



Hewlett-Packard Company

TPC Benchmark™ C
Full Disclosure Report
for
ProLiant ML370-G3-2M/3.2GHz 2P
using
Microsoft SQL Server 2000 Enterprise Edition
and
Microsoft Windows Server 2003, Enterprise Edition

Second Edition
March 2004

Second Edition – March 2004

Hewlett-Packard Company (HP) 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. HP 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, HP 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. HP 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 2004 Hewlett-Packard Company.

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., 2002

HP, NonStop, ProLiant ML370G3R, ProLiant DL360 and ProLiant are registered trademarks of Hewlett-Packard Company.

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

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	III
PREFACE	V
TPC BENCHMARK C OVERVIEW	V
ABSTRACT	VI
OVERVIEW.....	VI
TPC BENCHMARK C METRICS	VI
STANDARD AND EXECUTIVE SUMMARY STATEMENTS	VI
AUDITOR	VI
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
QUEUEING 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	17
<i>Durable Media Failure</i>	17
<i>Instantaneous Interruption and Loss of Memory</i>	17
CLAUSE 4 RELATED ITEMS.....	19
INITIAL CARDINALITY OF TABLES	19
DATABASE LAYOUT	19
TYPE OF DATABASE.....	20
DATABASE MAPPING.....	20
60 DAY SPACE.....	20
CLAUSE 5 RELATED ITEMS.....	21

THROUGHPUT	21
KEYING AND THINK TIMES.....	21
RESPONSE TIME FREQUENCY DISTRIBUTION CURVES AND OTHER GRAPHS	22
STEADY STATE DETERMINATION	27
WORK PERFORMED DURING STEADY STATE.....	27
MEASUREMENT PERIOD DURATION.....	27
REGULATION OF TRANSACTION MIX.....	28
TRANSACTION STATISTICS	28
CHECKPOINT COUNT AND LOCATION	29
CHECKPOINT DURATION.....	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 5.2, released December 2003.

TPC Benchmark C Overview

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

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

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

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

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

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

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

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

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark C test conducted on the HP ProLiant ML370G3R. The operating system used for the benchmark was Microsoft Windows Server 2003, Enterprise Edition. 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 (three year capital cost per measured tpmC), and the availability date are reported as:

60,364 tpmC
\$3.51 per tpmC

The availability date is March 17, 2004

Standard and Executive Summary Statements

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

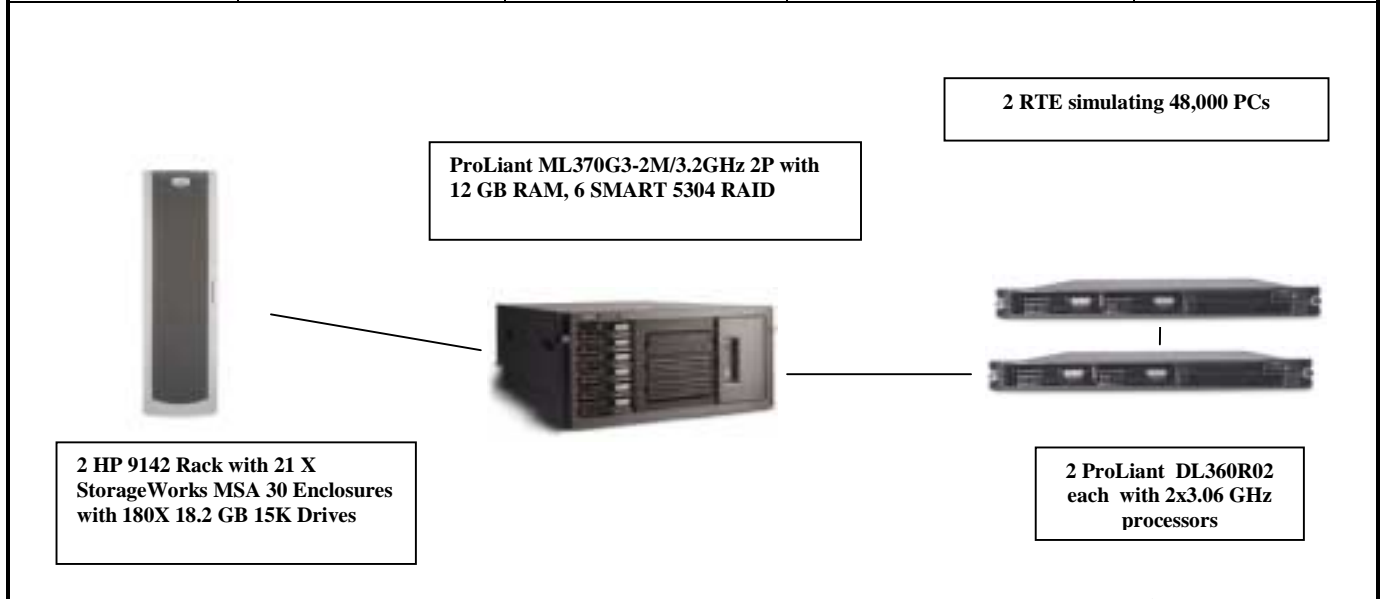
Auditor

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

Hewlett Packard	ProLiant ML370-G3-2M/3.2GHz 2P	TPC-C Rev. 5.2
Company	C/S with ProLiant DL360R02	Report Date: Mar 2, 2004

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
\$211,519	60,364	\$3.51	Mar 17, 2004

Processors	Database Manager	Operating System	Other Software	Number of Users
2 Intel Xeon 3.20 GHz – 2M L3 cache – Server 4 Pentium III 3.06 GHz – Clients	Microsoft SQL Server 2000 Enterprise Edition SP3	Microsoft Windows Server 2003, Enterprise Edition	Microsoft Visual C++ Microsoft COM+	48,000



	Server		Each Client	
System Components	Quantity	Description	Quantity	Description
Processor	2	3.20 GHz Intel Xeon w/ 2MB Cache	2	3.06 GHz Pentium III Xeon w/ 256K cache
Memory	6	2 GB DDR	1	1 GB (2x 512 MB)
Disk Controllers	1	Integrated Ultra-3 SCSI Controller	1	Integrated SMART Array Controller
	6	SMART 5304 Array Controller		
Disk Drives	1	18 GB SCSI Drive	1	36 GB SCSI Drive
	10	36 GB SCSI Drive		
	280	18.2 GB SCSI Drive		
Total Storage		5088 GB		36 GB
Backup Storage	1	DVD Writer		

