskette Drive A . . . . . 1.44 Megabyte (3.5
inch)

Drive Controller 1, Compaq Integrated Smart Array
Controller
IDA Firmware Revision . . . . . 1.34
Array Accelerator Memory . . . . . 8188 Kbytes
Accelerator Status . . . . . Not Configured
Battery count . . . . . 0
Batteries charged . . . . . 0
Batteries failed . . . . . 0
Internal ProLiant . . . . . Bus 2, Rev. JB21

Logical Drive 1 . . . . . 9095 Megabyte
Fault Tolerance . . . . . Mirroring
OS Format . . . . . Multi-Sector
Distribution
Drive geometry (Cyl, Hds, Sec) 2177, 255, 32
Array Accelerator . . . . . Disabled

Hard Drive 1
SCSI Bus . . . . . 2
SCSI ID . . . . . 0
Serial Number . . . . . B3128353
Firmware Revision 1 . . . . . B016
Model Number . . . . . COMPAQ BD009122C6
Initialized for Monitoring . Yes
Reference time . . . . . 226771
Sectors read . . . . . *1062857344
Hard read errors . . . . . 0
Read errors retry . . . . . 0
ECC read errors . . . . . 0

```

```

Sectors written . . . . . 324100471
Hard write errors . . . . . 0
Write errors retry . . . . . 0
Seek count . . . . . 787256
Seek errors . . . . . 0
Spin cycles . . . . . 1
Spin up time . . . . . 0
Seek time track . . . . . 36%
Seek time third . . . . . 71%
Seek time full . . . . . 72%
Reallocated sectors . . . . . 68
Recovers read failed . . . . . 0
Bus faults . . . . . 0

Hard Drive 2
SCSI Bus . . . . . 2
SCSI ID . . . . . 1
Serial Number . . . . .
LS7485180000101929CF
Firmware Revision 1 . . . . . 3B07
Model Number . . . . . COMPAQ BD009122BA
Initialized for Monitoring . Yes
Reference time . . . . . 199520
Sectors read . . . . . *2087984254
Hard read errors . . . . . 0
Read errors retry . . . . . 0
ECC read errors . . . . . 0
Sectors written . . . . . 452020248
Hard write errors . . . . . 0
Write errors retry . . . . . 0
Seek count . . . . . 721208
Seek errors . . . . . 0
Spin cycles . . . . . 3
Spin up time . . . . . 0
Seek time track . . . . . 47%
Seek time third . . . . . 70%
Seek time full . . . . . 72%
Reallocated sectors . . . . . 1703
Recovers read failed . . . . . 0
Bus faults . . . . . 0

Graphics Mode . . . . . 03 (80-Column Text)

Primary Monitor attached to . . ATI RAGE IIC PCI
Graphics Controller
with Video Graphics Color Monitor

Base Memory
System Total . . . . . 636 Kbytes
Amount Free . . . . . 594 Kbytes
(609088 Bytes)

Extended Memory
System Total . . . . . 8387584 Kbytes

Expanded Memory
LIM Driver Support . . . . . LIM driver not
loaded

Operating System . . . . . MS-DOS version 7.00
(from diskette)

```

```

Environment variables
PATH=
PROMPT=$PSG
COMSPEC=A:\COMMAND.COM
CMDLINE=inspect /u
End of environment

Revisions Table

Previous Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A
Memory Board 1 Revision . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A

Current Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A
Memory Board 1 Revision . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A

System serial number . . . . . D947BX71K002

Memory Allocation (including INSPECT)
PSP SIZE NAME TRAPPED INTERRUPTS
-----
0887 007200 COMMAND.COM 2Fh 2Eh 24h 23h 22h
0A52 218144 INSPECT.EXE FFh F9h F3h F2h F1h
F0h EEh EDh EBh E4h 3Fh 00h

System Configuration Memory
00 - 0F : 01 00 25 00 14 00 03 19 07 00 26
82 50 80 00 00
10 - 1F : 40 00 00 00 03 80 02 00 3C 00 00
00 00 00 00 02
20 - 2F : 00 00 00 00 7F 20 20 40 00 92 00
00 00 18 02 AC
30 - 3F : 00 3C 20 80 00 00 XX XX XX XX XX
XX XX XX XX XX

BIOS Data Area
40:0000 : F8 03 F8 02 00 00 00 00 78 03 00
00 00 00 00 9F
40:0010 : 27 44 00 7C 02 00 00 00 00 00 1E
00 1E 00 00 00
40:0020 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0030 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 01 01
40:0040 : 25 00 04 00 00 2B 01 0F 02 03 50
00 00 10 00 00
40:0050 : 00 18 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0060 : 0E 0D 00 D4 03 29 30 C2 11 85 76
00 54 6A 0E 00
40:0070 : 00 00 00 12 00 01 00 00 14 14 14
14 01 01 01 01

```

```

40:0080 : 1E 00 3E 00 18 10 00 60 F9 11 0B
01 00 00 00 05
40:0090 : 17 00 00 00 2B 00 00 00 00 00 00
00 00 00 00 00
40:00A0 : 00 00 00 00 00 00 00 00 5C 13 00
C0 00 00 00 00
40:00B0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00C0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00D0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00E0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00F0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00

```

Interrupt Vector Table (including INSPECT)

```

00 - 03 : 0A62:0555 0070:0465
07BE:0016 0070:0465
04 - 07 : 0070:0465 F000:FF54
F000:4D64 F000:9BD0
08 - 0B : 07BE:001F 07BE:0028
F000:9BD0 F000:9BD0
0C - 0F : F000:9BD0 07BE:0082
07BE:009A 0070:0465
10 - 13 : C000:12DE F000:F84D
F000:F841 0070:03EE
14 - 17 : F000:DDBF 0204:0240
0070:042D F000:EF83
18 - 1B : F000:B0BB 087F:002F
F000:FE6E 0070:045F
1C - 1F : F000:FF53 F000:0000
0000:0522 C000:2019
20 - 23 : 00C9:0FA8 00C9:0FB2
0887:0314 0887:016D
24 - 27 : 0887:0178 00C9:0FBC
00C9:0FC6 00C9:0FD0
28 - 2B : 00C9:106F 0070:0466
00C9:106F 00C9:106F
2C - 2F : 00C9:106F 00C9:106F
0887:0162 0888:01CC
30 - 33 : C90F:E4EA F000:9B00
00C9:106F 00C9:106F
34 - 37 : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
38 - 3B : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
3C - 3F : 00C9:106F 00C9:106F
00C9:106F 1B1E:04F3
40 - 43 : F000:EC59 CC1F:01C6
F000:F065 C000:242C
44 - 47 : F000:9BD0 F000:9BD0
0000:0000 F000:9BD0
48 - 4B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
4C - 4F : F000:9BD0 F000:9BD0
F000:9BD0 0070:04FC
50 - 53 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
54 - 57 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0

```

```

58 - 5B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
5C - 5F : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
60 - 63 : 0000:0000 0000:0000
0000:0000 0000:0000
64 - 67 : 0000:0000 0000:0000
0000:0000 0000:0000
68 - 6B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
6C - 6F : F000:9BD0 C000:12DE
F000:9BD0 F000:9BD0
70 - 73 : 07BE:0035 F000:9C1F
F000:9BD0 F000:9BD0
74 - 77 : 07BE:00E2 F000:9C21
07BE:00FA F000:9BD0
78 - 7B : 0000:0000 0000:0000
0000:0000 0000:0000
7C - 7F : 0000:0000 0000:0000
0000:0000 0000:0000
80 - 83 : 0000:0000 0000:0000
0000:0000 0000:0000
84 - 87 : 0000:0000 0000:0000
0000:0000 0000:0000
88 - 8B : 0000:0000 0000:0000
0000:0000 0000:0000
8C - 8F : 0000:0000 0000:0000
0000:0000 0000:0000
90 - 93 : 0000:0000 0000:0000
0000:0000 0000:0000
94 - 97 : 0000:0000 0000:0000
0000:0000 0000:0000
98 - 9B : 0000:0000 0000:0000
0000:0000 0000:0000
9C - 9F : 0000:0000 0000:0000
0000:0000 0000:0000
A0 - A3 : 0000:0000 0000:0000
0000:0000 0000:0000
A4 - A7 : 0000:0000 0000:0000
0000:0000 0000:0000
A8 - AB : 0000:0000 0000:0000
0000:0000 0000:0000
AC - AF : 0000:0000 0000:0000
0000:0000 0000:0000
B0 - B3 : 0000:0000 0000:0000
0000:0000 0000:0000
B4 - B7 : 0000:0000 0000:0000
0000:0000 0000:0000
B8 - BB : 0000:0000 0000:0000
0000:0000 0000:0000
BC - BF : 0000:0000 0000:0000
0000:0000 0000:0000
C0 - C3 : 0000:0000 0000:0000
0000:0000 0000:0000
C4 - C7 : 0000:0000 0000:0000
0000:0000 0000:0000
C8 - CB : 0000:0000 0000:0000
0000:0000 0000:0000
CC - CF : 0000:0000 0000:0000
0000:0000 0000:0000
D0 - D3 : 0000:0000 0000:0000
0000:0000 0000:0000

```

```

D4 - D7 : 0000:0000 0000:0000
0000:0000 0000:0000
DB - DB : 0000:0000 0000:0000
0000:0000 0000:0000
DC - DF : 0000:0000 0000:0000
0000:0000 0000:0000
E0 - E3 : 0000:0000 0000:0000
0000:0000 00D8:0000
E4 - E7 : 0CFC:00D8 0000:000D
0020:0020 FFFF:0CFC
EB - EB : C0ED:0000 00C8:00C8
0078:0CFD 0CFC:0078
EC - EF : 0000:1F07 1F00:0006
1F85:0046 0087:1F00
F0 - F3 : 13C1:0046 1CA0:0013
13C1:D527 1400:1CA0
F4 - F7 : F000:4D30 03DA:0007
0000:0000 0000:7289
FB - FB : 5FFE:0020 1497:6304
0052:0003 0000:08DF
FC - FF : 514B:00FC 0000:00F6
EFE2:0000 0246:F000

```

PCI Devices Information

```

Signature . . . . . PCI
Config Mechanism #1 . . . . . Supported
Config Mechanism #2 . . . . . Not Supported
Spec Cycle for Config #1 . . . . . Supported
Spec Cycle for Config #2 . . . . . Not Supported
BIOS Interface Version . . . . . 2.10
Last PCI Bus Number . . . . . 142
Number of PCI Devices . . . . . 12

PCI Bus Number . . . . . 0
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 7
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E77C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7600000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 2
Function Number . . . . . 00h
Slot Number . . . . . 8
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10

```

```

IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E75C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7400000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 9
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E73C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7200000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2800h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 13
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 1002h
Device ID . . . . . 4756h
Revision ID . . . . . 7Ah
Device Type . . . . . VGA Compatible
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFFE0000h
IRQ Line . . . . . 255
IRQ Pin . . . . . Not Used
Memory Address Base . . . . . E3000000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2C00h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E71D0000h
Memory Address Length . . . . . 1000h

PCI Bus Number . . . . . 0
Device Number . . . . . 14
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 0E11h
Device ID . . . . . 0010h
Revision ID . . . . . 01h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF80000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
IO Address Base . . . . . 3000h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E6000000h
Memory Address Length . . . . . 100000h

```

```

Memory Address Base . . . . . E5000000h
Memory Address Length . . . . . 1000000h

PCI Bus Number . . . . . 5
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 3
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1DC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1C00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 5
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 4
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1BC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1A00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h

```

```

Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Revision ID . . . . . 00h
Device Type . . . . . Other Network
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller

```

```

Programming Interface . . . . . 00h
Expansion ROM Base Address . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

```

ProLiant is a trademark of Compaq Computer Corporation.

```

Date . . . . . 07/19/2000
Time . . . . . 14:10:09

```

```

Product . . . . . ProLiant

```

```

Machine ID
From System Board . . . . . CPQ1608

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 8
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 7
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 6
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 5
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 4
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 3
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz

```

```

Slot . . . . . 2
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 1
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor(s) Mapped Out . . . . . None

Numeric Coprocessor . . . . . Integrated 387-
Compatible

```

```

Expansion Bus . . . . . ISA, PCI

```

```

System Identification Number . . . D947BX71K005

```

```

CPU Mode . . . . . Real Mode

```

```

System ROM
Revision . . . . . 06/16/2000
Family . . . . . P42
Flashable . . . . . Yes
Supports F10 partition . . . . . Yes

```

```

Video Controller ROM
Revision . . . . . 3.96

```

```

Option ROMs
Address Range . . . . . C0000 - C7FFF
Data Dump . . . . . (1998/10/08 15:41)

Address Range . . . . . C8000 - CBFFF
Data Dump . . . . . (04/22/98 Maxwell
Smart Array Option ROM/BIOS (C)Co...)

```

```

Address Range . . . . . CC000 - CFFFF
Data Dump . . . . . (04/22/98 ROC
ROC-LCR Option ROM/BIOS (C)Copyri...)

```

```

Address Range . . . . . E8000 - EDFFF
Data Dump . . . . . ( CPQSCSI d)

```

```

Bootblock ROM . . . . . 04/17/2000

```

```

Standby Recovery Server
Status . . . . . Disabled
COM Port . . . . . COM1
Server Configuration . . . . . Recovery
Timeout Value . . . . . 1 minutes

```

```

Memory Boards Identified:
System Board
DIMM Slot 1 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 2 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 3 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 4 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 5 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 6 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 7 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 8 (SDRAM) . . . . . 1024 Megabytes

```

```

DIMM Slot 9 . . . . . 0 Megabytes
DIMM Slot 10 . . . . . 0 Megabytes
DIMM Slot 11 . . . . . 0 Megabytes
DIMM Slot 12 . . . . . 0 Megabytes
DIMM Slot 13 . . . . . 0 Megabytes
DIMM Slot 14 . . . . . 0 Megabytes
DIMM Slot 15 . . . . . 0 Megabytes
DIMM Slot 16 . . . . . 0 Megabytes
Total Compaq Memory . . . . . 8192 Megabytes

```

```

Keyboard . . . . . Enhanced

LPT Ports . . . . . LPT1 (Address 3BC)

COM Ports . . . . . COM1 (Address 3F8)
COM2 (Address 2F8)

```

```

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h

```

```

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h

```

```

Other Network Controller

```

```

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Subsystem Vendor ID . . . . . 10E4h
Subsystem ID . . . . . 0000h
Revision ID . . . . . 00h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7BF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7BE0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

```

```

Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B163h
Revision ID . . . . . 08h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7FF0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 40h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

```

```

Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B163h
Revision ID . . . . . 08h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9040h
IO Address Length . . . . . 40h
Memory Address Base . . . . . F7C00000h
Memory Address Length . . . . . 100000h

```

```

Diskette Drive A . . . . . 1.44 Megabyte (3.5
inch)

```

```

Drive Controller 1, Compaq Integrated Smart Array
Controller
IDA Firmware Revision . . . . . 1.34
Array Accelerator Memory . . . . . 8188 Kbytes
Accelerator Status . . . . . Not Configured
Battery count . . . . . 0
Batteries charged . . . . . 0
Batteries failed . . . . . 0
Internal ProLiant . . . . . Bus 2, Rev. JB21

Logical Drive 1 . . . . . 9095 Megabyte
Fault Tolerance . . . . . Mirroring
OS Format . . . . . Multi-Sector
Distribution
Drive geometry (Cyl, Hds, Sec) 2177, 255, 32
Array Accelerator . . . . . Disabled

```

```

Hard Drive 1
SCSI Bus . . . . . 2
SCSI ID . . . . . 0
Serial Number . . . . .
LS8108410000102118HE
Firmware Revision 1 . . . . . 3B07
Model Number . . . . . COMPAQ BD009122BA
Initialized for Monitoring . . . . . Yes
Reference time . . . . . 62446
Sectors read . . . . . *1330816303
Hard read errors . . . . . 0
Read errors retry . . . . . 0
ECC read errors . . . . . 0
Sectors written . . . . . 29227180
Hard write errors . . . . . 0
Write errors retry . . . . . 0
Seek count . . . . . 216528
Seek errors . . . . . 0
Spin cycles . . . . . 4
Spin up time . . . . . 0
Seek time track . . . . . 47%
Seek time third . . . . . 69%
Seek time full . . . . . 71%
Reallocated sectors . . . . . 557
Recovers read failed . . . . . 0
Bus faults . . . . . 0

```