Hewlett-Packard Company		ProLiant ML370-G3-2M/3.2GHz 2P Client/Server			TPC-C Rev. 5.2 Report Date: 2-Mar-04		
Description	Part Number	Third Party	Unit Price	Qty	Extended Price	3 yr. Maint. Price	
Server Hardware							
		Brand	Pricing				
ProLiant ML370G3T 2M 1P 3.2GHz, Integrated Gigabit NIC	352530-xx1		1	3,099	1	3,099	
X3.20GHz-2M /533MHz Processor Option Kit - ML370G3	352568-B21		1	1,499	1	1,499	
4GB PC2100 DDR SDRAM DIMM 2x2048 WW	300682-B21		1	3,499	3	10,497	
2-Bay Hot Plug Wide Ultra2/Ultra3 SCSI Drive Cage	244059-B21		1	370	1	370	
StorageWorks Enclosure Model MSA 30 - Rack-mountable	302969-B21		1	2,978	21	62,538	
Rack Model 9142 (42U - Opal) - Flat Pallet	120663-B21		1	1,321	2	2,642	
Side Panel Kit - 9142 Rack	120670-B21		1	207	1	207	
Smart Array 5304/256 Controller	283551-B21		1	2,247	6	13,482	
HP DVD Writer dvd400i	Q2119A#ABA		1	150	1	150	
UPS T700	136386-001		1	146	1	146	
18.2GB 15Krpm U320 UNI HDD	286775-B22		1	269	280	75,320	
18.2GB 15Krpm U320 UNI HDD (OS)	286775-B23		1	269	1	269	
18.2GB 15Krpm U320 UNI HDD (10% spares)	286775-B22		1	269	28	7,532	
36GB 10K U320 Pluggable Hard Drive WW (log)	286713-B22		1	299	12	3,588	
CarePaq Service - 300 Series Servers 3Yr,7x24,4hr	162657-002		1	949	1	949	
FM-4E724-36 3YR 24X7/4HR EMPTY DISK ENCL	171242-002		1	157	21	3,297	
				Subtotal	173,807	11,778	
Server Software							
Microsoft SQL Server 2000 Enterprise Edition(per processor)	810-00961	Microsoft	2	17,279	2	34,558	
Visual C++ .Net Standard	254-00170	Microsoft	2	109	1	109	
Microsoft Windows 2003, Enterprise Edition	P72-00264	Microsoft	2	2,399	1	2,399	
Database Server Support Package	PRO-PRORS-16U-01	Microsoft	2	1,950	3	5,850	
				Subtotal	37,066	5,850	
Client Hardware							
DL360 G3 X3.06GHz 512KB 512MB 1P RCK US	322470-001		1	2,549	2	5,098	
Two integrated Gigabit NIC, Integrated Smart Array Controller							
1GB 133MHz SDRAM DIMM Memory (2x512MB)	201694-B21		1	600	1	600	
X3.06GHz 533MHz Proc Opt Kit DL360G3 WW	322472-B21		1	699	2	1,398	
S5500 15 carbon / silver monitor	261602-001		1	129	1	129	
Scroll Mouse-Carbon	231947-B21		1	5	1	5	
PS/2 Easy Access Internet Keyboard	265977-001		1	12	1	12	
36GB 10K U320 Pluggable Hard Drive WW (OS)	286713-B22		1	299	2	598	
HP CP 3Y 4H 24x7 HW 300 Srs 4-Hour 24 x7 Coverage 3 Years	162657-002		1	949	2	1,898	
				Subtotal	9,738	1,898	
Client Software							
Microsoft Windows 2000 Server	C11-00821	Microsoft	2	738	2	1,476	
				Subtotal	1,476	0	
User Connectivity							
Netgear GS508T 8 port Copper Gigabit Switch	1058966	Netgear	3	485	3	1,455	
7ft CAT 5e Network Patch Cables	CBLC57	LanAdapters	4	1	6	6	
				Subtotal	1,461	0	
Large Purchase and Net 30 discount (See Note 1)	16.0%		1				
					(\$29,367)	(\$2,188)	
				Total	\$194,181	\$17,338	
Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark pricing specifications. If you find that the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.				Three-Year Cost of Ownership: \$211,519			
				tpmC Rating: 60,364			
				\$ / tpmC: \$3.51			
Pricing: 1=HP 2=Microsoft 3=Compuplus.com 4=LanAdapters.com							
Note 1 = Discount based on HP Direct guidance with large purchase and Net 30 discount.							

Numerical Quantities Summary

MQTH, Computed Maximum Qualified Throughput

60,364 tpmC

Response Times (in seconds)	Average	90%	Maximum
New-Order	0.34	0.55	7.65
Payment	0.25	0.42	4.47
Order-Status	0.26	0.42	8.29
Delivery (interactive portion)	0.10	0.11	0.14
Delivery (deferred portion)	0.21	0.35	1.44
Stock-Level	1.17	1.57	4.43
Menu	0.10	0.11	0.34

Transaction Mix, in percent of total transaction

New-Order	44.96%
Payment	43.01%
Order-Status	4.01%
Delivery	4.02%
Stock-Level	4.01%

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	18.02/12.04	18.20/120.32
Payment	3.00/0.00	3.02/12.02	3.20/120.31
Order-Status	2.00/0.00	2.02/10.01	2.17/100.32
Delivery (interactive)	2.00/0.00	2.02/5.03	2.17/50.31
Stock-Level	2.00/0.00	2.02/5.02	2.15/50.31

Test Duration

Ramp-up time	20 minutes
Measurement interval	120 minutes
Transactions (all types) completed during measurement interval	16,039,885
Ramp down time	5 minutes

Checkpointing

Number of checkpoints	4
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 Hewlett-Packard Company. The benchmark was developed and engineered by Hewlett-Packard Company. Testing took place at HP 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.

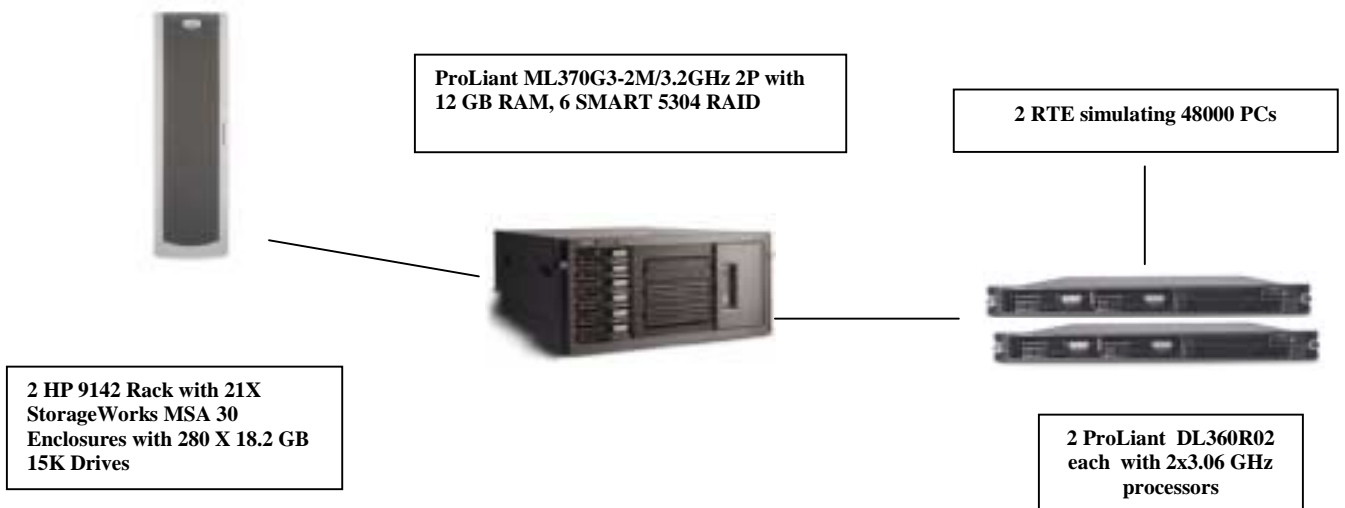
Appendix C contains the tunable parameters to for the database, the operating system, and the transaction monitor.

Configuration Items

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

The configuration diagrams for both the tested and priced systems are included on the following pages.

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: 280 18.2GB 15K drives for the database data connected to 5 SMART 5304 Raid controllers, 1 18 GB 10K drive for the operating system, and 10 36 GB 15K drives for the transaction log connected to SMART 5304 Raid controller..

Benchmarked Configuration:

Integrated Ultra3 SCSI Controller

LOGICAL DRIVE C: Total Capacity = 18 GB
Microsoft 2003 Enterprise Server, MSSQL_tpcc_root.mdf

SMART-5304 Controller, Slot 6, Array A

LOGICAL DRIVE E: Total Capacity = 120 GB RAID 0+1
MSSQL_tpcc_log

SMART-5304 Controller, Slot 1, Array A

LOGICAL DRIVE F: Total Capacity = 78 GB RAID 0
MSSQL_big1

SMART-5304 Controller, Slot 1, Array B

LOGICAL DRIVE Y: Total Capacity = 300 GB RAID 0+1
Backup1

SMART-5304 Controller, Slot 2, Array A

LOGICAL DRIVE G: Total Capacity = 78 GB RAID 0
MSSQL_big2

SMART-5304 Controller, Slot 2, Array B

LOGICAL DRIVE Z: Total Capacity = 300 GB RAID 0+1
Backup2

SMART-5304 Controller, Slot 3, Array A

LOGICAL DRIVE H: Total Capacity = 78 GB RAID 0
MSSQL_big3

SMART-5304 Controller, Slot 4, Array A

LOGICAL DRIVE I: Total Capacity = 78 GB RAID 0
MSSQL_big4

SMART-5304 Controller, Slot 5, Array A

LOGICAL DRIVE J: Total Capacity = 150 GB RAID 0
MSSQL_misc1

Priced Configuration vs. Measured Configuration:

The measured and priced configuration 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 1.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the minimum key value for these new rows.

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

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.

No partitioning was used in this benchmark.

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.

No replications, duplications or additional attributes were used in this benchmark.

Clause 2 Related Items

Random Number Generation

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

In the Benchcraft 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 Benchcraft 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 affected 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. The auditor manually exercised each specification on a representative HP ProLiant web 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%
	Average items per order	10.00
Payment	Home warehouse payments	85.02%
	Remote warehouse payments	14.99%

Statistic		Value
	Accessed by last name	60.01%
Order Status	Accessed by last name	60.02%
Transaction Mix	New Order	44.96%
	Payment	43.01%
	Order status	4.01%
	Delivery	4.02%
	Stock level	4.01%

Queuing Mechanism

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

Microsoft COM+ on each client machine served as the queuing mechanism to the database. Each delivery request was submitted to Microsoft COM+ asynchronously with control being returned to the client process immediately and the deferred delivery part completing asynchronously.

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.

Consistency

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

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

A run was executed under full load lasting over two hours and included a checkpoint.

The 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 nine were executed using shell scripts to issue queries to the database. Each script included timestamps to demonstrate the concurrency of operations. The results of the queries were captured to files. The captured files were verified by the auditor to demonstrate that 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.

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

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 full database was started, but only 490 of the warehouses were accessed for this test.
- 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 4900 users.
- The test was allowed to run for a minimum of 10 minutes.
- One log disk was removed from the server.
- 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 a 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 server. A new data disk was inserted into the data drive cabinet. After the RAID recovery process finished, the system was rebooted and Microsoft SQL Server was started.
- The database was restored from backup and the transaction log dump was applied.
- 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 13 and 14 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 executed on a fully scaled database of 4800 warehouses under a full load of 48000 users. The following steps were executed:

- 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 48000 users.
- The test was allowed to run for a minimum of 10 minutes.
- A checkpoint was performed.
- The system crash and loss of memory were induced by switching the power off. The power cords were then physically removed from the SUT. No battery backup or Uninterruptible Power Supply (UPS) were used to preserve the contents of memory.
- The RTE was shutdown.
- 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 Server

Table	Cardinality as built
Warehouse	4,800
District	48,000
Customer	144,000,000
History	144,000,000
Orders	144,000,000
New Order	43,200,000
Order Line	1,439,997,837
Stock	480,000,000
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 6 SMART-5304 Array controllers with 4 SCSI channels. Each controller is capable of accessing up to 14 disk drives per channel, and supports RAID 0, RAID 0+1, and RAID 5 per each logical volume configured. The data tables were stored on 5 RAID arrays of (42) 18.2GB 15K drives each with 1 arrays RAID 0 logical drives on each controller. Each of these controllers also housed a RAID 0+1 volume used for backup of the database. The other SMART Array controller had one array consisting of (10) 36 GB 15K drives with a RAID 0+1 logical volume for the database log. The Array Accelerators on the data controllers were configured as 100% write cache and were enabled for all logical drives on these controllers. The controller for the transaction log had the cache disabled. 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, DL/I, COBOL read/write used to implement the TPC-C transaction. If more than one interface/access language is used to implement TPC-C, each interface/access language must be described and a list of which interface/access language is used with which transaction type must be disclosed.

Microsoft SQL Server 2000 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 database was not replicated.

60 Day Space

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

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)*.
- Transactions were run against the database with a full load of users.
- The free space was again queried using *dbcc sqlperf(logspace)*.
- The space used was calculated as the difference between the first and second query.
- The number of NEW-ORDERS was verified from the difference in the sum(d_next_o_id) taken from before and after the 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 60-day space requirement is shown in Appendix D.

Clause 5 Related Items

Throughput

Measured tpmC must be reported

Measured tpmC 60,364 tpmC
Price per tpmC \$3.51 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	0.34	0.55	7.65
Payment	0.25	0.42	4.47
Order-Status	0.26	0.42	8.29
Interactive Delivery	0.10	0.11	0.14
Deferred Delivery	0.21	0.35	1.44
Stock-Level	1.17	1.57	4.43
Menu	0.10	0.11	0.34

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	18.20
Payment	3.00	3.02	3.20
Order-Status	2.00	2.02	2.17
Interactive Delivery	2.00	2.02	2.17
Stock-Level	2.00	2.02	2.15

Table 5.4: Think Times

Type	Minimum	Average	Maximum
New-Order	0.00	12.04	120.32
Payment	0.00	12.02	120.31
Order-Status	0.00	10.01	100.32
Interactive Delivery	0.00	5.03	50.31
Stock-Level	0.00	5.02	50.31