```

Hard Drive 2
SCSI Bus . . . . . 2
SCSI ID . . . . . 1
Serial Number . . . . .
LS74775900001018JQT5
Firmware Revision 1 . . . . . 3B07
Model Number . . . . . COMPAQ BD009122BA
Initialized for Monitoring . . . . . Yes

```

```

Graphics Mode . . . . . 03 (80-Column Text)

```

```

Primary Monitor attached to . . ATI RAGE IIC PCI
Graphics Controller
with Video Graphics Color Monitor

```

```

Base Memory
System Total . . . . . 636 Kbytes
Amount Free . . . . . 594 Kbytes
(609088 Bytes)

```

```

Extended Memory
System Total . . . . . 8387584 Kbytes

```

```

Expanded Memory
LIM Driver Support . . . . . LIM driver not
loaded

```

```

Operating System . . . . . MS-DOS version 7.00
(from diskette)

```

```

Environment variables

```

```

PATH=
PROMPT=$PSG
COMSPEC=A:\COMMAND.COM
CMDLINE=inspect /u
End of environment

```

```

Revisions Table

```

```

Previous Revisions

```

```

I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . . . . A
Memory Board 1 Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . . . . A

```

```

Current Revisions

```

```

I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . . . . A
Memory Board 1 Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . . . . A

```

```

System serial number . . . . . D947BX71K005

```

```

Memory Allocation (including INSPECT)

```

```

PSP SIZE NAME TRAPPED INTERRUPTS
-----
0887 007200 COMMAND.COM 2Fh 2Eh 24h 23h 22h
0A52 218144 INSPECT.EXE FFh F9h F3h F2h F1h
F0h EEh EBh
E3h 3Fh 00h

```

```

System Configuration Memory

```

```

00 - 0F : 21 00 10 00 14 00 03 19 07 00 26
82 50 80 00 00
10 - 1F : 40 00 00 00 03 80 02 00 3C 00 00
00 00 00 00 02
20 - 2F : 00 00 00 00 7F 20 00 40 00 92 00
00 00 18 02 8C

```

30 - 3F : 00 3C 20 80 00 00 XX XX XX XX XX XX
XX XX XX XX XX

BIOS Data Area

40:0000 : F8 03 F8 02 00 00 00 00 BC 03 00
00 00 00 00 9F
40:0010 : 27 44 00 7C 02 00 00 00 00 00 1E
00 1E 00 00 00
40:0020 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0030 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 01 01
40:0040 : 25 00 04 00 00 2B 01 0F 02 03 50
00 00 10 00 00
40:0050 : 00 18 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0060 : 0E 0D 00 D4 03 29 30 C2 11 85 76
00 BE 2B 0E 00
40:0070 : 00 00 00 12 00 01 00 00 14 14 14
14 01 01 01 01
40:0080 : 1E 00 3E 00 18 10 00 60 F9 11 0B
01 00 00 00 05
40:0090 : 17 00 00 00 2B 00 10 00 00 00 00
00 00 00 00 00
40:00A0 : 00 00 00 00 00 00 00 00 5C 13 00
C0 00 00 00 00
40:00B0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00C0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00D0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00E0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00F0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00

Interrupt Vector Table (including INSPECT)

00 - 03 : 0A62:0555 0070:0465
07BE:0016 0070:0465
04 - 07 : 0070:0465 F000:FF54
F000:4D64 F000:9BD0
08 - 0B : 07BE:001F 07BE:0028
F000:9BD0 F000:9BD0
0C - 0F : F000:9BD0 F000:9BD0
07BE:009A 0070:0465
10 - 13 : C000:12DE F000:F84D
F000:F841 0070:03EE
14 - 17 : F000:DDBF 0204:0240
0070:042D F000:EF83
18 - 1B : F000:B0BB 087F:002F
F000:FE6E 0070:045F
1C - 1F : F000:FF53 F000:0000
0000:0522 C000:2019
20 - 23 : 00C9:0FA8 00C9:0FB2
0887:0314 0887:016D
24 - 27 : 0887:0178 00C9:0FBC
00C9:0FC6 00C9:0FD0
28 - 2B : 00C9:106F 0070:0466
00C9:106F 00C9:106F
2C - 2F : 00C9:106F 00C9:106F
0887:0162 0888:01CC

30 - 33 : C90F:E4EA F000:9BD0
00C9:106F 00C9:106F
34 - 37 : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
38 - 3B : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
3C - 3F : 00C9:106F 00C9:106F
00C9:106F 1B1E:04F3
40 - 43 : F000:EC59 CC1F:01C6
F000:F065 C000:242C
44 - 47 : F000:9BD0 F000:9BD0
0000:0000 F000:9BD0
48 - 4B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
4C - 4F : F000:9BD0 F000:9BD0
F000:9BD0 0070:04FC
50 - 53 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
54 - 57 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
58 - 5B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
5C - 5F : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
60 - 63 : 0000:0000 0000:0000
0000:0000 0000:0000
64 - 67 : 0000:0000 0000:0000
0000:0000 0000:0000
68 - 6B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
6C - 6F : F000:9BD0 C000:12DE
F000:9BD0 F000:9BD0
70 - 73 : 07BE:0035 F000:9C1F
F000:9BD0 07BE:00CA
74 - 77 : 07BE:00E2 F000:9C21
07BE:00FA F000:9BD0
78 - 7B : 0000:0000 0000:0000
0000:0000 0000:0000
7C - 7F : 0000:0000 0000:0000
0000:0000 0000:0000
80 - 83 : 0000:0000 0000:0000
0000:0000 0000:0000
84 - 87 : 0000:0000 0000:0000
0000:0000 0000:0000
88 - 8B : 0000:0000 0000:0000
0000:0000 0000:0000
8C - 8F : 0000:0000 0000:0000
0000:0000 0000:0000
90 - 93 : 0000:0000 0000:0000
0000:0000 0000:0000
94 - 97 : 0000:0000 0000:0000
0000:0000 0000:0000
98 - 9B : 0000:0000 0000:0000
0000:0000 0000:0000
9C - 9F : 0000:0000 0000:0000
0000:0000 0000:0000
A0 - A3 : 0000:0000 0000:0000
0000:0000 0000:0000
A4 - A7 : 0000:0000 0000:0000
0000:0000 0000:0000
A8 - AB : 0000:0000 0000:0000
0000:0000 0000:0000

AC - AF : 0000:0000 0000:0000
0000:0000 0000:0000
B0 - B3 : 0000:0000 0000:0000
0000:0000 0000:0000
B4 - B7 : 0000:0000 0000:0000
0000:0000 0000:0000
B8 - BB : 0000:0000 0000:0000
0000:0000 0000:0000
BC - BF : 0000:0000 0000:0000
0000:0000 0000:0000
C0 - C3 : 0000:0000 0000:0000
0000:0000 0000:0000
C4 - C7 : 0000:0000 0000:0000
0000:0000 0000:0000
C8 - CB : 0000:0000 0000:0000
0000:0000 0000:0000
CC - CF : 0000:0000 0000:0000
0000:0000 0000:0000
D0 - D3 : 0000:0000 0000:0000
0000:0000 0000:0000
D4 - D7 : 0000:0000 0000:0000
0000:0000 0000:0000
D8 - DB : 0000:0000 0000:0000
0000:0000 0000:0000
DC - DF : 0000:0000 0000:0000
0000:0000 0000:0000
E0 - E3 : 0000:0000 0000:0000
0008:0008 0E11:0CFC
E4 - E7 : C0ED:0000 0000:0809
0000:0400 0000:6E51
E8 - EB : 0008:0000 00C8:00C8
007B:0CFC 0CFC:0078
EC - EF : 0000:1F07 5300:0006
1F85:0046 0087:5300
F0 - F3 : 13C1:0046 1CE4:0013
13C1:D527 1400:1CE4
F4 - F7 : F000:4D30 03DA:0007
0000:0001 0000:7289
F8 - FB : 5FFE:0020 1497:6304
008C:0003 0000:0923
FC - FF : 514B:00FC 0000:00F6
EFE2:0000 0246:F000

PCI Devices Information

Signature PCI
Config Mechanism #1 Supported
Config Mechanism #2 Not Supported
Spec Cycle for Config #1 Supported
Spec Cycle for Config #2 Not Supported
BIOS Interface Version 2.10
Last PCI Bus Number 142
Number of PCI Devices 12

PCI Bus Number 0
Device Number 1
Function Number 00h
Slot Number 7
Vendor ID 0E11h
Device ID B060h
Revision ID 02h
Device Type RAID Controller
Programming Interface 00h
Expansion ROM Base Address FFF00000h

```

IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E77C000h
Memory Address Length . . . . . 4000h
Memory Address Base . . . . . E760000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 2
Function Number . . . . . 00h
Slot Number . . . . . 8
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF0000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E75C000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E740000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 9
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF0000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E73C000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E720000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2800h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 13
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 1002h
Device ID . . . . . 4756h
Revision ID . . . . . 7Ah
Device Type . . . . . VGA Compatible
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFFE000h
IRQ Line . . . . . 255
IRQ Pin . . . . . Not Used
Memory Address Base . . . . . E300000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2C00h

```

```

IO Address Length . . . . . 100h
Memory Address Base . . . . . E71D000h
Memory Address Length . . . . . 1000h

PCI Bus Number . . . . . 0
Device Number . . . . . 14
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 0E11h
Device ID . . . . . 0010h
Revision ID . . . . . 01h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF8000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
IO Address Base . . . . . 3000h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E600000h
Memory Address Length . . . . . 100000h
Memory Address Base . . . . . E500000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 5
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 3
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF0000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1DC000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1C0000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 5
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 4
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF0000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1BC000h
Memory Address Length . . . . . 4000h
Memory Address Base . . . . . F1A0000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 6
Device Number . . . . . 4

```

```

Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F0000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E0000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Revision ID . . . . . 00h
Device Type . . . . . Other Network
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7BF000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7BE000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F400000h
Memory Address Length . . . . . 200000h

PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 08h

```



```

Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7FF0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 40h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

```

```

PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 08h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9040h
IO Address Length . . . . . 40h
Memory Address Base . . . . . F7C00000h
Memory Address Length . . . . . 100000h

```

ProLiant is a trademark of Compaq Computer Corporation.

```

Date . . . . . 07/19/2000
Time . . . . . 13:59:43

```

```

Product . . . . . ProLiant

```

```

Machine ID
From System Board . . . . . CPQ1608

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 8
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 7
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 6
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 5
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 4
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 3
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 2
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 1
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

Processor(s) Mapped Out None

Numeric Coprocessor Integrated 387- Compatible

Expansion Bus ISA, PCI

System Identification Number D932BX71A010

CPU Mode Real Mode

```

System ROM
Revision . . . . . 06/16/2000
Family . . . . . P42
Flashable . . . . . Yes
Supports F10 partition . . . . . Yes

```

```

Video Controller ROM
Revision . . . . . 3.96

```

```

Option ROMs
Address Range . . . . . C0000 - C7FFF
Data Dump . . . . . (1998/10/08 15:41)

```

```

Address Range . . . . . C8000 - CBFFF
Data Dump . . . . . (04/22/98 Maxwell
Smart Array Option ROM/BIOS (C)Co...)

```

```

Address Range . . . . . CC000 - CFFFF
Data Dump . . . . . (04/22/98 ROC
ROC-LCR Option ROM/BIOS (C)Copyri...)

```

```

Address Range . . . . . E8000 - EDFFF

```

```

Data Dump . . . . . ( CPQSCSI d)
Bootblock ROM . . . . . 04/17/2000

```

```

Standby Recovery Server
Status . . . . . Disabled
COM Port . . . . . COM1
Server Configuration . . . . . Recovery
Timeout Value . . . . . 1 minutes

```

Memory Boards Identified:

```

System Board
DIMM Slot 1 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 2 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 3 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 4 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 5 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 6 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 7 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 8 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 9 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 10 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 11 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 12 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 13 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 14 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 15 (SDRAM) . . . . . 512 Megabytes
DIMM Slot 16 (SDRAM) . . . . . 512 Megabytes
Total Compaq Memory . . . . . 8192 Megabytes

```

Keyboard Enhanced

LPT Ports LPT1 (Address 378)

COM Ports COM1 (Address 3F8)
COM2 (Address 2F8)

```

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F38F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 5000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F3F00000h
Memory Address Length . . . . . 100000h

```

```

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller

```

```

PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F38E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 5020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F3E00000h
Memory Address Length . . . . . 100000h

```

Other Network Controller

```

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Subsystem Vendor ID . . . . . 10E4h
Subsystem ID . . . . . 0000h
Revision ID . . . . . 00h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

```

Compaq NC3131 Fast Ethernet NIC

```

Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7CF0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 6000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h

```

```

Memory Address Length . . . . . 100000h

```

Compaq NC3131 Fast Ethernet NIC

```

Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7CE0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 6020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

```

```

Diskette Drive A . . . . . 1.44 Megabyte (3.5
inch)

```

Drive Controller 1, Compaq Integrated Smart Array Controller

```

IDA Firmware Revision . . . . . 1.34
Array Accelerator Memory . . . . . 8188 Kbytes
Accelerator Status . . . . . Not Configured
Battery count . . . . . 0
Batteries charged . . . . . 0
Batteries failed . . . . . 0
Internal ProLiant . . . . . Bus 2, Rev. JB21
Logical Drive 1 . . . . . 9095 Megabyte
Fault Tolerance . . . . . Mirroring
OS Format . . . . . Multi-Sector

```

Distribution

```

Drive geometry (Cyl, Hds, Sec) 2177, 255, 32
Array Accelerator . . . . . Disabled

```

Hard Drive 1

```

SCSI Bus . . . . . 2
SCSI ID . . . . . 0
Serial Number . . . . .
LS811901000010211J7J
Firmware Revision 1 . . . . . 3B07
Model Number . . . . . COMPAQ BD009122BA
Initialized for Monitoring . . . . . Yes
Reference time . . . . . 95103
Sectors read . . . . . *852306254
Hard read errors . . . . . 0
Read errors retry . . . . . 0
ECC read errors . . . . . 0
Sectors written . . . . . 137129165
Hard write errors . . . . . 0
Write errors retry . . . . . 0

```

```

Seek count . . . . . 323488
Seek errors . . . . . 0
Spin cycles . . . . . 2
Spin up time . . . . . 0
Seek time track . . . . . 0%
Seek time third . . . . . 0%
Seek time full . . . . . 0%
Reallocated sectors . . . . . 804
Recovers read failed . . . . . 0
Bus faults . . . . . 0

```

Hard Drive 2

```

SCSI Bus . . . . . 2
SCSI ID . . . . . 1
Serial Number . . . . .
LJY5456100002004039Q
Firmware Revision 1 . . . . . 3B02
Model Number . . . . . COMPAQ BD00911934
Initialized for Monitoring . . . . . Yes
Reference time . . . . . 180380
Sectors read . . . . . *272475158
Hard read errors . . . . . 0
Read errors retry . . . . . 0
ECC read errors . . . . . 1
Sectors written . . . . . 330134724
Hard write errors . . . . . 0
Write errors retry . . . . . 0
Seek count . . . . . 647504
Seek errors . . . . . 0
Spin cycles . . . . . 4
Spin up time . . . . . 0
Seek time track . . . . . 22%
Seek time third . . . . . 69%
Seek time full . . . . . 71%
Reallocated sectors . . . . . 596
Recovers read failed . . . . . 0
Bus faults . . . . . 0

```

```

Graphics Mode . . . . . 03 (80-Column Text)

```

```

Primary Monitor attached to . . . . . ATI RAGE IIC PCI
Graphics Controller
with Video Graphics Color Monitor

```

Base Memory

```

System Total . . . . . 636 Kbytes
Amount Free . . . . . 594 Kbytes
(609088 Bytes)

```

Extended Memory

```

System Total . . . . . 8387584 Kbytes

```

Expanded Memory

```

LIM Driver Support . . . . . LIM driver not
loaded

```

```

Operating System . . . . . MS-DOS version 7.00
(from diskette)