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

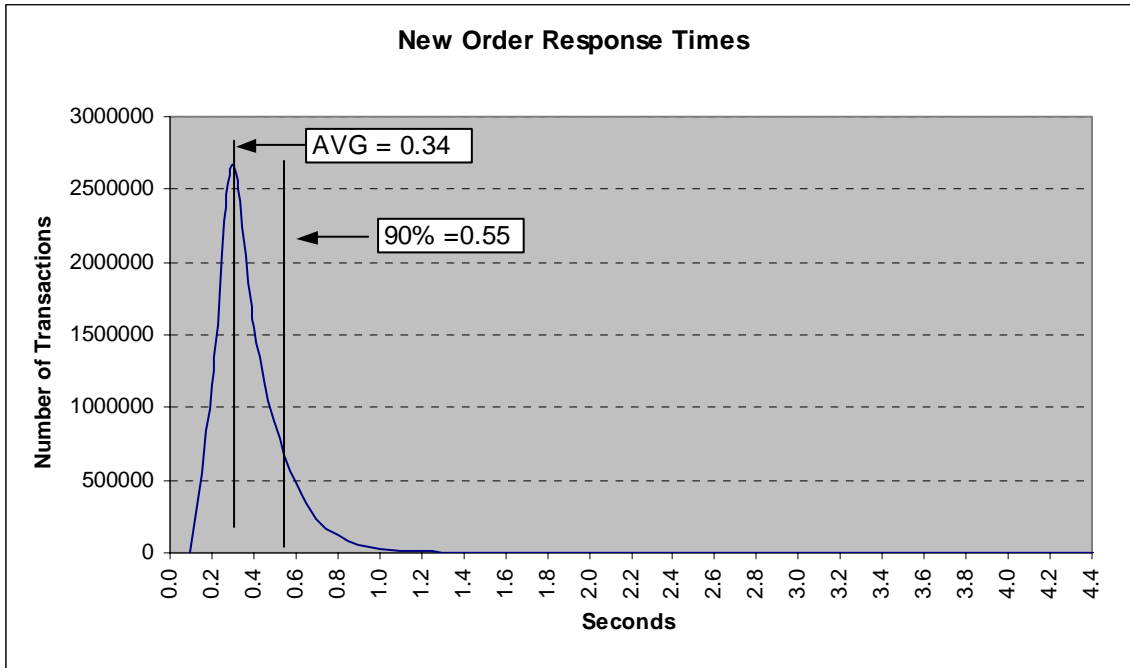


Figure 4. Payment Response Time Distribution

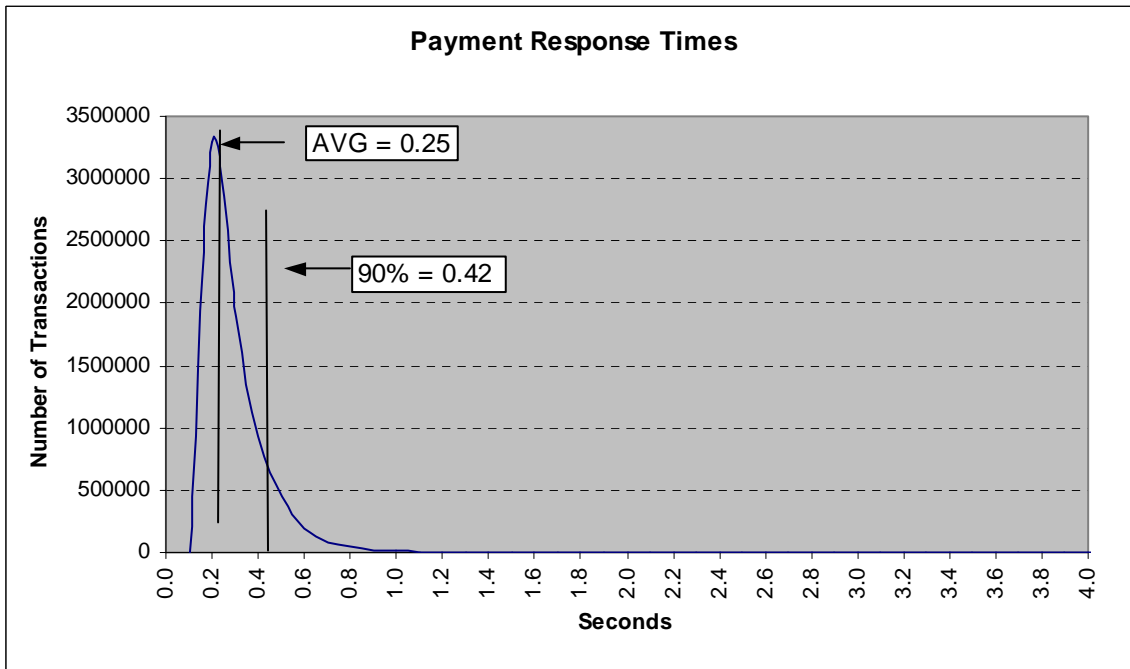


Figure 5. Order Status Response Time Distribution

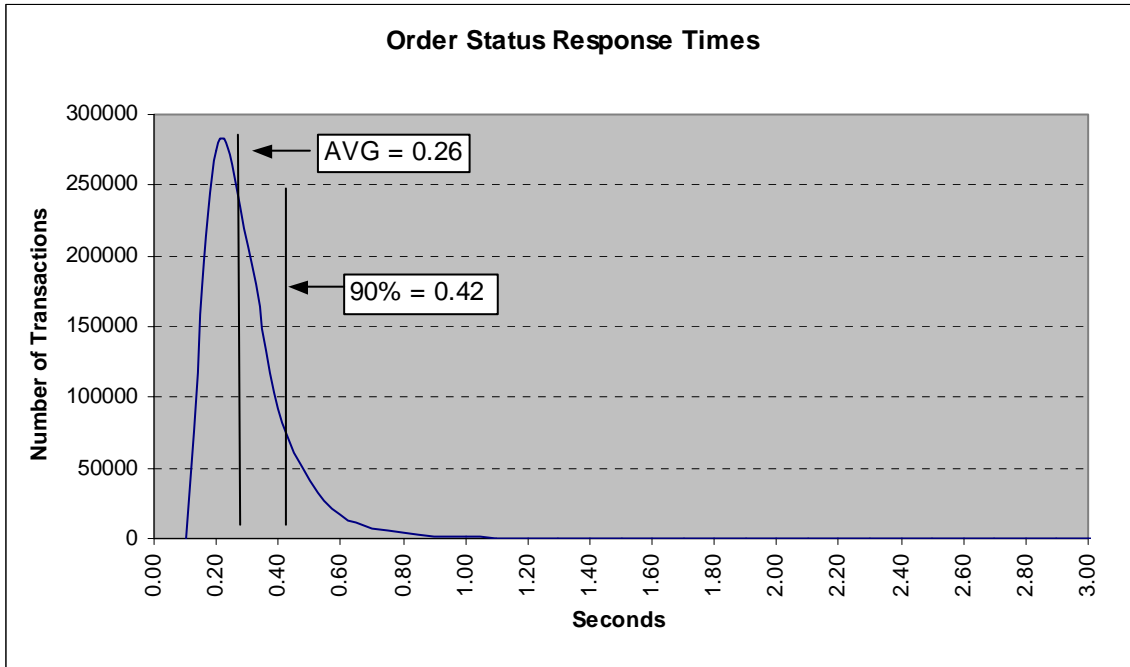


Figure 6. Delivery Response Time Distribution

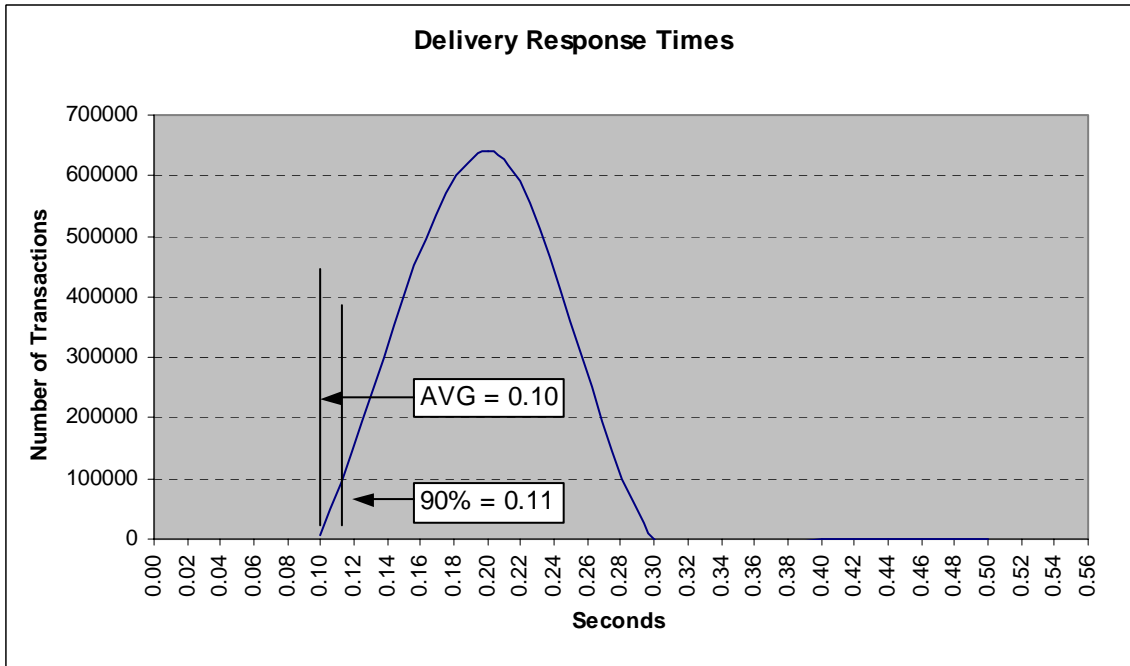


Figure 7. Stock Level Response Time Distribution

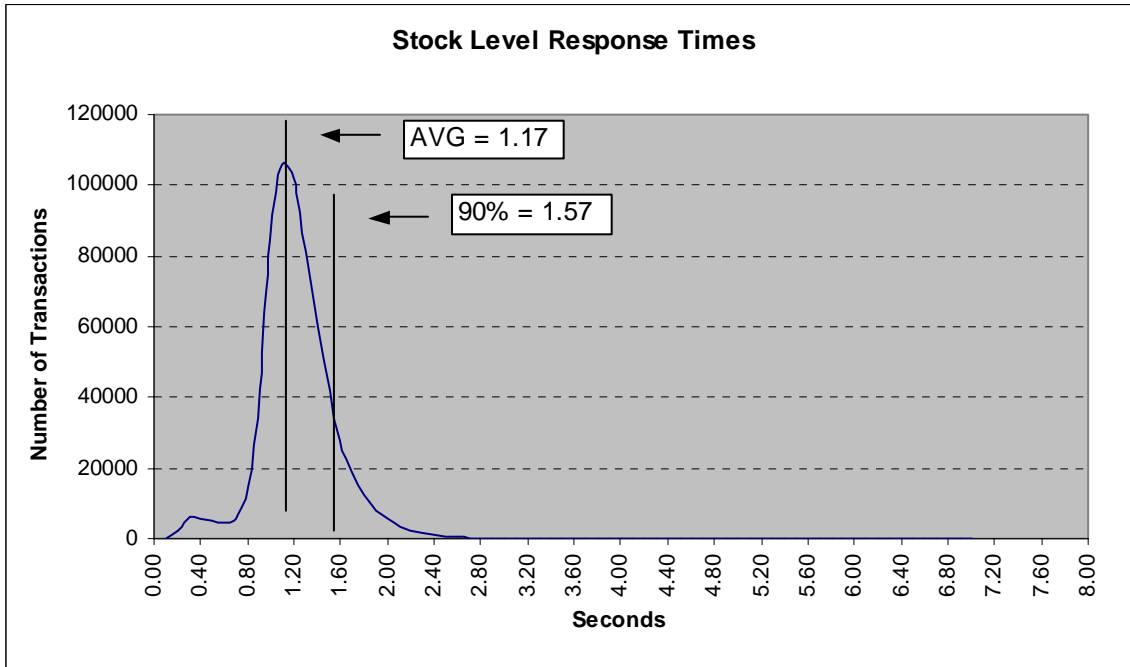


Figure 8. Response Time vs. Throughput

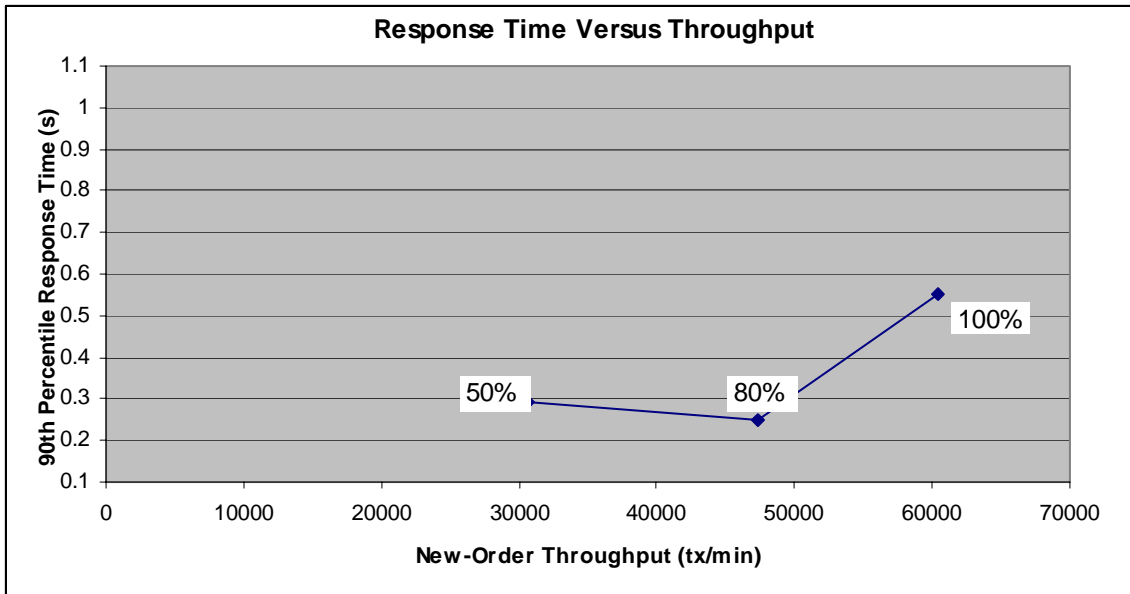


Figure 9. New Order Think Time Distribution

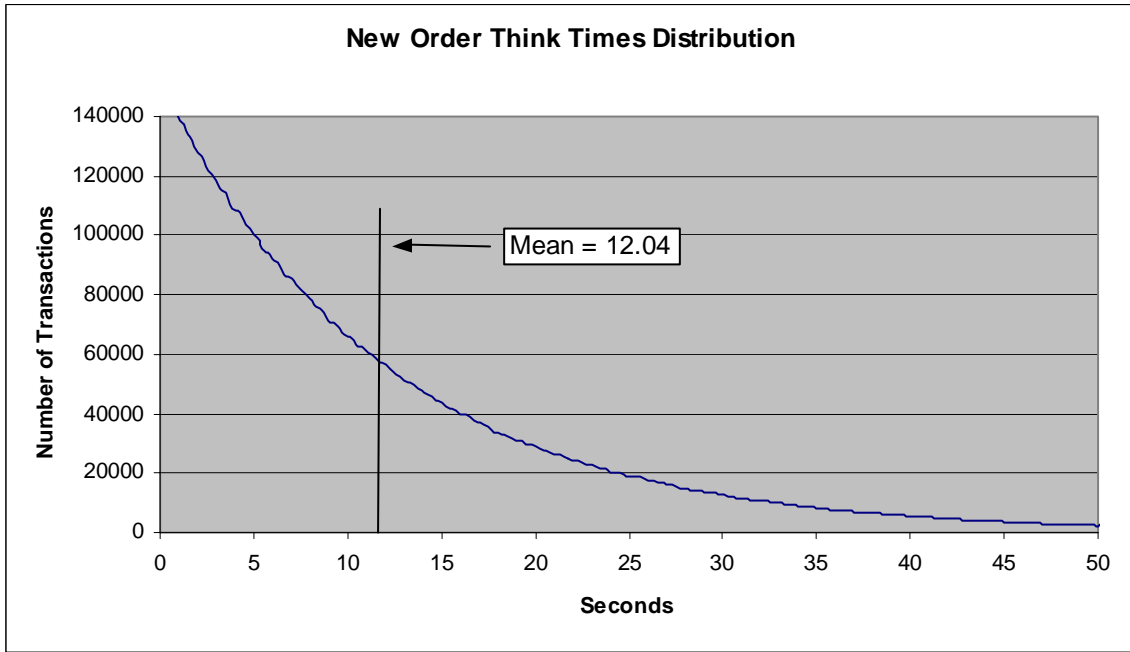
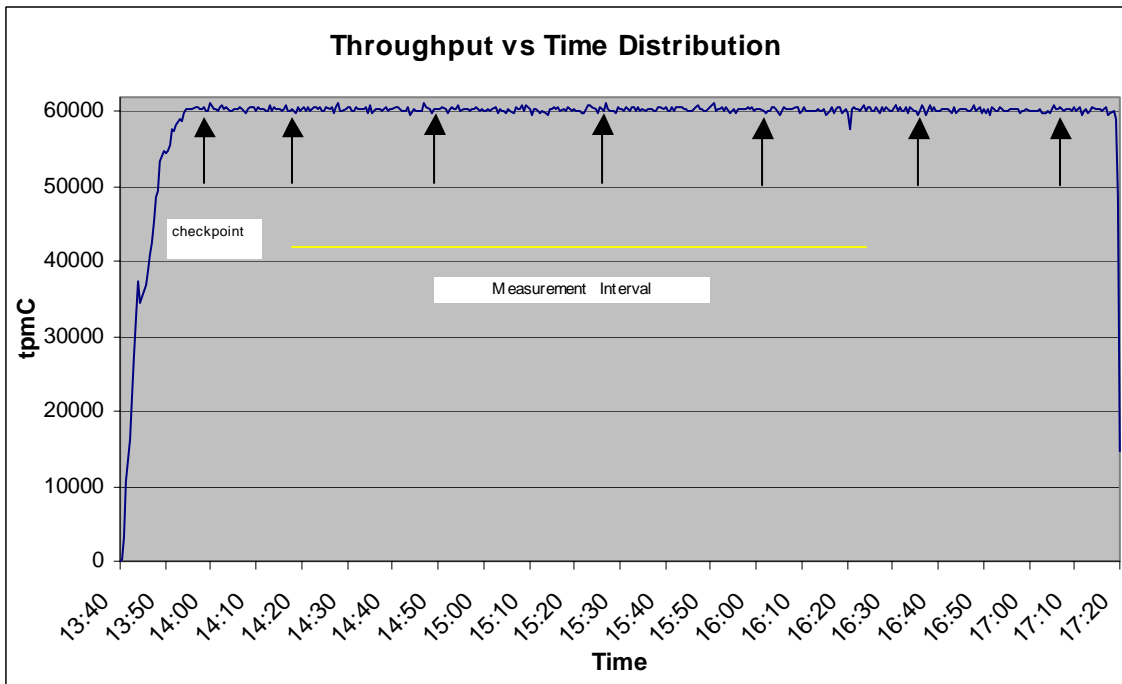


Figure 10. Throughput vs. Time Distribution



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 Ethernet LANs using DBLIB and RPC calls.

To perform checkpoints at specific intervals, the SQL Server *recovery interval* was set to 110 and a script was written to schedule multiple checkpoints at specific intervals. The script included a wait time between each checkpoint equal to 30 minutes so that the checkpoint interval was an integral multiple of the measurement interval, which was 120 minutes. The checkpoint script was started manually after the RTE had all users logged in and the database had achieved steady state.

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 is depicted on the graph in Figure 10.

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 120 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 order lines per New-Order transaction must be disclosed. The percentage of remote Payment transactions must be disclosed. The percentage of customer selections by customer last name in the Payment and Order-Status transactions must be disclosed. The percentage of Delivery transactions skipped due to there being fewer than necessary orders in the New-Order table must be disclosed.

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	85.02%
	Remote warehouse payments	14.99%
	Accessed by last name	60.01%
Delivery	Skipped transactions (interactive)	0
	Skipped transactions (deferred)	0
Order Status	Accessed by last name	60.02%
Transaction Mix	New Order	44.96%
	Payment	43.01%
	Order status	4.01%
	Delivery	4.02%
	Stock level	4.01%

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 20 minutes after the start of the ramp-up. Subsequent checkpoints occurred every 30 minutes. The measurement interval contains four checkpoints.

Checkpoint Duration

The start time and duration in seconds of at least the four longest checkpoints during the Measurement Interval must be disclosed.

Checkpoint Start Time	Duration
14:17:10 p.m.	10 minutes, 21 seconds
14:47:08 p.m.	10 minutes, 4 seconds
15:17:06 p.m.	11 minutes, 13 seconds
15:47:04 p.m.	10 minutes, 39 seconds

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 Benchcraft RTE. Benchcraft 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 1 HP ProLiant server. This driver machine emulated the users web browsers.

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, 2 driver (RTE) machine was connected through a 10/100 switch to the client machines at 100Mbps, thus providing the path from the RTE to the clients. The server (SUT) was connected to the clients through the Gigabit network switch.