```

Environment variables

```

PATH=
PROMPT=$P$G

```

```

COMSPEC=A:\COMMAND.COM
CMDLINE=inspect /u
End of environment

Revisions Table

Previous Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A
Memory Board 1 Revision . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A

```

```

Current Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A
Memory Board 1 Revision . . . 01
Assembly Version . . . . . 1
Functional Revision Level . . A

```

System serial number D932BX71A010

```

Memory Allocation (including INSPECT)
PSP  SIZE  NAME  TRAPPED INTERRUPTS
-----
0887 007200  COMMAND.COM  2Fh 2Eh 24h 23h 22h
0A52 218144  INSPECT.EXE  FFh F9h F0h EEh EDh
EBh E8h 3Fh

```

```

System Configuration Memory
00 - 0F : 55 00 59 00 13 00 03 19 07 00 26
82 50 80 00 00
10 - 1F : 40 00 00 00 03 80 02 00 3C 00 00
00 00 00 00 02
20 - 2F : 00 00 00 00 7F 20 00 40 00 92 00
00 00 18 02 8C
30 - 3F : 00 3C 20 80 00 00 XX XX XX XX XX XX
XX XX XX XX XX

```

```

BIOS Data Area
40:0000 : F8 03 F8 02 00 00 00 00 78 03 00
00 00 00 00 9F
40:0010 : 27 44 00 7C 02 81 00 00 00 00 20
00 22 00 1B 01
40:0020 : 1B 01 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0030 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 01 01
40:0040 : 25 00 04 00 00 2B 01 0F 02 03 50
00 00 10 00 00
40:0050 : 00 18 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0060 : 0E 0D 00 D4 03 29 30 C2 11 85 76
00 39 FF 0D 00
40:0070 : 00 00 00 12 00 01 00 00 14 14 14
14 01 01 01 01
40:0080 : 1E 00 3E 00 18 10 00 60 F9 11 0B
01 00 00 00 05

```

```

40:0090 : 17 00 00 00 2B 00 10 00 00 00 00
00 00 00 00 00
40:00A0 : 00 00 00 00 00 00 00 00 5C 13 00
C0 00 00 00 00
40:00B0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00C0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00D0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00E0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00F0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00

```

Interrupt Vector Table (including INSPECT)

```

00 - 03 : 0A62:0555 0070:0465
07BE:0016 0070:0465
04 - 07 : 0070:0465 F000:FF54
F000:4D64 F000:9BD0
08 - 0B : 07BE:001F 07BE:0028
F000:9BD0 F000:9BD0
0C - 0F : F000:9BD0 07BE:0082
07BE:009A 0070:0465
10 - 13 : C000:12DE F000:F84D
F000:F841 0070:03EE
14 - 17 : F000:DDBF 0204:0240
0070:042D F000:EF83
18 - 1B : F000:BOBB 087F:002F
F000:FE6E 0070:045F
1C - 1F : F000:FF53 F000:0000
0000:0522 C000:2019
20 - 23 : 00C9:0FA8 00C9:0FB2
0887:0314 0887:016D
24 - 27 : 0887:0178 00C9:0FBC
00C9:0FC6 00C9:0FD0
28 - 2B : 00C9:106F 0070:0466
00C9:106F 00C9:106F
2C - 2F : 00C9:106F 00C9:106F
0887:0162 0888:01CC
30 - 33 : C90F:E4EA F000:9B00
00C9:106F 00C9:106F
34 - 37 : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
38 - 3B : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
3C - 3F : 00C9:106F 00C9:106F
00C9:106F 1B1E:04F3
40 - 43 : F000:EC59 CC1F:01C6
F000:F065 C000:242C
44 - 47 : F000:9BD0 F000:9BD0
0000:0000 F000:9BD0
48 - 4B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
4C - 4F : F000:9BD0 F000:9BD0
F000:9BD0 0070:04FC
50 - 53 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
54 - 57 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
58 - 5B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0

```

```

5C - 5F : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
60 - 63 : 0000:0000 0000:0000
0000:0000 0000:0000
64 - 67 : 0000:0000 0000:0000
0000:0000 0000:0000
68 - 6B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
6C - 6F : F000:9BD0 C000:12DE
F000:9BD0 F000:9BD0
70 - 73 : 07BE:0035 F000:9C1F
F000:9BD0 F000:9BD0
74 - 77 : 07BE:00E2 F000:9C21
07BE:00FA F000:9BD0
78 - 7B : 0000:0000 0000:0000
0000:0000 0000:0000
7C - 7F : 0000:0000 0000:0000
0000:0000 0000:0000
80 - 83 : 0000:0000 0000:0000
0000:0000 0000:0000
84 - 87 : 0000:0000 0000:0000
0000:0000 0000:0000
88 - 8B : 0000:0000 0000:0000
0000:0000 0000:0000
8C - 8F : 0000:0000 0000:0000
0000:0000 0000:0000
90 - 93 : 0000:0000 0000:0000
0000:0000 0000:0000
94 - 97 : 0000:0000 0000:0000
0000:0000 0000:0000
98 - 9B : 0000:0000 0000:0000
0000:0000 0000:0000
9C - 9F : 0000:0000 0000:0000
0000:0000 0000:0000
A0 - A3 : 0000:0000 0000:0000
0000:0000 0000:0000
A4 - A7 : 0000:0000 0000:0000
0000:0000 0000:0000
A8 - AB : 0000:0000 0000:0000
0000:0000 0000:0000
AC - AF : 0000:0000 0000:0000
0000:0000 0000:0000
B0 - B3 : 0000:0000 0000:0000
0000:0000 0000:0000
B4 - B7 : 0000:0000 0000:0000
0000:0000 0000:0000
B8 - BB : 0000:0000 0000:0000
0000:0000 0000:0000
BC - BF : 0000:0000 0000:0000
0000:0000 0000:0000
C0 - C3 : 0000:0000 0000:0000
0000:0000 0000:0000
C4 - C7 : 0000:0000 0000:0000
0000:0000 0000:0000
C8 - CB : 0000:0000 0000:0000
0000:0000 0000:0000
CC - CF : 0000:0000 0000:0000
0000:0000 0000:0000
D0 - D3 : 0000:0000 0000:0000
0000:0000 0000:0000
D4 - D7 : 0000:0000 0000:0000
0000:0000 0000:0000

```

```

D8 - DB : 0000:0000 0000:0000
0000:0000 0000:0000
DC - DF : 0000:0000 0000:0000
0000:0000 0000:0000
E0 - E3 : 0000:0000 0000:0000
0000:0000 0000:0000
E4 - E7 : 0000:0000 0000:0000
0000:0000 F886:0212
E8 - EB : 34B0:0017 00C8:00C8
0078:0CFD 0CFC:0078
EC - EF : 0000:1F07 2E00:0006
1F85:0046 0087:2E00
F0 - F3 : 147F:0046 F000:4D30
0000:0007 C000:0000
F4 - F7 : C000:74F5 0101:0202
0000:0000 0000:5FFE
F8 - FB : 5FFE:0020 1497:6304
003A:0003 0000:0827
FC - FF : 514B:00FC 0000:00F6
EFE2:0000 0246:F000

```

PCI Devices Information

```

Signature . . . . . PCI
Config Mechanism #1 . . . . . Supported
Config Mechanism #2 . . . . . Not Supported
Spec Cycle for Config #1 . . . . . Supported
Spec Cycle for Config #2 . . . . . Not Supported
BIOS Interface Version . . . . . 2.10
Last PCI Bus Number . . . . . 16
Number of PCI Devices . . . . . 12

PCI Bus Number . . . . . 0
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 7
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F37C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F3600000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 2
Function Number . . . . . 00h
Slot Number . . . . . 8
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F35C0000h

```

```

Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F3400000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 9
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F33C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F3200000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2800h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 13
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 1002h
Device ID . . . . . 4756h
Revision ID . . . . . 7Ah
Device Type . . . . . VGA Compatible
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFFE0000h
IRQ Line . . . . . 255
IRQ Pin . . . . . Not Used
Memory Address Base . . . . . F0000000h
Memory Address Length . . . . . 1000000h
IO Address Base . . . . . 2C00h
IO Address Length . . . . . 100h
Memory Address Base . . . . . F31D0000h
Memory Address Length . . . . . 1000h

PCI Bus Number . . . . . 0
Device Number . . . . . 14
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 0E11h
Device ID . . . . . 0010h
Revision ID . . . . . 01h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF80000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
IO Address Base . . . . . 3000h
IO Address Length . . . . . 100h
Memory Address Base . . . . . F2000000h
Memory Address Length . . . . . 1000000h
Memory Address Base . . . . . F1000000h
Memory Address Length . . . . . 1000000h

```

```

PCI Bus Number . . . . . 5
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 3
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F3DC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F3C00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4000h
IO Address Length . . . . . 100h

```

```

PCI Bus Number . . . . . 5
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 4
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F3BC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F3A00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4400h
IO Address Length . . . . . 100h

```

```

PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F38F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 5000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F3F00000h
Memory Address Length . . . . . 100000h

```

```

PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h

```

```

Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F38E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 5020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F3E00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Revision ID . . . . . 00h
Device Type . . . . . Other Network
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7CF0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 6000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h

```

```

IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7CE0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 6020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

ProLiant is a trademark of Compaq Computer
Corporation.

Date . . . . . 07/19/2000
Time . . . . . 14:10:21

Product . . . . . ProLiant

Machine ID
From System Board . . . . . CPQ1608

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 8
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 7
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 6
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 5
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 4
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 3
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 2
Secondary Cache . . . . . 2048K

```

```

CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 1
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor(s) Mapped Out . . . . . None

Numeric Coprocessor . . . . . Integrated 387-
Compatible

Expansion Bus . . . . . ISA, PCI

System Identification Number . . . . . D934BX71A400

CPU Mode . . . . . Real Mode

System ROM
Revision . . . . . 06/16/2000
Family . . . . . P42
Flashable . . . . . Yes
Supports F10 partition . . . . . Yes

Video Controller ROM
Revision . . . . . 3.96

Option ROMs
Address Range . . . . . C0000 - C7FFF
Data Dump . . . . . (1998/10/08 15:41)

Address Range . . . . . C8000 - CBFFF
Data Dump . . . . . (04/22/98 Maxwell
Smart Array Option ROM/BIOS (C)Co...)

Address Range . . . . . CC000 - CFFFF
Data Dump . . . . . (04/22/98 ROC
ROC-LCR Option ROM/BIOS (C)Copyri...)

Address Range . . . . . E8000 - EDFFF
Data Dump . . . . . ( CPQSCSI d)

Bootblock ROM . . . . . 04/17/2000

Standby Recovery Server
Status . . . . . Disabled
COM Port . . . . . COM1
Server Configuration . . . . . Recovery
Timeout Value . . . . . 1 minutes

Memory Boards Identified:
System Board
DIMM Slot 1 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 2 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 3 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 4 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 5 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 6 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 7 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 8 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 9 . . . . . 0 Megabytes
DIMM Slot 10 . . . . . 0 Megabytes

```

```

DIMM Slot 11 . . . . . 0 Megabytes
DIMM Slot 12 . . . . . 0 Megabytes
DIMM Slot 13 . . . . . 0 Megabytes
DIMM Slot 14 . . . . . 0 Megabytes
DIMM Slot 15 . . . . . 0 Megabytes
DIMM Slot 16 . . . . . 0 Megabytes
Total Compaq Memory . . . . . 8192 Megabytes

Keyboard . . . . . Standard 11-Bit

LPT Ports . . . . . LPT1 (Address 3BC)

COM Ports . . . . . COM1 (Address 3F8)
                  COM2 (Address 2F8)

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h

Other Network Controller
PCI Bus Number . . . . . 13
Device Number . . . . . 1

```

```

Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Subsystem Vendor ID . . . . . 10E4h
Subsystem ID . . . . . 0000h
Revision ID . . . . . 00h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

```

```

Diskette Drive A . . . . . 1.44 Megabyte (3.5
inch)

Drive Controller 1, Compaq Integrated Smart Array
Controller
IDA Firmware Revision . . . . . 1.34
Array Accelerator Memory . . . . . 8188 Kbytes
Accelerator Status . . . . . Not Configured
Battery count . . . . . 0
Batteries charged . . . . . 0
Batteries failed . . . . . 0
Internal ProLiant . . . . . Bus 2, Rev. JB21

Logical Drive 1 . . . . . 9095 Megabyte
Fault Tolerance . . . . . Mirroring
OS Format . . . . . Multi-Sector
Distribution
Drive geometry (Cyl, Hds, Sec) 2177, 255, 32
Array Accelerator . . . . . Disabled

Hard Drive 1
SCSI Bus . . . . . 2
SCSI ID . . . . . 0
Serial Number . . . . .
LV148240000U008039C
Firmware Revision 1 . . . . . 3B02
Model Number . . . . . COMPAQ BD00911934
Initialized for Monitoring . . . . . Yes
Reference time . . . . . 182618
Sectors read . . . . . *1168777841
Hard read errors . . . . . 0
Read errors retry . . . . . 0
ECC read errors . . . . . 0
Sectors written . . . . . 320341253
Hard write errors . . . . . 0
Write errors retry . . . . . 0
Seek count . . . . . 637120
Seek errors . . . . . 2
Spin cycles . . . . . 0
Spin up time . . . . . 0
Seek time track . . . . . 22%
Seek time third . . . . . 69%
Seek time full . . . . . 71%
Reallocated sectors . . . . . 682
Recovers read failed . . . . . 0
Bus faults . . . . . 0

Hard Drive 2
SCSI Bus . . . . . 2
SCSI ID . . . . . 1
Serial Number . . . . .
LJX9045500001003JL9Y
Firmware Revision 1 . . . . . 3B02
Model Number . . . . . COMPAQ BD00911934
Initialized for Monitoring . . . . . Yes
Reference time . . . . . 194775
Sectors read . . . . . *572554979
Hard read errors . . . . . 0
Read errors retry . . . . . 0
ECC read errors . . . . . 0

```

```

Sectors written . . . . . 227506701
Hard write errors . . . . . 0
Write errors retry . . . . . 0
Seek count . . . . . 688510
Seek errors . . . . . 0
Spin cycles . . . . . 1
Spin up time . . . . . 0
Seek time track . . . . . 22%
Seek time third . . . . . 69%
Seek time full . . . . . 71%
Reallocated sectors . . . . . 278
Recovers read failed . . . . . 0
Bus faults . . . . . 0

Graphics Mode . . . . . 03 (80-Column Text)

Primary Monitor attached to . . ATI RAGE IIC PCI
Graphics Controller
with Video Graphics Color Monitor

Base Memory
System Total . . . . . 636 Kbytes
Amount Free . . . . . 594 Kbytes
(609088 Bytes)

Extended Memory
System Total . . . . . 8387584 Kbytes

Expanded Memory
LIM Driver Support . . . . . LIM driver not
loaded

Operating System . . . . . MS-DOS version 7.00
(from diskette)

Environment variables
PATH=
PROMPT=$PSG
COMSPEC=A:\COMMAND.COM
CMDLINE=inspect /u
End of environment

Revisions Table

Previous Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . A
Memory Board 1 Revision . . . 01
Assembly Version . . . . . 1
Functional Revision Level . A

Current Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . A
Memory Board 1 Revision . . . 01
Assembly Version . . . . . 1
Functional Revision Level . A

System serial number . . . . . D934BX71A400