The priced configuration was connected in the same manner as the tested configuration.

Operator Intervention

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

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

Clause 7 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 3 year price of the entire configuration must be reported, including: hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

The 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 | 60,364 tpmC |
| • Price per tpmC | \$3.51 per tpmC |
| • Availability | March 17, 2004 |

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:

- 2 Microsoft Windows 2000 Server
- 1 Microsoft 2003 Enterprise Server
- 1 Microsoft SQL Server 2000 Enterprise Edition (per processor)
- 1 Microsoft Visual C++
- HP 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 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.

TPC Benchmark C Full Disclosure Report is available at www.tpc.org.



PERFORMANCE METRICS INC.
TPC Certified Auditors

March 1, 2004

Mr. Paul Cao
Hewlett-Packard Company
20555 SH 249
Houston, TX 77070

I have verified by remote the TPC Benchmark™ C for the following configuration:

Platform: HP ProLiant ML370G3T-2M- 2P
Database Manager: Microsoft SQL Server 2000 Enterprise Edition
Operating System: Microsoft Windows 2003 Enterprise Edition
Transaction Monitor: COM+

System Under Test: HP ProLiant ML370G3T-2M- 2P with:				
CPU's	Memory	Disks (total)	90% Response	TpmC
2 Xeon @ 3.2 Ghz	Main: 12 GB	281 @ 18.2GB 10 @ 72 GB	0.55	60,364.12

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.
- The database was properly scaled with 4800 warehouses.
- The ACID properties were successfully demonstrated.
- Input data was generated according to the specified percentages.
- Eight hours of mirrored log space was present on the tested system.
- Eight hours of growth space for the dynamic tables was present on the tested system.
- The data for the 60 days space calculation was verified.
- The controller cache for the log disks was disabled.
- The steady state portion of the test was 120 minutes.
- More than one checkpoint was taken before the measured interval opened.
- Four checkpoints were completed inside the measured interval.

PERFORMANCE METRICS INC.
TPC Certified Auditors

- The system pricing was checked for major components and maintenance.
- Third party quotes were verified for compliance.

Auditor Notes: None

Sincerely,

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

Lorna Livingtree
Auditor

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 Copyright
 * 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
    STDMETHODCALLTYPE CanBePooled() { return
m_bCanBePooled; }
    STDMETHODCALLTYPE Activate() { return S_OK; }
    // we don't support COM Services
    transactions (no enlistment)
    STDMETHODCALLTYPE Deactivate() { /*
nothing to do */ }

// IObjectConstruct
    STDMETHODCALLTYPE Construct(IDispatch * pUnk);

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
(levine@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
);

```

WebcInt.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>
{{{
}}}

#####
#####

```

WebcInt.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>
{{{
}}}
```

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

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 odbc32.lib
odbc32.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 /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
odbc32.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 /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 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 ntwdblib.lib kernel32.lib
user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib /nologo /subsystem:windows /dll /debug
/machine:I386 /out:".bin\tpcc_dblib.dll"
/pdbtype:sept

```

```
!ENDIF
```

```
# Begin Target
```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

db_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
odbc32.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
odbc32.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 /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
odbc32.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
odbc32.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 /MD /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 /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
odbc32.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 odbc32.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_BUF_OVERFLOW 25
//Buffer overflow during receive
// TPC-W error types
#define ERR_TYPE_TPCW_CONN 50
//Benchcraft connection class

```

```

#define ERR_TYPE_TPCW_HTML 51
//error from TpcwHtml dll
#define ERR_TYPE_TPCW_USER 52
//error from TPC-W user class
#define ERR_TYPE_TPCW_ENG_BASE 53
#define ERR_TYPE_TPCW_ENG_OS 54
#define ERR_TYPE_HTML_RESP 55
#define ERR_TYPE_TPCW_ODBC 56
#define ERR_TYPE_SCHANNEL 57

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

class CBaseErr
{
public:
    CBaseErr(LPCTSTR szLoc = NULL)
    {
        m_idMsg =
INV_ERROR_CODE;

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

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

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

        if (szLoc)

```

```

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

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

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

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

    if (szStr)
        j = wsprintf(szTmp,
"%s\n",szStr);
    if (ErrorNum() != INV_ERROR_CODE)
        j += wsprintf(szTmp+j,
"Error = %d\n", ErrorNum());
    if (m_szLoc)
        j += wsprintf(szTmp+j,
"Location = %s\n", GetLocation());
    j += wsprintf(szTmp+j, "%s\n",
ErrorText());
    ::MessageBox(hwnd, szTmp,
m_szApp, MB_OK);
}

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

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

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

```

```

}; //short m_errType;

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

    CSocketErr(Action eAction, LPCTSTR
szLocation = NULL);

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

    Action m_eAction;
    char *m_szErrorText;

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

class CSystemErr : public CBaseErr
{
public:
    enum Action
    {
        eNone = 0,
        eTransactNamedPipe,
        eWaitNamedPipe,
        eSetNamedPipeHandleState,
        eCreateFile,
        eCreateProcess,
        eCallNamedPipe,
        eCreateEvent,
        eCreateThread,
        eVirtualAlloc,
    };

```

```

        eReadFile = 10,
        eWriteFile,
        eMapViewOfFile,
        eCreateFileMapping,
        eInitializeSecurityDescriptor,
        eSetSecurityDescriptorDacl,
        eCreateNamedPipe,
        eConnectNamedPipe,
        eWaitForSingleObject,
        eRegOpenKeyEx,
        eRegQueryValueEx = 20,
        ebeginthread,
        eRegEnumValue,
        eRegSetValueEx,
        eRegCreateKeyEx,
        eWaitForMultipleObjects,
        eRegisterClassEx,
        eCreateWindow,
        eCreateSemaphore,
        eFSeek,
        eFRead,
        eFWrite,
        eTmpFile,
        eSetFilePointer,
        eNew,
    };

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

    Action m_eAction;

private:
    char m_szMsg[ERR_MSG_BUF_SIZE];
};

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

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

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

    int ErrorType() {return ERR_BUF_OVERFLOW;};

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

```

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.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

#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_RI
    GHT |
    ES_NUMBER,WS_EX_RTLREADING

    EDITTEXT        ED_IIS_THREAD_TIMEOUT,164,291,34,12,ES_RIGHT |
    ES_NUMBER,
    WS_EX_RTLREADING

    EDITTEXT        ED_IIS_LISTEN_BACKLOG,164,305,34,12,ES_RIGHT |
    ES_NUMBER,
    WS_EX_RTLREADING

    DEFPUSHBUTTON   "OK",IDOK,53,331,50,14
    PUSHBUTTON      "Cancel",IDCANCEL,119,331,50,14

    EDITTEXT        IDC_PATH,106,26,91,13,ES_AUTOHSCROLL | ES_READONLY

    LTEXT           "Number of Delivery
Threads:",IDC_STATIC,35,45,115,12

    LTEXT           "Max Number of
Connections:",IDC_STATIC,35,73,115,12

    RTEXT           "Version
4.11",IDC_VERSION,120,4,89,9

    LTEXT           "IIS Max Thread Pool
Limit:",IDC_STATIC,36,263,115,12

    LTEXT           "Web Service Backlog Queue
Size:",IDC_STATIC,36,277,115,
12

    LTEXT           "IIS Thread Timeout
(seconds):",IDC_STATIC,36,291,115,12

    LTEXT           "IIS Listen
Backlog:",IDC_STATIC,36,307,115,10

    GROUPBOX       "Database
Interface",IDC_STATIC,35,208,163,27,WS_GROUP

    LTEXT           "Installation
directory:",IDC_STATIC,35,29,71,10

    GROUPBOX       "Transaction
Monitor",IDC_STATIC,33,90,165,37

    LTEXT           "Server
Name:",IDC_STATIC,35,155,56,8
```

```

LTEXT      "User ID:", IDC_STATIC, 35, 168, 60, 8
LTEXT      "User
Password:", IDC_STATIC, 35, 181, 83, 8
LTEXT      "Database
Name:", IDC_STATIC, 35, 194, 54, 8
GROUPBOX   "SQL Server Connection
Properties", IDC_STATIC, 22, 139, 187,
102
GROUPBOX   "Web Client
Properties", IDC_STATIC, 22, 15, 187, 118
GROUPBOX   "IIS
Settings", IDC_STATIC, 22, 247, 187, 79
LTEXT      "Max Pending
Deliveries:", IDC_STATIC, 35, 59, 115, 12
END

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

IDD_DIALOG3 DIALOG DISCARDABLE 0, 0, 91, 40
STYLE DS_SYSMODAL | DS_MODALFRAME | DS_3DLOOK |
DS_CENTER | WS_CAPTION
CAPTION "Installing TPC-C Web Client"
FONT 12, "Arial Black"
BEGIN
CONTROL
"Progress1", IDC_PROGRESS1, "mctl_progress32", WS_BORD
ER,
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
"..\\..\\tm_tuxedo_dll\\bin\\tpcc_tuxedo.dll"

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

////////////////////////////////////
////////////////////////////////////
//

```

```

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

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

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

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

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

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

```

install_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,
&pCatalogCollectionApp);
if (!SUCCEEDED(hr)) goto Error;

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

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

hr = pCOMAdminCat-
>InstallComponent(bstrTemp,
bstrTemp2,

```

```

bstrTemp3,

bstrTemp4);
if (!SUCCEEDED(hr)) goto Error;

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

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

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

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

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

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

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

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

    bstrTemp = "MaxPoolSize";

```



```

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

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

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

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

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

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

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

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

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

```

```

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

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

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

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

                        pCatalogObjectMethod->Release();
                        pCatalogObjectMethod = NULL;

                        lCountMethod-
-;
                }

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

                pCatalogObjectItf-
>Release();
                pCatalogObjectItf =
NULL;

                lCountItf--;
        }

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

```

```

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

        pCatalogCollectionApp->Release();
        pCatalogCollectionApp = NULL;

        pCatalogCollectionCo->Release();
        pCatalogCollectionCo = NULL;

        pCatalogCollectionItf->Release();
        pCatalogCollectionItf = NULL;

        pCatalogCollectionMethod->Release();
        pCatalogCollectionMethod = NULL;

Error:
        CoUninitialize();

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

```

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 odbc32.lib
!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 odbc32.lib
odbc32.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\txnolog.lib wsoc32.lib
kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbc32.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

```

```

odbc32.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\txnolog.lib wsoc32.lib
kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbc32.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 odbc32.lib
odbc32.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\txnolog.lib wsoc32.lib
kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbc32.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_dl"
# PROP BASE Intermediate_Dir "isapi_dl"

```

```

# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /GX /Zi /Od /D
"_DEBUG" /D "WIN32" /D "_WINDOWS" /FR /YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /GX /Zi /O2 /D "NDEBUG" /D
"ICECAP" /D "WIN32" /D "_WINDOWS" /FR /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "DEBUG" /mktyplib203 /o
"NUL" /win32
# ADD MTL /nologo /D "DEBUG" /mktyplib203 /o "NUL"
/win32
# ADD BASE RSC /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
odbc32.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\txnolog.lib wsoc32.lib
kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbc32.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

```

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(
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
odbc32.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
odbc32.lib /nologo /subsystem:windows /dll
/machine:I386 /out:".bin\tpcc_com.dll"

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

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

```

```
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbc32.lib
odbc32.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
odbc32.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.def

```
LIBRARY TPCC.DLL

EXPORTS

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

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 d_id; //district id
    assigned at welcome form
    int iSyncId;
    int //synchronization id
    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_TO_LONG,
ERR_PAYMENT_MISSING_CDI_KEY,
ERR_PAYMENT_MISSING_CID_CLT,
ERR_PAYMENT_MISSING_CID_KEY,
ERR_PAYMENT_MISSING_CLT,
ERR_PAYMENT_MISSING_CLT_KEY,
ERR_PAYMENT_MISSING_CWI_KEY,
ERR_PAYMENT_MISSING_DID_KEY,
ERR_PAYMENT_MISSING_HAM_KEY,

ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
ERR_STOCKLEVEL_THRESHOLD_INVALID,
ERR_STOCKLEVEL_THRESHOLD_RANGE,
ERR_VERSION_MISMATCH,
ERR_W_ID_INVALID
};

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

    CWEBCLNT_ERR(WEBERROR Err, char
*szTextDetail, DWORD dwSystemErr)

```

```

{
    m_Error = Err;
    m_szTextDetail = new
char[strlen(szTextDetail)+1];
    strcpy( m_szTextDetail,
szTextDetail );
    dwSystemErr =
        m_SystemErr =
        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 APSTUDIO_INVOKED
////////////////////////////////////
////////////////////////////////////
//
// 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 Copyright
 * 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(dlllexport) CTPCC_COM*
CTPCC_COM_new(BOOL);

typedef CTPCC_COM* (TYPE_CTPCC_COM)(BOOL);

```

tpcc_com_all_i.c

```

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

```

```

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

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

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:14 2001
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf (OptLev=i2), 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 =
{1,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,0x71,0x00,0x
C0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x0
,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,0
x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

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

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

```

```

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:14 2001
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf (OptLev=i2), 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 =
{1,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,0x71,0x00,0x
C0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x0
0,0xC0,0x4F,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,0
x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

```

tpcc_com_all_i.c

```

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

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

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

```

```

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:14 2001
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf (OptLev=i2), 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 x1;
    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 =
{1,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,0x71,0x00,0x
C0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x0
0,0xC0,0x4F,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,0
x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

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

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

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

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:14 2001
*/

```

```

/* 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 =
{1,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,0x71,0x00,0x
C0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x0
0,0xC0,0x4F,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,0
x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

```

tpcc_com_all_i.c

```

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

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

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

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:14 2001
*/

```

```

/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf (OptLev=i2), 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 =
{1,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

```

```

#endif !_MIDL_USE_GUIDDEF_

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

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

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

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

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

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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

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

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

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

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:14 2001
*/
/* 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_HEADER( )

#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 =
{1,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

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

```

```

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

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

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

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

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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

```

tpcc_com_all.h

```

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

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

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:14 2001
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf (OptLev=i2), 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_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 /* __TPCCLib_LIBRARY_DEFINED__ */

/* Additional Prototypes for ALL interfaces */
/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

tpcc_com_all_i.c

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

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

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

```

```

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:14 2001
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run),
ms_ext, c_ext
error checks: allocation ref bounds_check enum
stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany),
__declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@MIDL_FILE_HEADING( )

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

#ifdef __cplusplus
extern "C"{
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

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

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

DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

```

```

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
b7,b8) \
    const type name =
{1,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,0x71,0x00,0x
C0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x0
0,0xC0,0x4F,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,0
x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

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

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

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:14 2001
*/
/* Compiler settings for .\src\tpcc_com_all.idl:

```

```

Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b
run,appending), ms_ext, c_ext, robust
error checks: allocation ref bounds_check enum
stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany),
__declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@MIDL_FILE_HEADING( )

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

#ifdef __cplusplus
extern "C"{
#endif

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

#ifdef _MIDL_USE_GUIDDEF_

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

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

DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
b7,b8) \
    const type name =
{1,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, 0x00
C0, 0x4F, 0xBF, 0xE0, 0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder, 0x975BAABF, 0x84A7, 0x11D2, 0xBA, 0x47, 0x0
0, 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, 0
x00, 0xC0, 0x4F, 0xBF, 0xE0, 0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

```

tpcc_com_all.rc

```

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

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

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

```

```

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

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

```



```
//
1 TYPELIB "tpcc_com_all.tlb"

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

tpcc_com_all.i.c

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

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

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

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:14 2001
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf (OptLev=i2), 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 =
{1,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,0x71,0x00,0x
C0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x0
0,0xC0,0x4F,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,0
x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
#endif
```

```
}
#endif

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

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

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

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

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:14 2001
*/
/* 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_

#ifdef __cplusplus
#endif
```

```

#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 =
{1,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,0x71,0x00,0x
C0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x0
0,0xC0,0x4F,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,0
x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

```

tpcc_com_all_i.c

```

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

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

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

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:14 2001
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf (OptLev=i2), 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 =
{1,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,0x71,0x00,0x
C0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x0
0,0xC0,0x4F,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,0
x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

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

```

```

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

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

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

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:14 2001
*/
/* 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_HEADERING( )

#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 =
{1,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,0x71,0x00,0x
C0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x0
0,0xC0,0x4F,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,0
x00,0xC0,0x4F,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'
    }
    'TPCC.NewOrder.1'
        ProgID = s

        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'
    }
    'TPCC.OrderStatus.1'
        ProgID = s

        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_p.c

```

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

/* this ALWAYS GENERATED file contains the proxy stub
code */

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:08 2001
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=i2), 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)
#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,{0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0
x00,0x00,0x00,0x46}} */

```

```

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEED6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0
x4F,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
};