```

```

Memory Allocation (including INSPECT)
PSP SIZE NAME TRAPPED INTERRUPTS
-----
0887 007200 COMMAND.COM 2Fh 2Eh 24h 23h 22h
0A52 218144 INSPECT.EXE FFh F9h F0h EEh EDh
EBh E3h 3Fh

System Configuration Memory
00 - 0F : 33 00 10 00 14 00 03 19 07 00 26
82 50 80 00 00
10 - 1F : 40 00 00 00 03 80 02 00 3C 00 00
00 00 00 00 02
20 - 2F : 00 00 00 00 7F 20 00 40 00 92 00
00 00 18 02 8C
30 - 3F : 00 3C 20 80 00 00 XX XX XX XX XX XX
XX XX XX XX XX

BIOS Data Area
40:0000 : F8 03 F8 02 00 00 00 00 BC 03 00
00 00 00 00 9F
40:0010 : 27 44 00 7C 02 00 00 00 00 00 1E
00 1E 00 00 00
40:0020 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0030 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 01 01
40:0040 : 25 00 04 00 00 2B 01 0F 02 03 50
00 00 10 00 00
40:0050 : 00 18 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0060 : 0E 0D 00 D4 03 29 30 C2 11 85 76
00 9B 2C 0E 00
40:0070 : 00 00 00 12 00 01 00 00 14 14 14
14 01 01 01 01
40:0080 : 1E 00 3E 00 18 10 00 60 F9 11 0B
01 00 00 00 05
40:0090 : 17 00 00 00 2B 00 00 00 00 00 00
00 00 00 00 00
40:00A0 : 00 00 00 00 00 00 00 00 5C 13 00
C0 00 00 00 00
40:00B0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00C0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00D0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00E0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00F0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00

Interrupt Vector Table (including INSPECT)
00 - 03 : 0A62:0555 0070:0465
07BE:0016 0070:0465
04 - 07 : 0070:0465 F000:FF54
F000:4D64 F000:9BD0
08 - 0B : 07BE:001F 07BE:0028
F000:9BD0 F000:9BD0
0C - 0F : F000:9BD0 F000:9BD0
07BE:009A 0070:0465

```

```

10 - 13 : C000:12DE F000:F84D
F000:F841 0070:03EE
14 - 17 : F000:DBDF 0204:0240
0070:042D F000:EF83
18 - 1B : F000:BOBB 087F:002F
F000:FE6E 0070:045F
1C - 1F : F000:FF53 F000:0000
0000:0522 C000:2019
20 - 23 : 00C9:0FA8 00C9:0FB2
0887:0314 0887:016D
24 - 27 : 0887:0178 00C9:0FBC
00C9:0FC6 00C9:0FDD
28 - 2B : 00C9:106F 0070:0466
00C9:106F 00C9:106F
2C - 2F : 00C9:106F 00C9:106F
0887:0162 0888:01CC
30 - 33 : C90F:E4EA F000:9B00
00C9:106F 00C9:106F
34 - 37 : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
38 - 3B : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
3C - 3F : 00C9:106F 00C9:106F
00C9:106F 1B1E:04F3
40 - 43 : F000:EC59 CC1F:01C6
F000:F065 C000:242C
44 - 47 : F000:9BD0 F000:9BD0
0000:0000 F000:9BD0
48 - 4B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
4C - 4F : F000:9BD0 F000:9BD0
F000:9BD0 0070:04FC
50 - 53 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
54 - 57 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
58 - 5B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
5C - 5F : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
60 - 63 : 0000:0000 0000:0000
0000:0000 0000:0000
64 - 67 : 0000:0000 0000:0000
0000:0000 0000:0000
68 - 6B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
6C - 6F : F000:9BD0 C000:12DE
F000:9BD0 F000:9BD0
70 - 73 : 07BE:0035 F000:9C1F
F000:9BD0 07BE:00CA
74 - 77 : 07BE:00E2 F000:9C21
07BE:00FA F000:9BD0
78 - 7B : 0000:0000 0000:0000
0000:0000 0000:0000
7C - 7F : 0000:0000 0000:0000
0000:0000 0000:0000
80 - 83 : 0000:0000 0000:0000
0000:0000 0000:0000
84 - 87 : 0000:0000 0000:0000
0000:0000 0000:0000
88 - 8B : 0000:0000 0000:0000
0000:0000 0000:0000

```

```

8C - 8F : 0000:0000 0000:0000
0000:0000 0000:0000
90 - 93 : 0000:0000 0000:0000
0000:0000 0000:0000
94 - 97 : 0000:0000 0000:0000
0000:0000 0000:0000
98 - 9B : 0000:0000 0000:0000
0000:0000 0000:0000
9C - 9F : 0000:0000 0000:0000
0000:0000 0000:0000
A0 - A3 : 0000:0000 0000:0000
0000:0000 0000:0000
A4 - A7 : 0000:0000 0000:0000
0000:0000 0000:0000
A8 - AB : 0000:0000 0000:0000
0000:0000 0000:0000
AC - AF : 0000:0000 0000:0000
0000:0000 0000:0000
B0 - B3 : 0000:0000 0000:0000
0000:0000 0000:0000
B4 - B7 : 0000:0000 0000:0000
0000:0000 0000:0000
B8 - BB : 0000:0000 0000:0000
0000:0000 0000:0000
BC - BF : 0000:0000 0000:0000
0000:0000 0000:0000
C0 - C3 : 0000:0000 0000:0000
0000:0000 0000:0000
C4 - C7 : 0000:0000 0000:0000
0000:0000 0000:0000
C8 - CB : 0000:0000 0000:0000
0000:0000 0000:0000
CC - CF : 0000:0000 0000:0000
0000:0000 0000:0000
D0 - D3 : 0000:0000 0000:0000
0000:0000 0000:0000
D4 - D7 : 0000:0000 0000:0000
0000:0000 0000:0000
D8 - DB : 0000:0000 0000:0000
0000:0000 0000:0000
DC - DF : 0000:0000 0000:0000
0000:0000 0000:0000
E0 - E3 : 0000:0000 0000:0000
0008:0008 0E11:0CFC
E4 - E7 : 0000:0000 0000:0809
0000:0400 0000:6E51
E8 - EB : 0008:0000 00C8:00C8
0078:0CFD 0CFC:0078
EC - EF : 0000:1F07 2C00:0006
1F85:0046 0087:2C00
F0 - F3 : 13C1:0046 F000:4D30
0000:0007 C000:0000
F4 - F7 : C000:74F3 0101:0202
0000:0000 0000:5FFE
FB - FE : 5FFE:0020 1497:6304
00F0:0003 0000:0923
FC - FF : 514B:00FC 0000:00F6
EFE2:0000 0246:F000

PCI Devices Information
Signature . . . . . PCI
Config Mechanism #1 . . . . . Supported
Config Mechanism #2 . . . . . Not Supported

```

```

Spec Cycle for Config #1 . . . . Supported
Spec Cycle for Config #2 . . . . Not Supported
BIOS Interface Version . . . . . 2.10
Last PCI Bus Number . . . . . 142
Number of PCI Devices . . . . . 12

PCI Bus Number . . . . . 0
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 7
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E77C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7600000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 2
Function Number . . . . . 00h
Slot Number . . . . . 8
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E75C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7400000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 9
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . FFF00000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E73C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7200000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2800h
IO Address Length . . . . . 100h

```

```

PCI Bus Number . . . . . 0
Device Number . . . . . 13
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 1002h
Device ID . . . . . 4756h
Revision ID . . . . . 7Ah
Device Type . . . . . VGA Compatible
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . FFFE0000h
IRQ Line . . . . . 255
IRQ Pin . . . . . Not Used
Memory Address Base . . . . . E3000000h
Memory Address Length . . . . . 1000000h
IO Address Base . . . . . 2C00h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E71D0000h
Memory Address Length . . . . . 1000h

PCI Bus Number . . . . . 0
Device Number . . . . . 14
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 0E11h
Device ID . . . . . 0010h
Revision ID . . . . . 01h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . FFF80000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
IO Address Base . . . . . 3000h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E6000000h
Memory Address Length . . . . . 1000000h
Memory Address Base . . . . . E5000000h
Memory Address Length . . . . . 1000000h

PCI Bus Number . . . . . 5
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 3
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1DC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1C00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 5
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 4
Vendor ID . . . . . 0E11h

```



```

Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1BC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1A00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Revision ID . . . . . 00h
Device Type . . . . . Other Network
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h

```

```

IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

ProLiant is a trademark of Compaq Computer
Corporation.

Date . . . . . 07/19/2000
Time . . . . . 14:30:07

Product . . . . . ProLiant

Machine ID
From System Board . . . . . CPQ1608

Processor . . . . . Pentium III(R) Xeon
at 700 MHz

```

```

Slot . . . . . 8
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A1

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 7
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A1

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 6
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A1

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 5
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A1

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 4
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A1

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 3
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A1

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 2
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A1

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 1
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A1

Processor(s) Mapped Out . . . . . None

Numeric Coprocessor . . . . . Integrated 387-
Compatible

Expansion Bus . . . . . ISA, PCI

System Identification Number . . . . . D947BX71K030

CPU Mode . . . . . Real Mode

System ROM
Revision . . . . . 06/16/2000
Family . . . . . P42
Flashable . . . . . Yes
Supports F10 partition . . . . . Yes

```

```

Video Controller ROM
  Revision . . . . . 3.96

Option ROMs
  Address Range . . . . . C0000 - C7FFF
  Data Dump . . . . . (1998/10/08 15:41)

  Address Range . . . . . C8000 - CBFFF
  Data Dump . . . . . (04/22/98 Maxwell)
Smart Array Option ROM/BIOS (C)Co...
  Address Range . . . . . CC000 - CFFFF
  Data Dump . . . . . (04/22/98 ROC
ROC-LCR Option ROM/BIOS (C)Copyri...
  Address Range . . . . . E8000 - EDFFF
  Data Dump . . . . . ( CPQSCSI d)

Bootblock ROM . . . . . 04/17/2000

Standby Recovery Server
  Status . . . . . Disabled
  COM Port . . . . . COM1
  Server Configuration . . . . . Recovery
  Timeout Value . . . . . 1 minutes

Memory Boards Identified:
  System Board
  DIMM Slot 1 (SDRAM) . . . . . 1024 Megabytes
  DIMM Slot 2 (SDRAM) . . . . . 1024 Megabytes
  DIMM Slot 3 (SDRAM) . . . . . 1024 Megabytes
  DIMM Slot 4 (SDRAM) . . . . . 1024 Megabytes
  DIMM Slot 5 (SDRAM) . . . . . 1024 Megabytes
  DIMM Slot 6 (SDRAM) . . . . . 1024 Megabytes
  DIMM Slot 7 (SDRAM) . . . . . 1024 Megabytes
  DIMM Slot 8 (SDRAM) . . . . . 1024 Megabytes
  DIMM Slot 9 . . . . . 0 Megabytes
  DIMM Slot 10 . . . . . 0 Megabytes
  DIMM Slot 11 . . . . . 0 Megabytes
  DIMM Slot 12 . . . . . 0 Megabytes
  DIMM Slot 13 . . . . . 0 Megabytes
  DIMM Slot 14 . . . . . 0 Megabytes
  DIMM Slot 15 . . . . . 0 Megabytes
  DIMM Slot 16 . . . . . 0 Megabytes
  Total Compaq Memory . . . . . 8192 Megabytes

Keyboard . . . . . Standard 11-Bit

LPT Ports . . . . . LPT1 (Address 3BC)

COM Ports . . . . . COM1 (Address 3F8)
  COM2 (Address 2F8)

Compaq NC3131 Fast Ethernet NIC
  Device Type . . . . . Ethernet Controller
  PCI Bus Number . . . . . 6
  Device Number . . . . . 4
  Function Number . . . . . 00h
  Slot Number . . . . . 1
  Vendor ID . . . . . 0E11h
  Device ID . . . . . 1229h
  Subsystem Vendor ID . . . . . 0E11h
  Subsystem ID . . . . . B0DDh
  Revision ID . . . . . 05h
  Programming Interface . . . . . 00h
  Expansion ROM Base Address . . . . . 0h
  IRQ Line . . . . . 15
  IRQ Pin . . . . . INTA#
  Memory Address Base . . . . . EF2E0000h
  Memory Address Length . . . . . 1000h
  IO Address Base . . . . . 9020h
  IO Address Length . . . . . 20h
  Memory Address Base . . . . . F1E00000h
  Memory Address Length . . . . . 100000h

Compaq NC3131 Fast Ethernet NIC
  Device Type . . . . . Ethernet Controller
  PCI Bus Number . . . . . 6
  Device Number . . . . . 4
  Function Number . . . . . 00h
  Slot Number . . . . . 1
  Vendor ID . . . . . 0E11h
  Device ID . . . . . 1229h
  Subsystem Vendor ID . . . . . 0E11h
  Subsystem ID . . . . . B0DDh
  Revision ID . . . . . 05h

```

```

Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF2F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h

Compaq NC3131 Fast Ethernet NIC
  Device Type . . . . . Ethernet Controller
  PCI Bus Number . . . . . 6
  Device Number . . . . . 5
  Function Number . . . . . 00h
  Slot Number . . . . . 1
  Vendor ID . . . . . 0E11h
  Device ID . . . . . 1229h
  Subsystem Vendor ID . . . . . 0E11h
  Subsystem ID . . . . . B0DDh
  Revision ID . . . . . 05h
  Programming Interface . . . . . 00h
  Expansion ROM Base Address . . . . . 0h
  IRQ Line . . . . . 15
  IRQ Pin . . . . . INTA#
  Memory Address Base . . . . . EF2E0000h
  Memory Address Length . . . . . 1000h
  IO Address Base . . . . . 9020h
  IO Address Length . . . . . 20h
  Memory Address Base . . . . . F1E00000h
  Memory Address Length . . . . . 100000h

Other Network Controller
  PCI Bus Number . . . . . 13
  Device Number . . . . . 1
  Function Number . . . . . 00h
  Slot Number . . . . . 10
  Vendor ID . . . . . 10E4h
  Device ID . . . . . 0005h
  Subsystem Vendor ID . . . . . 10E4h
  Subsystem ID . . . . . 0000h
  Revision ID . . . . . 00h
  Programming Interface . . . . . 00h
  Expansion ROM Base Address . . . . . 0h
  IRQ Line . . . . . 15
  IRQ Pin . . . . . INTA#
  Memory Address Base . . . . . F7DF0000h
  Memory Address Length . . . . . 1000h
  Memory Address Base . . . . . F7DE0000h
  Memory Address Length . . . . . 10000h
  Memory Address Base . . . . . F4000000h
  Memory Address Length . . . . . 2000000h

Compaq NC3131 Fast Ethernet NIC
  Device Type . . . . . Ethernet Controller
  PCI Bus Number . . . . . 14
  Device Number . . . . . 4
  Function Number . . . . . 00h
  Slot Number . . . . . 11

```

```

Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . A000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

Compaq NC3131 Fast Ethernet NIC
  Device Type . . . . . Ethernet Controller
  PCI Bus Number . . . . . 14
  Device Number . . . . . 5
  Function Number . . . . . 00h
  Slot Number . . . . . 11
  Vendor ID . . . . . 0E11h
  Device ID . . . . . 1229h
  Subsystem Vendor ID . . . . . 0E11h
  Subsystem ID . . . . . B0DDh
  Revision ID . . . . . 05h
  Programming Interface . . . . . 00h
  Expansion ROM Base Address . . . . . 0h
  IRQ Line . . . . . 15
  IRQ Pin . . . . . INTA#
  Memory Address Base . . . . . F74E0000h
  Memory Address Length . . . . . 1000h
  IO Address Base . . . . . A020h
  IO Address Length . . . . . 20h
  Memory Address Base . . . . . F7E00000h
  Memory Address Length . . . . . 100000h

Diskette Drive A . . . . . 1.44 Megabyte (3.5
inch)

Drive Controller 1, Compaq Integrated Smart Array
Controller
  IDA Firmware Revision . . . . . 1.34
  Array Accelerator Memory . . . . . 8188 Kbytes
  Accelerator Status . . . . . Not Configured
  Battery count . . . . . 0
  Batteries charged . . . . . 0
  Batteries failed . . . . . 0
  Internal ProLiant . . . . . Bus 2, Rev. JB21

  Logical Drive 1 . . . . . 9095 Megabyte
  Fault Tolerance . . . . . Mirroring
  OS Format . . . . . Multi-Sector

Distribution
  Drive geometry (Cyl, Hds, Sec) 2177, 255, 32
  Array Accelerator . . . . . Disabled

  Hard Drive 1
  SCSI Bus . . . . . 2

```

```

SCSI ID . . . . . 0
Serial Number . . . . .
LS760571000010200NGK
Firmware Revision 1 . . . . . 3B07
Model Number . . . . . COMPAQ BD009122BA
Initialized for Monitoring . Yes
Reference time . . . . . 223533
Sectors read . . . . . *1110297524
Hard read errors . . . . . 0
Read errors retry . . . . . 0
ECC read errors . . . . . 0
Sectors written . . . . . 317644192
Hard write errors . . . . . 0
Write errors retry . . . . . 0
Seek count . . . . . 765714
Seek errors . . . . . 0
Spin cycles . . . . . 1
Spin up time . . . . . 0
Seek time track . . . . . 47%
Seek time third . . . . . 70%
Seek time full . . . . . 71%
Reallocated sectors . . . . . 364
Recovers read failed . . . . . 0
Bus faults . . . . . 0

Hard Drive 2
SCSI Bus . . . . . 2
SCSI ID . . . . . 1
Serial Number . . . . .
LSD61741000W041BBQ
Firmware Revision 1 . . . . . 3B09
Model Number . . . . . COMPAQ BD009122BA
Initialized for Monitoring . No

Graphics Mode . . . . . 03 (80-Column Text)

Primary Monitor attached to . . ATI RAGE IIC PCI
Graphics Controller
with Video Graphics Color Monitor

Base Memory
System Total . . . . . 636 Kbytes
Amount Free . . . . . 594 Kbytes
(609088 Bytes)

Extended Memory
System Total . . . . . 8387584 Kbytes

Expanded Memory
LIM Driver Support . . . . . LIM driver not
loaded

Operating System . . . . . MS-DOS version 7.00
(from diskette)

Environment variables
PATH=
PROMPT=$PSG
COMSPEC=A:\COMMAND.COM
CMDLINE=inspect /u
End of environment

```

```

Revisions Table

Previous Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . A
Memory Board 1 Revision . . . 01
Assembly Version . . . . . 1
Functional Revision Level . A

Current Revisions
I/O Board Revision . . . . . 01
Assembly Version . . . . . 1
Functional Revision Level . A
Memory Board 1 Revision . . . 01
Assembly Version . . . . . 1
Functional Revision Level . A

System serial number . . . . . D947BX71K030

Memory Allocation (including INSPECT)
PSP SIZE NAME TRAPPED INTERRUPTS
-----
0887 007200 COMMAND.COM 2Fh 2Eh 24h 23h 22h
0A52 218144 INSPECT.EXE Ffh F9h F4h F3h F2h
Flh F0h EEh EDh EBh E3h 3Fh 00h

System Configuration Memory
00 - 0F : 20 00 30 00 14 00 03 19 07 00 26
82 50 80 00 00
10 - 1F : 40 00 00 00 03 80 02 00 3C 00 00
00 00 00 00 02
20 - 2F : 00 00 00 00 7F 20 00 40 00 92 00
00 00 18 02 8C
30 - 3F : 00 3C 20 80 00 00 XX XX XX XX XX XX XX
XX XX XX XX XX

BIOS Data Area
40:0000 : F8 03 F8 02 00 00 00 00 BC 03 00
00 00 00 00 9F
40:0010 : 27 44 00 7C 02 00 00 00 00 00 1E
00 1E 00 00 00
40:0020 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0030 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 01 01
40:0040 : 25 00 04 00 00 2B 01 0F 02 03 50
00 00 10 00 00
40:0050 : 00 18 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0060 : 0E 0D 00 D4 03 29 30 C2 11 85 76
00 F1 80 0E 00
40:0070 : 00 00 00 12 00 01 00 00 14 14 14
14 01 01 01 01
40:0080 : 1E 00 3E 00 18 10 00 60 F9 11 0B
01 00 00 00 05
40:0090 : 17 00 00 00 2B 00 00 00 00 00 00
00 00 00 00 00
40:00A0 : 00 00 00 00 00 00 00 00 5C 13 00
C0 00 00 00 00

```

```

40:00B0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00C0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00D0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00E0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00F0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00

Interrupt Vector Table (including INSPECT)
00 - 03 : 0A62:0555 0070:0465
07BE:0016 0070:0465
04 - 07 : 0070:0465 F000:FF54
F000:4D64 F000:9BD0
08 - 0B : 07BE:001F 07BE:0028
F000:9BD0 F000:9BD0
0C - 0F : F000:9BD0 F000:9BD0
07BE:009A 0070:0465
10 - 13 : C000:12DE F000:F84D
F000:F841 0070:03EE
14 - 17 : F000:DDBF 0204:0240
0070:042D F000:EF83
18 - 1B : F000:BOBB 087F:002F
F000:FE6E 0070:045F
1C - 1F : F000:FF53 F000:0000
0000:0522 C000:2019
20 - 23 : 00C9:0FA8 00C9:0FB2
0887:0314 0887:016D
24 - 27 : 0887:0178 00C9:0FBC
00C9:0FC6 00C9:0FD0
28 - 2B : 00C9:106F 0070:0466
00C9:106F 00C9:106F
2C - 2F : 00C9:106F 00C9:106F
0887:0162 0888:01CC
30 - 33 : C90F:E4EA F000:9B00
00C9:106F 00C9:106F
34 - 37 : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
38 - 3B : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
3C - 3F : 00C9:106F 00C9:106F
00C9:106F 1B1E:04F3
40 - 43 : F000:EC59 CC1F:01C6
F000:F065 C000:242C
44 - 47 : F000:9BD0 F000:9BD0
0000:0000 F000:9BD0
48 - 4B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
4C - 4F : F000:9BD0 F000:9BD0
F000:9BD0 0070:04FC
50 - 53 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
54 - 57 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
58 - 5B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
5C - 5F : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
60 - 63 : 0000:0000 0000:0000
0000:0000 0000:0000

```

```

64 - 67 : 0000:0000 0000:0000
0000:0000 0000:0000
68 - 6B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
6C - 6F : F000:9BD0 C000:12DE
F000:9BD0 F000:9BD0
70 - 73 : 07BE:0035 F000:9C1F
F000:9BD0 07BE:00CA
74 - 77 : 07BE:00E2 F000:9C21
07BE:00FA F000:9BD0
78 - 7B : 0000:0000 0000:0000
0000:0000 0000:0000
7C - 7F : 0000:0000 0000:0000
0000:0000 0000:0000
80 - 83 : 0000:0000 0000:0000
0000:0000 0000:0000
84 - 87 : 0000:0000 0000:0000
0000:0000 0000:0000
88 - 8B : 0000:0000 0000:0000
0000:0000 0000:0000
8C - 8F : 0000:0000 0000:0000
0000:0000 0000:0000
90 - 93 : 0000:0000 0000:0000
0000:0000 0000:0000
94 - 97 : 0000:0000 0000:0000
0000:0000 0000:0000
98 - 9B : 0000:0000 0000:0000
0000:0000 0000:0000
9C - 9F : 0000:0000 0000:0000
0000:0000 0000:0000
A0 - A3 : 0000:0000 0000:0000
0000:0000 0000:0000
A4 - A7 : 0000:0000 0000:0000
0000:0000 0000:0000
A8 - AB : 0000:0000 0000:0000
0000:0000 0000:0000
AC - AF : 0000:0000 0000:0000
0000:0000 0000:0000
B0 - B3 : 0000:0000 0000:0000
0000:0000 0000:0000
B4 - B7 : 0000:0000 0000:0000
0000:0000 0000:0000
B8 - BB : 0000:0000 0000:0000
0000:0000 0000:0000
BC - BF : 0000:0000 0000:0000
0000:0000 0000:0000
C0 - C3 : 0000:0000 0000:0000
0000:0000 0000:0000
C4 - C7 : 0000:0000 0000:0000
0000:0000 0000:0000
C8 - CB : 0000:0000 0000:0000
0000:0000 0000:0000
CC - CF : 0000:0000 0000:0000
0000:0000 0000:0000
D0 - D3 : 0000:0000 0000:0000
0000:0000 0000:0000
D4 - D7 : 0000:0000 0000:0000
0000:0000 0000:0000
D8 - DB : 0000:0000 0000:0000
0000:0000 0000:0000
DC - DF : 0000:0000 0000:0000
0000:0000 0000:0000

```

```

E0 - E3 : 0000:0000 0000:0000
0008:0008 0E11:0CFC
E4 - E7 : COED:0000 0000:0809
0000:0400 0000:6E51
E8 - EB : 0008:0000 00C8:00C8
0078:0CFD 0CFC:0078
EC - EF : 0000:1F07 1800:0006
1F85:0046 0087:1800
F0 - F3 : 13C1:0046 1CDB:0013
13C1:D527 1400:1CDB
F4 - F7 : 1BB0:0246 0101:7248
0000:0000 0000:5FFE
F8 - FB : 5FFE:0020 1497:6304
0036:0003 0000:091A
FC - FF : 514B:00FC 0000:00F6
EFE2:0000 0246:F000

```

PCI Devices Information

```

Signature . . . . . PCI
Config Mechanism #1 . . . . . Supported
Config Mechanism #2 . . . . . Not Supported
Spec Cycle for Config #1 . . . . . Supported
Spec Cycle for Config #2 . . . . . Not Supported
BIOS Interface Version . . . . . 2.10
Last PCI Bus Number . . . . . 146
Number of PCI Devices . . . . . 11

PCI Bus Number . . . . . 0
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 7
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E51C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E5000000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 2
Function Number . . . . . 00h
Slot Number . . . . . 8
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E4FC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E4E00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2400h

```

```

IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 9
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E4DC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E4C00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2800h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 13
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 1002h
Device ID . . . . . 4756h
Revision ID . . . . . 7Ah
Device Type . . . . . VGA Compatible
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFEE0000h
IRQ Line . . . . . 255
IRQ Pin . . . . . Not Used
Memory Address Base . . . . . E0000000h
Memory Address Length . . . . . 1000000h
IO Address Base . . . . . 2C00h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E4BD0000h
Memory Address Length . . . . . 1000h

PCI Bus Number . . . . . 0
Device Number . . . . . 14
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 0E11h
Device ID . . . . . 0010h
Revision ID . . . . . 01h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF80000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
IO Address Base . . . . . 3000h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E3000000h
Memory Address Length . . . . . 1000000h
Memory Address Base . . . . . E2000000h
Memory Address Length . . . . . 1000000h

PCI Bus Number . . . . . 5
Device Number . . . . . 4
Function Number . . . . . 00h

```

```

Slot Number . . . . . 4
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1DC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1C00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF2F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF2E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Revision ID . . . . . 00h
Device Type . . . . . Other Network
Controller

```

```

Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . A000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . A020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

ProLiant is a trademark of Compaq Computer
Corporation.

Date . . . . . 07/19/2000
Time . . . . . 14:30:19

Product . . . . . ProLiant

Machine ID
From System Board . . . . . CPQ1608

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 8
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 7
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 6
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 5
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 4
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 3
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 2
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 1
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor(s) Mapped Out . . . . . None

Numeric Coprocessor . . . . . Integrated 387-
Compatible

Expansion Bus . . . . . ISA, PCI

System Identification Number . . . . . D947BX71K118

CPU Mode . . . . . Real Mode

System ROM
Revision . . . . . 06/16/2000
Family . . . . . P42
Flashable . . . . . Yes

```

```

Supports F10 partition . . . . Yes
Video Controller ROM
Revision . . . . . 3.96
Option ROMs
Address Range . . . . . C0000 - C7FFF
Data Dump . . . . . (1998/10/08 15:41)
Address Range . . . . . C8000 - CBFFF
Data Dump . . . . . (04/22/98 Maxwell)
Smart Array Option ROM/BIOS (C)Co...
Address Range . . . . . CC000 - CFFFF
Data Dump . . . . . (04/22/98 ROC
ROC-LCR Option ROM/BIOS (C)Copyri...
Address Range . . . . . E8000 - EDFFF
Data Dump . . . . . ( CPQSCSI d)
Bootblock ROM . . . . . 04/17/2000
Standby Recovery Server
Status . . . . . Disabled
COM Port . . . . . COM1
Server Configuration . . . . . Recovery
Timeout Value . . . . . 1 minutes
Memory Boards Identified:
System Board
DIMM Slot 1 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 2 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 3 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 4 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 5 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 6 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 7 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 8 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 9 . . . . . 0 Megabytes
DIMM Slot 10 . . . . . 0 Megabytes
DIMM Slot 11 . . . . . 0 Megabytes
DIMM Slot 12 . . . . . 0 Megabytes
DIMM Slot 13 . . . . . 0 Megabytes
DIMM Slot 14 . . . . . 0 Megabytes
DIMM Slot 15 . . . . . 0 Megabytes
DIMM Slot 16 . . . . . 0 Megabytes
Total Compaq Memory . . . . . 8192 Megabytes
Keyboard . . . . . Standard 11-Bit
LPT Ports . . . . . LPT1 (Address 3BC)
COM Ports . . . . . COM1 (Address 3F8)
COM2 (Address 2F8)
Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Revision ID . . . . . 05h
Programming Interface . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h
Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h

```

```

Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h
Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h
Other Network Controller
PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Subsystem Vendor ID . . . . . 10E4h
Subsystem ID . . . . . 0000h
Revision ID . . . . . 00h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h
Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 4

```

```

Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h
Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h
Diskette Drive A . . . . . 1.44 Megabyte (3.5
inch)
Drive Controller 1, Compaq Integrated Smart Array
Controller
IDA Firmware Revision . . . . . 1.34
Array Accelerator Memory . . . . . 8188 Kbytes
Accelerator Status . . . . . Not Configured
Battery count . . . . . 0
Batteries charged . . . . . 0
Batteries failed . . . . . 0
Internal ProLiant . . . . . Bus 2, Rev. JB21
Logical Drive 1 . . . . . 9095 Megabyte
Fault Tolerance . . . . . Mirroring
OS Format . . . . . Multi-Sector
Distribution
Drive geometry (Cyl, Hds, Sec) 2177, 255, 32
Array Accelerator . . . . . Disabled

```

```

Hard Drive 1
  SCSI Bus . . . . . 2
  SCSI ID . . . . . 0
  Serial Number . . . . .
LS71629800007019KC9P
  Firmware Revision 1 . . . . . 3B07
  Model Number . . . . . COMPAQ BD009122BA
  Initialized for Monitoring . . . . . Yes
  Reference time . . . . . 202990
  Sectors read . . . . . *4101370934
  Hard read errors . . . . . 0
  Read errors retry . . . . . 0
  ECC read errors . . . . . 0
  Sectors written . . . . . 417224884
  Hard write errors . . . . . 0
  Write errors retry . . . . . 0
  Seek count . . . . . 688226
  Seek errors . . . . . 0
  Spin cycles . . . . . 1
  Spin up time . . . . . 0
  Seek time track . . . . . 47%
  Seek time third . . . . . 70%
  Seek time full . . . . . 72%
  Reallocated sectors . . . . . 1470
  Recovers read failed . . . . . 0
  Bus faults . . . . . 0

Hard Drive 2
  SCSI Bus . . . . . 2
  SCSI ID . . . . . 1
  Serial Number . . . . .
LSD344120000W043015B
  Firmware Revision 1 . . . . . 3B09
  Model Number . . . . . COMPAQ BD009122BA
  Initialized for Monitoring . . . . . Yes
  Reference time . . . . . 39651
  Sectors read . . . . . *3160477874
  Hard read errors . . . . . 0
  Read errors retry . . . . . 0
  ECC read errors . . . . . 0
  Sectors written . . . . . 57330023
  Hard write errors . . . . . 0
  Write errors retry . . . . . 0
  Seek count . . . . . 164512
  Seek errors . . . . . 0
  Spin cycles . . . . . 4
  Spin up time . . . . . 113
  Seek time track . . . . . 22%
  Seek time third . . . . . 36%
  Seek time full . . . . . 35%
  Reallocated sectors . . . . . 17
  Recovers read failed . . . . . 0
  Bus faults . . . . . 0

Graphics Mode . . . . . 03 (80-Column Text)

Primary Monitor attached to . . . . . ATI RAGE IIC PCI
Graphics Controller
with Video Graphics Color Monitor

Base Memory
  System Total . . . . . 636 Kbytes

```

```

Amount Free . . . . . 594 Kbytes
(609088 Bytes)

Extended Memory
  System Total . . . . . 8387584 Kbytes

Expanded Memory
  LIM Driver Support . . . . . LIM driver not
loaded

Operating System . . . . . MS-DOS version 7.00
(from diskette)

Environment variables
  PATH=
  PROMPT=$PSG
  COMSPEC=A:\COMMAND.COM
  CMDLINE=inspect /u
End of environment

Revisions Table

Previous Revisions
  I/O Board Revision . . . . . 01
  Assembly Version . . . . . 1
  Functional Revision Level . . . . . A
  Memory Board 1 Revision . . . . . 01
  Assembly Version . . . . . 1
  Functional Revision Level . . . . . A

Current Revisions
  I/O Board Revision . . . . . 01
  Assembly Version . . . . . 1
  Functional Revision Level . . . . . A
  Memory Board 1 Revision . . . . . 01
  Assembly Version . . . . . 1
  Functional Revision Level . . . . . A

System serial number . . . . . D947BX71K118

Memory Allocation (including INSPECT)
  PSP SIZE NAME TRAPPED INTERRUPTS
-----
0887 007200 COMMAND.COM 2Fh 2Eh 24h 23h 22h
0A52 218144 INSPECT.EXE FFh F9h F3h F2h F1h
F0h EEh EBh
E3h 3Fh 00h

System Configuration Memory
  00 - 0F : 31 00 30 00 14 00 03 19 07 00 26
82 50 80 00 00
  10 - 1F : 40 00 00 00 03 80 02 00 3C 00 00
00 00 00 00 02
  20 - 2F : 00 00 00 00 7F 20 00 40 00 92 00
00 00 18 02 8C
  30 - 3F : 00 3C 20 80 00 00 XX XX XX XX XX
XX XX XX XX XX

BIOS Data Area
  40:0000 : F8 03 F8 02 00 00 00 00 BC 03 00
00 00 00 00 9F

```

```

40:0010 : 27 44 00 7C 02 00 00 00 00 00 1E
00 1E 00 00 00
40:0020 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0030 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 01 01
40:0040 : 25 00 04 00 00 2B 01 0F 02 03 50
00 00 10 00 00
40:0050 : 00 18 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:0060 : 0E 0D 00 D4 03 29 30 C2 11 85 76
00 CB 81 0E 00
40:0070 : 00 00 00 12 00 01 00 00 14 14 14
14 01 01 01 01
40:0080 : 1E 00 3E 00 18 10 00 60 F9 11 0B
01 00 00 00 05
40:0090 : 17 00 00 00 2B 00 00 00 00 00 00
00 00 00 00 00
40:00A0 : 00 00 00 00 00 00 00 00 5C 13 00
C0 00 00 00 00
40:00B0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00C0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00D0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00E0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00F0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00

Interrupt Vector Table (including INSPECT)
  00 - 03 : 0A62:0555 0070:0465
07BE:0016 0070:0465
  04 - 07 : 0070:0465 F000:FF54
F000:4D64 F000:9BD0
  08 - 0B : 07BE:001F 07BE:0028
F000:9BD0 F000:9BD0
  0C - 0F : F000:9BD0 F000:9BD0
07BE:009A 0070:0465
  10 - 13 : C000:12DE F000:F84D
F000:F841 0070:03EE
  14 - 17 : F000:DDBF 0204:0240
0070:042D F000:EF83
  18 - 1B : F000:BOBB 087F:002F
F000:FE6E 0070:045F
  1C - 1F : F000:FF53 F000:0000
0000:0522 C000:2019
  20 - 23 : 00C9:0FA8 00C9:0FB2
0887:0314 0887:016D
  24 - 27 : 0887:0178 00C9:0FBC
00C9:0FC6 00C9:0FD0
  28 - 2B : 00C9:106F 0070:0466
00C9:106F 00C9:106F
  2C - 2F : 00C9:106F 00C9:106F
0887:0162 0888:01CC
  30 - 33 : C90F:E4EA F000:9B00
00C9:106F 00C9:106F
  34 - 37 : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
  38 - 3B : 00C9:106F 00C9:106F
00C9:106F 00C9:106F

```

```

3C - 3F : 00C9:106F 00C9:106F
00C9:106F 1B1E:04F3
40 - 43 : F000:EC59 CC1F:01C6
F000:F065 C000:242C
44 - 47 : F000:9BD0 F000:9BD0
0000:0000 F000:9BD0
48 - 4B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
4C - 4F : F000:9BD0 F000:9BD0
F000:9BD0 0070:04FC
50 - 53 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
54 - 57 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
58 - 5B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
5C - 5F : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
60 - 63 : 0000:0000 0000:0000
0000:0000 0000:0000
64 - 67 : 0000:0000 0000:0000
0000:0000 0000:0000
68 - 6B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
6C - 6F : F000:9BD0 C000:12DE
F000:9BD0 F000:9BD0
70 - 73 : 07BE:0035 F000:9C1F
F000:9BD0 07BE:00CA
74 - 77 : 07BE:00E2 F000:9C21
07BE:00FA F000:9BD0
78 - 7B : 0000:0000 0000:0000
0000:0000 0000:0000
7C - 7F : 0000:0000 0000:0000
0000:0000 0000:0000
80 - 83 : 0000:0000 0000:0000
0000:0000 0000:0000
84 - 87 : 0000:0000 0000:0000
0000:0000 0000:0000
88 - 8B : 0000:0000 0000:0000
0000:0000 0000:0000
8C - 8F : 0000:0000 0000:0000
0000:0000 0000:0000
90 - 93 : 0000:0000 0000:0000
0000:0000 0000:0000
94 - 97 : 0000:0000 0000:0000
0000:0000 0000:0000
98 - 9B : 0000:0000 0000:0000
0000:0000 0000:0000
9C - 9F : 0000:0000 0000:0000
0000:0000 0000:0000
A0 - A3 : 0000:0000 0000:0000
0000:0000 0000:0000
A4 - A7 : 0000:0000 0000:0000
0000:0000 0000:0000
AB - AB : 0000:0000 0000:0000
0000:0000 0000:0000
AC - AF : 0000:0000 0000:0000
0000:0000 0000:0000
B0 - B3 : 0000:0000 0000:0000
0000:0000 0000:0000
B4 - B7 : 0000:0000 0000:0000
0000:0000 0000:0000

```

```

B8 - BB : 0000:0000 0000:0000
0000:0000 0000:0000
BC - BF : 0000:0000 0000:0000
0000:0000 0000:0000
C0 - C3 : 0000:0000 0000:0000
0000:0000 0000:0000
C4 - C7 : 0000:0000 0000:0000
0000:0000 0000:0000
C8 - CB : 0000:0000 0000:0000
0000:0000 0000:0000
CC - CF : 0000:0000 0000:0000
0000:0000 0000:0000
D0 - D3 : 0000:0000 0000:0000
0000:0000 0000:0000
D4 - D7 : 0000:0000 0000:0000
0000:0000 0000:0000
D8 - DB : 0000:0000 0000:0000
0000:0000 0000:0000
DC - DF : 0000:0000 0000:0000
0000:0000 0000:0000
E0 - E3 : 0000:0000 0000:0000
0008:0008 0E11:0CFC
E4 - E7 : 0000:0000 0000:0809
0000:0400 0000:6E51
E8 - EB : 0008:0000 00C8:00C8
0078:0CFD 0CFD:0078
EC - EF : 0000:1F07 4000:0006
1F85:0046 0087:4000
F0 - F3 : 13C1:0046 1CCE:0013
13C1:D527 1400:1CCE
F4 - F7 : F000:4D30 03DA:0007
0000:0000 0000:7289
F8 - FB : 5FFE:0020 1497:6304
003E:0003 0000:090D
FC - FF : 514B:00FC 0000:00F6
EFE2:0000 0246:F000

PCI Devices Information
Signature . . . . . PCI
Config Mechanism #1 . . . . . Supported
Config Mechanism #2 . . . . . Not Supported
Spec Cycle for Config #1 . . . . . Supported
Spec Cycle for Config #2 . . . . . Not Supported
BIOS Interface Version . . . . . 2.10
Last PCI Bus Number . . . . . 142
Number of PCI Devices . . . . . 12

PCI Bus Number . . . . . 0
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 7
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E77C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7600000h
Memory Address Length . . . . . 100000h

```

```

IO Address Base . . . . . 2000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 2
Function Number . . . . . 00h
Slot Number . . . . . 8
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E75C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7400000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 9
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E73C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7200000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2800h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 13
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 1002h
Device ID . . . . . 4756h
Revision ID . . . . . 7Ah
Device Type . . . . . VGA Compatible
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFFE0000h
IRQ Line . . . . . 255
IRQ Pin . . . . . Not Used
Memory Address Base . . . . . E3000000h
Memory Address Length . . . . . 1000000h
IO Address Base . . . . . 2C00h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E71D0000h
Memory Address Length . . . . . 1000h

PCI Bus Number . . . . . 0
Device Number . . . . . 14

```



```

Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 0E11h
Device ID . . . . . 0010h
Revision ID . . . . . 01h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF80000h
IRQ Line . . . . . 11
IRQ Pin . . . . . INTA#
IO Address Base . . . . . 3000h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E6000000h
Memory Address Length . . . . . 1000000h
Memory Address Base . . . . . E5000000h
Memory Address Length . . . . . 1000000h

PCI Bus Number . . . . . 5
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 3
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1DC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1C00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 5
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 4
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1BC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1A00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller

```

```

Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Revision ID . . . . . 00h
Device Type . . . . . Other Network
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h

```

```

Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

ProLiant is a trademark of Compaq Computer
Corporation.

Date . . . . . 07/19/2000
Time . . . . . 14:29:57

Product . . . . . ProLiant

Machine ID
From System Board . . . . . CPQ1608

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 8
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 7
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 6
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 5
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

```

```

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 4
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 3
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 2
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor . . . . . Pentium III(R) Xeon
at 700 MHz
Slot . . . . . 1
Secondary Cache . . . . . 2048K
CPU ID . . . . . 06A0

Processor(s) Mapped Out . . . . . None

Numeric Coprocessor . . . . . Integrated 387-
Compatible

Expansion Bus . . . . . ISA, PCI

System Identification Number . . D947BX71K102

CPU Mode . . . . . Real Mode

System ROM
Revision . . . . . 06/16/2000
Family . . . . . P42
Flashable . . . . . Yes
Supports F10 partition . . . . . Yes

Video Controller ROM
Revision . . . . . 3.96

Option ROMs
Address Range . . . . . C0000 - C7FFF
Data Dump . . . . . (1998/10/08 15:41)

Address Range . . . . . C8000 - CBFFF
Data Dump . . . . . (04/22/98 Maxwell

Smart Array Option ROM/BIOS (C)Co...
Address Range . . . . . CC000 - CFFFF
Data Dump . . . . . (04/22/98 ROC
ROC-LCR Option ROM/BIOS (C)Copyri...
Address Range . . . . . E8000 - EDFFF
Data Dump . . . . . ( CPQSCSI d)

Bootblock ROM . . . . . 04/17/2000

Standby Recovery Server
Status . . . . . Disabled

```

```

COM Port . . . . . COM1
Server Configuration . . . . . Recovery
Timeout Value . . . . . 1 minutes

Memory Boards Identified:
System Board
DIMM Slot 1 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 2 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 3 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 4 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 5 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 6 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 7 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 8 (SDRAM) . . . . . 1024 Megabytes
DIMM Slot 9 . . . . . 0 Megabytes
DIMM Slot 10 . . . . . 0 Megabytes
DIMM Slot 11 . . . . . 0 Megabytes
DIMM Slot 12 . . . . . 0 Megabytes
DIMM Slot 13 . . . . . 0 Megabytes
DIMM Slot 14 . . . . . 0 Megabytes
DIMM Slot 15 . . . . . 0 Megabytes
DIMM Slot 16 . . . . . 0 Megabytes
Total Compaq Memory . . . . . 8192 Megabytes

Keyboard . . . . . Enhanced

LPT Ports . . . . . LPT1 (Address 378)

COM Ports . . . . . COM1 (Address 3F8)
COM2 (Address 2F8)

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h

```

```

Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h

Other Network Controller
PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Subsystem Vendor ID . . . . . 10E4h
Subsystem ID . . . . . 0000h
Revision ID . . . . . 00h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

Compaq NC3131 Fast Ethernet NIC
Device Type . . . . . Ethernet Controller
PCI Bus Number . . . . . 14

```

```

Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Subsystem Vendor ID . . . . . 0E11h
Subsystem ID . . . . . B0DDh
Revision ID . . . . . 05h
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7E00000h
Memory Address Length . . . . . 100000h

Diskette Drive A . . . . . 1.44 Megabyte (3.5
inch)

Drive Controller 1, Compaq Integrated Smart Array
Controller
  IDA Firmware Revision . . . . . 1.34
  Array Accelerator Memory . . . . . 8188 Kbytes
  Accelerator Status . . . . . Not Configured
  Battery count . . . . . 0
  Batteries charged . . . . . 0
  Batteries failed . . . . . 0
  Internal ProLiant . . . . . Bus 2, Rev. JB21

Logical Drive 1 . . . . . 9095 Megabyte
Fault Tolerance . . . . . Mirroring
OS Format . . . . . Multi-Sector
Distribution
  Drive geometry (Cyl, Hds, Sec) 2177, 255, 32
  Array Accelerator . . . . . Disabled

Hard Drive 1
  SCSI Bus . . . . . 2
  SCSI ID . . . . . 0
  Serial Number . . . . .
LS70218000010191UQG
  Firmware Revision 1 . . . . . 3B07
  Model Number . . . . . COMPAQ BD009122BA
  Initialized for Monitoring . . . . . Yes
  Reference time . . . . . 203169
  Sectors read . . . . . *3930016947
  Hard read errors . . . . . 0
  Read errors retry . . . . . 0
  ECC read errors . . . . . 0
  Sectors written . . . . . 406316992
  Hard write errors . . . . . 0
  Write errors retry . . . . . 0
  Seek count . . . . . 683578
  Seek errors . . . . . 0
  Spin cycles . . . . . 1
  Spin up time . . . . . 0
  Seek time track . . . . . 47%
  Seek time third . . . . . 70%

```

```

Seek time full . . . . . 72%
Reallocated sectors . . . . . 2343
Recovers read failed . . . . . 0
Bus faults . . . . . 0

Hard Drive 2
  SCSI Bus . . . . . 2
  SCSI ID . . . . . 1
  Serial Number . . . . .
LSD482890000W0421E1Z
  Firmware Revision 1 . . . . . 3B09
  Model Number . . . . . COMPAQ BD009122BA
  Initialized for Monitoring . . . . . Yes
  Reference time . . . . . 39877
  Sectors read . . . . . *3103825051
  Hard read errors . . . . . 0
  Read errors retry . . . . . 0
  ECC read errors . . . . . 0
  Sectors written . . . . . 65413354
  Hard write errors . . . . . 0
  Write errors retry . . . . . 0
  Seek count . . . . . 159936
  Seek errors . . . . . 0
  Spin cycles . . . . . 4
  Spin up time . . . . . 110
  Seek time track . . . . . 22%
  Seek time third . . . . . 36%
  Seek time full . . . . . 35%
  Reallocated sectors . . . . . 287
  Recovers read failed . . . . . 1
  Bus faults . . . . . 0

Graphics Mode . . . . . 03 (80-Column Text)

Primary Monitor attached to . . ATI RAGE IIC PCI
Graphics Controller
with Video Graphics Color Monitor

Base Memory
  System Total . . . . . 636 Kbytes
  Amount Free . . . . . 594 Kbytes
(609088 Bytes)

Extended Memory
  System Total . . . . . 8387584 Kbytes

Expanded Memory
  LIM Driver Support . . . . . LIM driver not
loaded

Operating System . . . . . MS-DOS version 7.00
(from diskette)

Environment variables
  PATH=
  PROMPT=$P$G
  COMSPEC=A:\COMMAND.COM
  CMDLINE=inspect /u
End of environment

Revisions Table

```

```

Previous Revisions
  I/O Board Revision . . . . . 01
  Assembly Version . . . . . 1
  Functional Revision Level . . . . . A
  Memory Board 1 Revision . . . . . 01
  Assembly Version . . . . . 1
  Functional Revision Level . . . . . A

Current Revisions
  I/O Board Revision . . . . . 01
  Assembly Version . . . . . 1
  Functional Revision Level . . . . . A
  Memory Board 1 Revision . . . . . 01
  Assembly Version . . . . . 1
  Functional Revision Level . . . . . A

System serial number . . . . . D947BX71K102

Memory Allocation (including INSPECT)
  PSP SIZE NAME TRAPPED INTERRUPTS
-----
0887 007200 COMMAND.COM 2Fh 2Eh 24h 23h 22h
0A52 218144 INSPECT.EXE FFh F9h F4h F3h F2h
F1h F0h EEh
Ebh E4h 3Fh 00h

System Configuration Memory
  00 - 0F : 10 00 30 00 14 00 03 19 07 00 26
  82 50 80 00 00
  10 - 1F : 40 00 00 00 03 80 02 00 3C 00 00
  00 00 00 00 02
  20 - 2F : 00 00 00 00 7F 20 20 40 00 92 00
  00 00 18 02 AC
  30 - 3F : 00 3C 20 80 00 00 XX XX XX XX XX XX
  XX XX XX XX XX

BIOS Data Area
  40:0000 : F8 03 F8 02 00 00 00 00 78 03 00
  00 00 00 00 9F
  40:0010 : 27 44 00 7C 02 00 00 00 00 00 1E
  00 1E 00 00 00
  40:0020 : 00 00 00 00 00 00 00 00 00 00 00
  00 00 00 00 00
  40:0030 : 00 00 00 00 00 00 00 00 00 00 00
  00 00 00 01 01
  40:0040 : 25 00 04 00 00 2B 01 0F 02 03 50
  00 00 10 00 00
  40:0050 : 00 18 00 00 00 00 00 00 00 00 00
  00 00 00 00 00
  40:0060 : 0E 0D 00 D4 03 29 30 C2 11 85 76
  00 3B 80 0E 00
  40:0070 : 00 00 00 12 00 01 00 00 14 14 14
  14 01 01 01 01
  40:0080 : 1E 00 3E 00 18 10 00 60 F9 11 0B
  01 00 00 00 05
  40:0090 : 17 00 00 00 2B 00 10 00 00 00 00
  00 00 00 00 00
  40:00A0 : 00 00 00 00 00 00 00 00 5C 13 00
  C0 00 00 00 00
  40:00B0 : 00 00 00 00 00 00 00 00 00 00 00
  00 00 00 00 00

```

```

40:00C0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00D0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00E0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00
40:00F0 : 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00

```

Interrupt Vector Table (including INSPECT)

```

00 - 03 : 0A62:0555 0070:0465
07BE:0016 0070:0465
04 - 07 : 0070:0465 F000:FF54
F000:4D64 F000:9BD0
08 - 0B : 07BE:001F 07BE:0028
F000:9BD0 F000:9BD0
0C - 0F : F000:9BD0 07BE:0082
07BE:009A 0070:0465
10 - 13 : C000:12DE F000:F84D
F000:F841 0070:03EE
14 - 17 : F000:DDBF 0204:0240
0070:042D F000:EF83
18 - 1B : F000:B0BB 087F:002F
F000:FE6E 0070:045F
1C - 1F : F000:FF53 F000:0000
0000:0522 C000:2019
20 - 23 : 00C9:0FA8 00C9:0FB2
0887:0314 0887:016D
24 - 27 : 0887:0178 00C9:0FBC
00C9:0FC6 00C9:0FD0
28 - 2B : 00C9:106F 0070:0466
00C9:106F 00C9:106F
2C - 2F : 00C9:106F 00C9:106F
0887:0162 0888:01CC
30 - 33 : C90F:E4EA F000:9B00
00C9:106F 00C9:106F
34 - 37 : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
38 - 3B : 00C9:106F 00C9:106F
00C9:106F 00C9:106F
3C - 3F : 00C9:106F 00C9:106F
00C9:106F 1B1E:04F3
40 - 43 : F000:EC59 CC1F:01C6
F000:F065 C000:242C
44 - 47 : F000:9BD0 F000:9BD0
0000:0000 F000:9BD0
48 - 4B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
4C - 4F : F000:9BD0 F000:9BD0
F000:9BD0 0070:04FC
50 - 53 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
54 - 57 : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
58 - 5B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
5C - 5F : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
60 - 63 : 0000:0000 0000:0000
0000:0000 0000:0000
64 - 67 : 0000:0000 0000:0000
0000:0000 0000:0000

```

```

68 - 6B : F000:9BD0 F000:9BD0
F000:9BD0 F000:9BD0
6C - 6F : F000:9BD0 C000:12DE
F000:9BD0 F000:9BD0
70 - 73 : 07BE:0035 F000:9C1F
F000:9BD0 F000:9BD0
74 - 77 : 07BE:00E2 F000:9C21
07BE:00FA F000:9BD0
78 - 7B : 0000:0000 0000:0000
0000:0000 0000:0000
7C - 7F : 0000:0000 0000:0000
0000:0000 0000:0000
80 - 83 : 0000:0000 0000:0000
0000:0000 0000:0000
84 - 87 : 0000:0000 0000:0000
0000:0000 0000:0000
88 - 8B : 0000:0000 0000:0000
0000:0000 0000:0000
8C - 8F : 0000:0000 0000:0000
0000:0000 0000:0000
90 - 93 : 0000:0000 0000:0000
0000:0000 0000:0000
94 - 97 : 0000:0000 0000:0000
0000:0000 0000:0000
98 - 9B : 0000:0000 0000:0000
0000:0000 0000:0000
9C - 9F : 0000:0000 0000:0000
0000:0000 0000:0000
A0 - A3 : 0000:0000 0000:0000
0000:0000 0000:0000
A4 - A7 : 0000:0000 0000:0000
0000:0000 0000:0000
A8 - AB : 0000:0000 0000:0000
0000:0000 0000:0000
AC - AF : 0000:0000 0000:0000
0000:0000 0000:0000
B0 - B3 : 0000:0000 0000:0000
0000:0000 0000:0000
B4 - B7 : 0000:0000 0000:0000
0000:0000 0000:0000
B8 - BB : 0000:0000 0000:0000
0000:0000 0000:0000
BC - BF : 0000:0000 0000:0000
0000:0000 0000:0000
C0 - C3 : 0000:0000 0000:0000
0000:0000 0000:0000
C4 - C7 : 0000:0000 0000:0000
0000:0000 0000:0000
C8 - CB : 0000:0000 0000:0000
0000:0000 0000:0000
CC - CF : 0000:0000 0000:0000
0000:0000 0000:0000
D0 - D3 : 0000:0000 0000:0000
0000:0000 0000:0000
D4 - D7 : 0000:0000 0000:0000
0000:0000 0000:0000
D8 - DB : 0000:0000 0000:0000
0000:0000 0000:0000
DC - DF : 0000:0000 0000:0000
0000:0000 0000:0000
E0 - E3 : 0000:0000 0000:0000
0000:0000 00D8:0000

```

```

E4 - E7 : 0CFC:00D8 0000:000D
0020:0020 FFFF:0CFC
EB - EB : C0ED:0000 00C8:00C8
0078:0CFD 0CFC:0078
EC - EF : 0000:1F07 3900:0006
1F85:0046 0087:3900
F0 - F3 : 146E:0046 1C88:0013
146E:D527 1400:1C88
F4 - F7 : 1BB0:0246 0101:7248
0000:0000 0000:5FFE
F8 - FB : 5FFE:0020 1497:6304
0008:0003 0000:081A
FC - FF : 514B:00FC 0000:00F6
EFE2:0000 0246:F000

```

PCI Devices Information

```

Signature . . . . . PCI
Config Mechanism #1 . . . . . Supported
Config Mechanism #2 . . . . . Not Supported
Spec Cycle for Config #1 . . . . . Supported
Spec Cycle for Config #2 . . . . . Not Supported
BIOS Interface Version . . . . . 2.10
Last PCI Bus Number . . . . . 142
Number of PCI Devices . . . . . 12

PCI Bus Number . . . . . 0
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 7
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E77C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7600000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 2
Function Number . . . . . 00h
Slot Number . . . . . 8
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E75C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7400000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2400h
IO Address Length . . . . . 100h

```

```

PCI Bus Number . . . . . 0
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 9
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . E73C0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . E7200000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 2800h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 0
Device Number . . . . . 13
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 1002h
Device ID . . . . . 4756h
Revision ID . . . . . 7Ah
Device Type . . . . . VGA Compatible
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFFE0000h
IRQ Line . . . . . 255
IRQ Pin . . . . . Not Used
Memory Address Base . . . . . E3000000h
Memory Address Length . . . . . 1000000h
IO Address Base . . . . . 2C00h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E71D0000h
Memory Address Length . . . . . 1000h

PCI Bus Number . . . . . 0
Device Number . . . . . 14
Function Number . . . . . 00h
Slot Number . . . . . 0
Vendor ID . . . . . 0E11h
Device ID . . . . . 0010h
Revision ID . . . . . 01h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF80000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
IO Address Base . . . . . 3000h
IO Address Length . . . . . 100h
Memory Address Base . . . . . E6000000h
Memory Address Length . . . . . 1000000h
Memory Address Base . . . . . E5000000h
Memory Address Length . . . . . 1000000h

PCI Bus Number . . . . . 5
Device Number . . . . . 3
Function Number . . . . . 00h
Slot Number . . . . . 3
Vendor ID . . . . . 0E11h

```

```

Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 10
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1DC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1C00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4000h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 5
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 4
Vendor ID . . . . . 0E11h
Device ID . . . . . B060h
Revision ID . . . . . 02h
Device Type . . . . . RAID Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . FFF00000h
IRQ Line . . . . . 5
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F1BC0000h
Memory Address Length . . . . . 40000h
Memory Address Base . . . . . F1A00000h
Memory Address Length . . . . . 100000h
IO Address Base . . . . . 4400h
IO Address Length . . . . . 100h

PCI Bus Number . . . . . 6
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 6
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 1
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15

```

```

IRQ Pin . . . . . INTA#
Memory Address Base . . . . . EF8E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 8020h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F1E00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 13
Device Number . . . . . 1
Function Number . . . . . 00h
Slot Number . . . . . 10
Vendor ID . . . . . 10E4h
Device ID . . . . . 0005h
Revision ID . . . . . 00h
Device Type . . . . . Other Network
Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F7DF0000h
Memory Address Length . . . . . 1000h
Memory Address Base . . . . . F7DE0000h
Memory Address Length . . . . . 10000h
Memory Address Base . . . . . F4000000h
Memory Address Length . . . . . 2000000h

PCI Bus Number . . . . . 14
Device Number . . . . . 4
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74F0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9000h
IO Address Length . . . . . 20h
Memory Address Base . . . . . F7F00000h
Memory Address Length . . . . . 100000h

PCI Bus Number . . . . . 14
Device Number . . . . . 5
Function Number . . . . . 00h
Slot Number . . . . . 11
Vendor ID . . . . . 0E11h
Device ID . . . . . 1229h
Revision ID . . . . . 05h
Device Type . . . . . Ethernet Controller
Programming Interface . . . . . 00h
Expansion ROM Base Address . . . . . 0h
IRQ Line . . . . . 15
IRQ Pin . . . . . INTA#
Memory Address Base . . . . . F74E0000h
Memory Address Length . . . . . 1000h
IO Address Base . . . . . 9020h
IO Address Length . . . . . 20h

```

Memory Address Base F7E0000h
 Memory Address Length 100000h

ProLiant is a trademark of Compaq Computer Corporation.

System Summary

System Information report written at: 07/20/2000 11:19:02 AM
 [System Information]

[Following are sub-categories of this main category]

The System Summary listed here is for one node. All nodes have the same system summary.

Item	Value
OS Name	Microsoft Windows 2000 Advanced Server
Version	5.0.2195 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	APR
System Manufacturer	Compaq
System Model	ProLiant
System Type	X86-based PC
Processor x86 Family 6 Model 10 Stepping 0	
GenuineIntel ~700 Mhz	
Processor x86 Family 6 Model 10 Stepping 0	
GenuineIntel ~700 Mhz	
Processor x86 Family 6 Model 10 Stepping 0	
GenuineIntel ~700 Mhz	
Processor x86 Family 6 Model 10 Stepping 0	
GenuineIntel ~700 Mhz	
Processor x86 Family 6 Model 10 Stepping 0	
GenuineIntel ~700 Mhz	
Processor x86 Family 6 Model 10 Stepping 0	
GenuineIntel ~700 Mhz	
Processor x86 Family 6 Model 10 Stepping 0	
GenuineIntel ~700 Mhz	
Processor x86 Family 6 Model 10 Stepping 0	
GenuineIntel ~700 Mhz	
BIOS Version	06/16/00
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition2
Locale	United States
User Name	APR\Administrator
Time Zone	Central Daylight Time
Total Physical Memory	7,863,612 KB
Available Physical Memory	695,788 KB
Total Virtual Memory	17,683,784 KB
Available Virtual Memory	3,452,052 KB
Page File Space	9,820,172 KB
Page File	C:\pagefile.sys

Services

These are services running on all nodes.
 [Services]

Display Name	Name	State	Start Mode	Path	Error Control
Service Name	Tag ID				
Alerter	Alerter	Stopped	Auto	c:\winnt\system32\services.exe	Share Process
Application Management	AppMgmt	Stopped	Manual	c:\winnt\system32\services.exe	Share Process
Computer Browser	Browser	Running	Auto	c:\winnt\system32\services.exe	Share Process
Indexing Service	cisvc	Stopped	Manual	c:\winnt\system32\cisvc.exe	Share Process
ClipBook	ClipSrv	Stopped	Manual	c:\winnt\system32\clipsrv.exe	Own Process
Distributed File System	Dfs	Stopped	Auto	c:\winnt\system32\dfssvc.exe	Own Process
DHCP Client	Dhcp	Stopped	Auto	c:\winnt\system32\services.exe	Share Process
Logical Disk Manager Administrative Service	dmadmin	Stopped	Manual	c:\winnt\system32\dmadmin.exe	Share Process
Logical Disk Manager	dmserver	Stopped	Auto	c:\winnt\system32\services.exe	Share Process
DNS Client	Dnscache	Stopped	Auto	c:\winnt\system32\services.exe	Share Process
Event Log	Eventlog	Running	Auto	c:\winnt\system32\services.exe	Share Process
COM+ Event System	EventSystem	Stopped	Manual	c:\winnt\system32\svchost.exe	Share Process
Fax Service	Fax	Stopped	Manual	c:\winnt\system32\faxsvc.exe	Own Process
Intersite Messaging IsmServ	IsmServ	Stopped	Disabled	c:\winnt\system32\ismsserv.exe	Own Process
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	Stopped	Auto		Own Process
	c:\winnt\system32\llssrv.exe	Normal	LocalSystem	0	
TCP/IP NetBIOS Helper Service	LmHosts	Stopped	Auto		Share Process
	c:\winnt\system32\services.exe	Normal	LocalSystem	0	
Messenger	Messenger	Stopped	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	MSIExec	Stopped	Manual		Share Process
	c:\winnt\system32\msiexec.exe	Normal	LocalSystem	0	
Microsoft Search	MSSEARCH	Stopped	Auto		Share Process
	files\common files\system\mssearch\bin\mssearch.exe"	Normal	LocalSystem	0	
MSSQLSERVER	MSSQLSERVER	Stopped	Manual		Own Process
	c:\sql80\mssql\bin\sqlservr.exe	Normal	LocalSystem	0	
MSSQLServerADHelper	MSSQLServerADHelper	Stopped	Manual		Own Process
	files\microsoft sql server\80\tools\bin\sqladhlp.exe	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	Stopped	Manual		Share Process
	c:\winnt\system32\svchost.exe	Normal	LocalSystem	0	
File Replication	NtFrs	Stopped	Manual		Own Process
	c:\winnt\system32\ntfrs.exe	Normal	LocalSystem	0	

```

NT LM Security Support Provider      NtLmSsp
Stopped Manual Share Process
c:\winnt\system32\lsass.exe Normal
LocalSystem 0
Removable Storage NtmsSvc Stopped 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 Stopped
Auto Share Process
c:\winnt\system32\lsass.exe Normal
LocalSystem 0
Protected Storage ProtectedStorage Stopped
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
Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Routing and Remote Access RemoteAccess
Stopped Disabled Share Process
c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Remote Registry Service RemoteRegistry
Stopped 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 Running Manual Own Process
c:\winnt\system32\rsvp.exe -s Normal
LocalSystem 0
Security Accounts Manager SamSs Running
Auto Share Process
c:\winnt\system32\lsass.exe Normal
LocalSystem 0
Smart Card Helper SCardDrv Stopped Manual
Share Process
c:\winnt\system32\scardsvr.exe
Ignore LocalSystem 0
Smart Card SCardSvr Stopped Manual
Share Process
c:\winnt\system32\scardsvr.exe
Ignore LocalSystem 0
Task Scheduler Schedule Stopped Auto
Share Process
c:\winnt\system32\mstask.exe Normal
LocalSystem 0

```

```

RunAs Service seclogon Stopped Auto
Share Process
c:\winnt\system32\services.exe
Ignore LocalSystem 0
System Event Notification SENS Stopped
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
Print Spooler Spooler Stopped Auto Own
Process c:\winnt\system32\spoolsv.exe Normal
LocalSystem 0
SQLSERVERAGENT SQLSERVERAGENT Stopped
Manual Own Process
c:\sql80\mssql\bin\sqlagent.exe
Normal LocalSystem 0
Performance Logs and Alerts SysmonLog Stopped
Manual Own Process
c:\winnt\system32\smlogsvc.exe
Normal LocalSystem 0
Telephony Tapisrv Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k tapisrv
Normal LocalSystem 0
Tardis time service Tardis Running Auto Own
Process c:\winnt\system32\tardisnt.exe
Normal LocalSystem 0
Terminal Services TermService Stopped
Auto 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
Stopped 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
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

```

Appendix D: 180-Day Space

TPC-C 180 Day Space Requirements						
Warehouses	21600				Tpmc	262,243.60
Table	Rows	Data KB	Index KB	Extra 5% KB	8hr Space	Total Space KB
Warehouse	21,600	2,400	464	143		3007
District	216,000	24,384	480	1,243		26107
Customer	648,000,000	471,272,736	29,411,648	25,034,219		525718603
History	648,000,000	38,686,656	126,624		446,530,510	38813280
NewOrder	194,400,000	3,463,776	9,024	173,640		3646440
Orders	648,000,000	21,159,264	10,322,928		5,251,302,921	31482192
OrderLine	6,479,993,522	431,993,624	1,022,656		158,793,654	433016280
Item	100,000	114,336	672	5,750		120758
Stock	2,160,000,000	691,200,000	1,464,192	34,633,210		727297402
Total		1,657,917,176	42,358,688	59,848,206	5,856,627,085	1,760,124,070
MB						
Dynamic Space	480,312	Sum of Data for Order, Orderline and History				
Static Space	1,238,559	Sum of Data+Index+5%-Dynamic Space				
Free Space	na	Total Allocated Spac - (Dynamic + Static Space)				
Daily Growth	93,303	(Dynamic Space/(W*62.5))*tpmc				
Daily Spread	-	(Free Space - 1.5*Daily Growth) Zero Assumed				
180 Day Space MB	18,033,061					
180 Day Space GB	17,610.41	GB				
Log Size	59999.99	MB				
KB Per New Order	5.36	KB				
8 hr log MB	657.573	MB				
8 hr log GB	642.16	GB				
Space Usage	GB Needed	Disks Measured	GB Priced	Disk Size	Formatted Size	
180 Day Space DB	17,610.41	-	-	18GB	17.30	
DTC disks --->	2.88	2,016.00	17,080.88	9GB	8.47	
Total DB		24.00	203.34	9GB	8.47	
		2,040.00	17,284.22	9GB		
8-hr log + mirror	1,284.32	96.00	1,660.80	18GB	17.30	
OS, Swap (mirror)	96.00	24.00	203.34	9GB		
Total Storage	18,993.61	GB	19,148.36	GB		

Appendix E:

Third Party Letters



July 18, 2000

Mr. Andrew Bond
Compaq Computer Corp.
P.O. Box 692000
Houston, TX 77269-2000

Dear Mr. Bond:

Here is the information you requested regarding U.S. pricing for several Microsoft products, to be used in conjunction with TPC-C benchmark testing.


Part Number	Description	Unit Price	Quantity	Price
810-00945	SQL Server 2000 Enterprise Edition Per processor licensing Discount schedule: Select B discount plan	\$15,802	96	\$1,516,992
C10-00475	Windows 2000 Advanced Server Server license only - No CALs Discount schedule: Open Program - No Level	\$2,399	12	\$28,788
C11-00821	Windows 2000 Server Server license only - No CALs Discount schedule: Open Program - No Level	\$738	36	\$26,568
048-00317	Visual C++ Professional 6.0 Win32 5-year maintenance for above software \$2,095 per year per database server	\$549	1	\$549

Some products may not be currently orderable but will be available through Microsoft's normal distribution channels by August 1, 2000.

This quote is valid for the next 90 days.

If I can be of any further assistance, please contact me at (425) 703-7053 or jressler@microsoft.com.

Yours truly,


Jeff Ressler
Product Manager
SQL Server Marketing

Microsoft Corporation is an equal opportunity employer.

Amherst Computer Products Southwest, LP
 Houston Office
 6401 Southwest Freeway
 Houston, TX 77074
 (713) 795-2000

COMPAQ COMPUTER CORPORATION
 NORTH RECEIVING
 20555 SH 249 MC141001
 HOUSTON, TX 77070
 ATTN: MIKE

MICHAEL J KAITSCHUCK
 Corp.Acct.Exec.
 Reference # 1145639
 DATE : 07/21/00

700-96P

Qty	Mcd	Model No.	Description	Price	Ext. Price
12	COM	168758-001	PROLIANT 8500R X 700-2M, 8P, 4GB	78,400.00	940,800.00
24	COM	PES-PE-1Y-SU	PROLIANT ENTERPRISE SERVER - PARTS EXCHANGE -1YEAR	597.00	14,328.00
1	COM	OS-8X5X4-SU	ONSITE 5X8, 4-HOUR SERVICE UPGRADE FROM COMPAQ STANDARD SERVICE	230,628.32	230,628.32
48	COM	328808-B21	1GB MEMORY KIT (2 X 512MB MEM SDRAM DIMMS)	4,300.80	206,438.40
12	COM	317450-B21	COMPAQ NC3122 PCI DUAL 10-100	263.20	3,158.40
60	COM	158939-B21	COMPAQ SMART ARRAY CONTROLLER 5304-4SCSI CHANNELS	2,462.88	147,772.80
156	COM	103381-001	COM STRGWORKS ENCL 4214R RACK MOUNTABLE	2,912.00	454,272.00
12	COM	119829-B21	SW4200 D BUS IO MOD CTLR	560.00	6,720.00
24	COM	PSS-PE-1YEAR	PROLIANT STORAGE SYSTEM - PARTS EXCHANGE - 1 YEAR	255.00	6,120.00
48	COM	325900-001	15IN/13.8V 28MM 1024X768 MNTR V500 OPAL	172.48	8,279.04

Amherst Computer Products Southwest, LP
 Houston Office
 6401 Southwest Freeway
 Houston, TX 77074
 (713) 795-2000

COMPAQ COMPUTER CORPORATION
 NORTH RECEIVING
 20555 SH 249 MC141001
 HOUSTON, TX 77070
 ATTN: MIKE

MICHAEL J KAITSCHUCK
 Corp.Acct.Exec.
 Reference # 1145639
 DATE :07/21/00

700-96P

Qty	Mcd	Model No.	Description	Price	Ext. Price
1	COM	295480-B22	COM 4-8GB SLR TAPE DRIVE OPAL	448.00	448.00
15	COM	165753-001	COMPAQ RACK-MOUNT 42U RACK	1,680.00	25,200.00
14	COM	165664-001	HD ACC RACK COUPLING KIT (42U)	152.32	2,132.48
1	COM	165652-001	COMPAQ RACK SIDEWALL (42U RACK)	208.32	208.32
14	COM	242705-001	COMPAQ RACK MOUNTABLE UPS MODEL 3000	1,786.40	25,009.60
12	COM	422869-001	SERVERNET II PCI ADAPTER	890.40	10,684.80
2	COM	452219-001	SERVERNET II 12 PORT SWITCH	4,984.00	9,968.00
2,064	COM	328939-B22	INTL 9.1GB PLUGGABLE WIDE INT ULTRA2 UNIVERSAL 10K DRIVE (1IN)	469.28	968,593.92
96	COM	128418-B22	18.2GB PLUGGABLE ULTRA2 SCSI INT 1IN UNIVERSAL 10K DRIVE	782.88	75,156.48
17	COM	153552-001	PROLIANT 1600 6/600 MODEL 1 SYST P3-600 MHZ 512KB	2,688.00	45,696.00

Amherst Computer Products Southwest, LP
Houston Office
6401 Southwest Freeway
Houston, TX 77074
(713) 795-2000

COMPAQ COMPUTER CORPORATION
NORTH RECEIVING
20555 SH 249 MC141001
HOUSTON, TX 77070
ATTN: MIKE

MICHAEL J KAITSCHUCK
Corp.Acct.Exec.
Reference # 1145639
DATE :07/21/00

700-96P

Qty	Mcd	Model No.	Description	Price	Ext. Price
			128MB 1GB		
6	COM	123740-001	COM PROLIANT 1850R PII1550 128MB MODEL 1	4,032.00	24,192.00
13	COM	157829-001	PROLIANT DL380 P3-733 128MB SYST 256K RACK 3U	4,592.00	59,696.00
72	COM	FM-LOPRT-12	1 YEAR PARTS ONLY WORKGROUP SERVER	259.00	18,648.00
1	COM	OS-8X5X4-SU	ONSITE 5X8, 4-HOUR SERVICE UPGRADE FROM COMPAQ STANDARD SERVICE	18,802.92	18,802.92
17	COM	153555-B21	P3-600 CPU PROCESSOR OPTION KITCHIP FOR PROLIANT 800 1600 1850R 3000	1,006.88	17,116.96
69	COM	313615-B21	COMPAQ 128MB 100MHZ SDRAM MODULE	360.64	24,884.16
39	COM	128277-B21	128MB REG SDRAM DIMM 133 MHZ MEM	311.36	12,143.04
6	COM	153555-B21	P3-600 CPU PROCESSOR OPTION KITCHIP FOR PROLIANT 800 1600 1850R 3000	899.00	5,394.00

Amherst Computer Products Southwest, LP
 Houston Office
 6401 Southwest Freeway
 Houston, TX 77074
 (713) 795-2000

COMPAQ COMPUTER CORPORATION
 NORTH RECEIVING
 20555 SH 249 MC141001
 HOUSTON, TX 77070
 ATTN: MIKE

MICHAEL J KAITTSCHUCK
 Corp.Acct.Exec.
 Reference # 1145639
 DATE : 07/21/00

700-96P

Qty	Mcd	Model No.	Description	Price	Ext. Price
13	COM	159756-B21	PROCESSOR OPTION KIT P3-667 CHIP 133 FSB 256K FOR ML370	782.88	10,177.44
48	COM	317450-B21	COMPAQ NC3122 PCI DUAL 10-100	263.20	12,633.60
36	COM	328939-B22	INTL 9.1GB PLUGGABLE WIDE INT ULTRA2 UNIVERSAL 10K DRIVE (11IN)	469.28	16,894.08
1	ALS	COMMENT	COMMENT	-356,733.70	-356,733.70
					3,045,463.06



To order by phone, call: 1-800-397-8508

Shop by Brand
PRODUCT SHOWCASES
Select a Brand

Thursday, July 20
 My Cart
 > Order Status

[PC Products](#) |
 [MAC Products](#) |
 [NETWORKING](#) |
 [CORPORATE](#) |
 [EDUCATION](#) |
 [SUPPLIES](#) |
 [CLEARANCE](#)
Accounts Accounts Accounts Accounts Accounts

SEARCH

EtherFast 16 Port 10/100 Ummngd Switch

Manufacturer: Linksys
 Manufacturer Part #: DSSX16
 MicroWarehouse #: DEH4324
 Platform: PC/Misc



Availability: In Stock

Ships same day if ordered by 3 pm (ET) weekdays.
 Customers with APO/FPO addresses, please call to order.

VIEW ALL PRODUCTS IN SECTION
View All Products in Section



\$469.99
Buy Now

Etherfast 10/100 Switches provide an easy, flexible way to boost your network's performance while migrating to the power of Fast Ethernet. Etherfast 10/100 Switches include RJ-45 ports that can operate in either half or full-duplex mode, enabling your network to run at speeds up to 200Mbps.

Features:

- 16 Ports
- Dual speed 10/100 per port
- Full duplex per port (Auto-Sensing)
- 16,000 Mac addresses
- Optional 100BaseFX fiber upgrade modules
- Rackmount hardware included
- Manufacturer's 5-year limited warranty

CustomerService
 • Business Leasing Info
 • Licensing Programs
 • Careers @ warehouse.com
 • About MicroWarehouse

Product Compare

Shop By Brand

Notebooks

Desktops

Servers

PDA's/Handhelds

Digital Cameras

Drives/Storage

Hardware

Memory

Monitors

Networking

Printers

Scanners

Software

Supplies

Cables Warehouse

Top of Page

[Home](#) | [Mac Warehouse](#) | [Micro Warehouse](#) | [DataComm Warehouse](#)

[Supplies Warehouse](#) | [Shopping Basket](#) | [Email Us](#)

Canada | France | Germany | Holland | Sweden | United Kingdom

™ & © 1999 Micro Warehouse, Inc. All rights reserved.
[Read our Privacy Policy Statement.](#)
[Disclaimers](#)

http://www2.wareho.../product.asp?pf%5Fid=DEH4324&Dept%5FID= 7/20/2000



Shop by Brand
PRODUCT SHOWCASES
NEW!
Select a Brand

Thursday, July 20

My Cart
Order Status

SEARCH

PC Products **MAC Products** **NETWORKING** **SUPPLIES** **CLEARANCE**
Get the Newest Category! Corporate Accounts Education Accounts Government Accounts

ETHERFAST 8-PORT 10/100 RACKMOUNT SWITCH

Manufacturer: Linksys
Manufacturer Part #: EZXS88R
MicroWarehouse #: DEH4162
Platform: PC/Mac



Availability: In Stock

Ships same day if ordered by 3 pm (ET) weekdays.
Customers with APO/FPO addresses, please call to order.

\$159.99



Buy Now

Etherfast 10/100 Switches provide an easy, flexible way to boost your network's performance while migrating to the power of Fast Ethernet. Etherfast 10/100 Switches include RJ-45 ports that can operate in either half or full-duplex mode, enabling your network to run at speeds up to 200Mbps.

Features:

- 8 Ports
- Dual speed 10/100 per port
- Full duplex per port (Auto-Sensing)
- 16,000 Mac addresses
- Optional 100BaseFX fiber upgrade modules
- Rackmount hardware included
- Manufacturer's 5-year limited warranty

- Product Compare
- Shop By Brand
- Notebooks
- Desktops
- Servers
- PDA's/Handhelds
- Digital Cameras
- Drives/Storage
- Hardware
- Memory
- Monitors
- Networking
- Printers
- Scanners
- Software
- Supplies
- Cables Warehouse
- Customer Service
- Business Leasing Info
- Licensing Programs
- Careers@warehouse.com
- About MicroWarehouse

Top of Page
[Home](#) | [Mac Warehouse](#) | [Micro Warehouse](#) | [DataComm Warehouse](#)
[Supplies Warehouse](#) | [Shopping Basket](#) | [Email Us](#)

Canada | France | Germany | Holland | Sweden | United Kingdom

TM & © 1999 Micro Warehouse, Inc. All rights reserved.
Read our [Privacy Policy Statement](#).
[Disclaimers](#)

<http://www2.warehouse.com/product.asp?pf%5Fid=DEH4162>

7/20/2000

Nikolaiev, Mike

Subject: FW: Quotation for CT1017D1

-----Original Message-----

Sent: Thursday, July 20, 2000 3:37 PM

To: Nikolaiev, Mike

Subject: FW: Quotation for CT1017D1

From: lmarket88@aol.com [mailto:lmarket88@aol.com]

To: Nikolaiev, Mike

Cc: lmarket88@aol.com

Subject: Quotation for CT1017D1

Dear Mike,

Thank you for your interesting in our product. The price for 17-port 10 Mbps Ethernet Hub (CT1017D1) is \$37.00 for quantity over 1000 units. The manufacture, ArkPC, warrants its products for a period of lifetime from the date of purchase. Quote valid for 90 days.

Please let me know if you need any further information.

Best Regards,

Thomas
I-Market