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,
    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 */
        FC_AUTO_HANDLE /*          0x33,          */
        /*          0x6c,          */
        Old Flags: object, Oi2 /*
        /* 2 */ NdrFcLong( 0x0 ), /* 0 */
        /* 6 */ NdrFcShort( 0x3 ), /* 3 */
        #ifndef _ALPHA_
        #ifndef _PPC_
        #if !defined(_MIPS_)
        /* 8 */ NdrFcShort( 0x1c ), /* x86 Stack
        size/offset = 28 */
        #else
        NdrFcShort( 0x20 ), /*
        MIPS Stack size/offset = 32 */
        #endif
        #else
        NdrFcShort( 0x20 ), /*
        PPC Stack size/offset = 32 */
        #endif
        #else
        NdrFcShort( 0x28 ), /*
        Alpha Stack size/offset = 40 */
        #endif
        /* 10 */ NdrFcShort( 0x0 ), /* 0 */
        /* 12 */ NdrFcShort( 0x8 ), /* 8 */
        /* 14 */ 0x7, /* Oi2 Flags: srv must
        size, clt must size, has return, */
        0x3, /*
        3 */

        /* Parameter txn_in */
        /* 16 */ NdrFcShort( 0x8b ), /* Flags: must size,
        must free, in, by val, */
        #ifndef _ALPHA_
        #ifndef _PPC_
        #if !defined(_MIPS_)
        /* 18 */ NdrFcShort( 0x4 ), /* x86 Stack
        size/offset = 4 */
        #else
        NdrFcShort( 0x8 ), /*
        MIPS Stack size/offset = 8 */
        #endif
        #else
        NdrFcShort( 0x8 ), /*
        PPC Stack size/offset = 8 */
        #endif
        #else
        NdrFcShort( 0x8 ), /*
        Alpha Stack size/offset = 8 */

```

```

#endif
/* 20 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Parameter txn_out */
/* 22 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 24 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
#else
NdrFcShort( 0x18 ), /*
MIPS Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
PPC Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
Alpha Stack size/offset = 24 */
#endif
/* 26 */ NdrFcShort( 0x3da ), /* Type
Offset=986 */

/* Return value */
/* 28 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 30 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
#else
NdrFcShort( 0x1c ), /*
MIPS Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /*
PPC Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x20 ), /*
Alpha Stack size/offset = 32 */
#endif
/* 32 */ 0x8, /* FC_LONG */
0 /*
*/

/* Procedure Payment */
/* 34 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
*/
Old Flags: object, Oi2 /*
/* 36 */ NdrFcLong( 0x0 ), /* 0 */
/* 40 */ NdrFcShort( 0x4 ), /* 4 */
#ifndef _ALPHA_
#ifndef _PPC_

```

```

#if !defined(_MIPS_)
/* 42 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
#else
NdrFcShort( 0x20 ), /*
MIPS Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x20 ), /*
PPC Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x28 ), /*
Alpha Stack size/offset = 40 */
#endif
/* 44 */ NdrFcShort( 0x0 ), /* 0 */
/* 46 */ NdrFcShort( 0x8 ), /* 8 */
/* 48 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

/* Parameter txn_in */

/* 50 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 52 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
#else
NdrFcShort( 0x8 ), /*
MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 54 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Parameter txn_out */

/* 56 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 58 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
#else
NdrFcShort( 0x18 ), /*
MIPS Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
PPC Stack size/offset = 24 */

```

```

#endif
#else
NdrFcShort( 0x18 ), /*
Alpha Stack size/offset = 24 */
#endif
/* 60 */ NdrFcShort( 0x3da ), /* Type
Offset=986 */

/* Return value */

/* 62 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 64 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
#else
NdrFcShort( 0x1c ), /*
MIPS Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /*
PPC Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x20 ), /*
Alpha Stack size/offset = 32 */
#endif
/* 66 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure Delivery */

/* 68 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 70 */ NdrFcLong( 0x0 ), /* 0 */
/* 74 */ NdrFcShort( 0x5 ), /* 5 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 76 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
#else
NdrFcShort( 0x20 ), /*
MIPS Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x20 ), /*
PPC Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x28 ), /*
Alpha Stack size/offset = 40 */
#endif
/* 78 */ NdrFcShort( 0x0 ), /* 0 */
/* 80 */ NdrFcShort( 0x8 ), /* 8 */
/* 82 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

```

```

/* Parameter txn_in */

/* 84 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 86 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
#else
NdrFcShort( 0x8 ), /*
MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 88 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Parameter txn_out */

/* 90 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 92 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
#else
NdrFcShort( 0x18 ), /*
MIPS Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
PPC Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
Alpha Stack size/offset = 24 */
#endif
/* 94 */ NdrFcShort( 0x3da ), /* Type
Offset=986 */

/* Return value */

/* 96 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 98 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
#else
NdrFcShort( 0x1c ), /*
MIPS Stack size/offset = 28 */
#endif

```

```

#else
                                NdrFcShort( 0x1c ), /*
PPC Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x20 ), /*
Alpha Stack size/offset = 32 */
#endif
/* 100 */ 0x8, /* FC_LONG */
                                0x0, /*
0 */

/* Procedure StockLevel */

/* 102 */ 0x33, /* FC_AUTO_HANDLE */
                                0x6c, /*
Old Flags: object, Oi2 */
/* 104 */ NdrFcLong( 0x0 ), /* 0 */
/* 108 */ NdrFcShort( 0x6 ), /* 6 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 110 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
#else
                                NdrFcShort( 0x20 ), /*
MIPS Stack size/offset = 32 */
#endif
#else
                                NdrFcShort( 0x20 ), /*
PPC Stack size/offset = 32 */
#endif
#else
                                NdrFcShort( 0x28 ), /*
Alpha Stack size/offset = 40 */
#endif
/* 112 */ NdrFcShort( 0x0 ), /* 0 */
/* 114 */ NdrFcShort( 0x8 ), /* 8 */
/* 116 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
                                0x3, /*
3 */

/* Parameter txn_in */

/* 118 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 120 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
#else
                                NdrFcShort( 0x8 ), /*
MIPS Stack size/offset = 8 */
#endif
#else
                                NdrFcShort( 0x8 ), /*
PPC Stack size/offset = 8 */
#endif
#else
                                NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */

```

```

#endif
/* 122 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Parameter txn_out */

/* 124 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 126 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
#else
                                NdrFcShort( 0x18 ), /*
MIPS Stack size/offset = 24 */
#endif
#else
                                NdrFcShort( 0x18 ), /*
PPC Stack size/offset = 24 */
#endif
#else
                                NdrFcShort( 0x18 ), /*
Alpha Stack size/offset = 24 */
#endif
/* 128 */ NdrFcShort( 0x3da ), /* Type
Offset=986 */

/* Return value */

/* 130 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 132 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
#else
                                NdrFcShort( 0x1c ), /*
MIPS Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x1c ), /*
PPC Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x20 ), /*
Alpha Stack size/offset = 32 */
#endif
/* 134 */ 0x8, /* FC_LONG */
                                0x0, /*
0 */

/* Procedure OrderStatus */

/* 136 */ 0x33, /* FC_AUTO_HANDLE */
                                0x6c, /*
Old Flags: object, Oi2 */
/* 138 */ NdrFcLong( 0x0 ), /* 0 */
/* 142 */ NdrFcShort( 0x7 ), /* 7 */
#ifndef _ALPHA_
#ifndef _PPC_

```

```

#if !defined(_MIPS_)
/* 144 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
#else
                                NdrFcShort( 0x20 ), /*
MIPS Stack size/offset = 32 */
#endif
#else
                                NdrFcShort( 0x20 ), /*
PPC Stack size/offset = 32 */
#endif
/* 146 */ NdrFcShort( 0x0 ), /* 0 */
/* 148 */ NdrFcShort( 0x8 ), /* 8 */
/* 150 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
                                0x3, /*
3 */

/* Parameter txn_in */

/* 152 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 154 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
#else
                                NdrFcShort( 0x8 ), /*
MIPS Stack size/offset = 8 */
#endif
#else
                                NdrFcShort( 0x8 ), /*
PPC Stack size/offset = 8 */
#endif
#else
                                NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 156 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Parameter txn_out */

/* 158 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 160 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
#else
                                NdrFcShort( 0x18 ), /*
MIPS Stack size/offset = 24 */
#endif
#else
                                NdrFcShort( 0x18 ), /*
PPC Stack size/offset = 24 */

```

```

#endif
#else
    NdrFcShort( 0x18 ), /*
Alpha Stack size/offset = 24 */
#endif
/* 162 */ NdrFcShort( 0x3da ), /* Type
Offset=986 */

    /* Return value */

/* 164 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifdef _ALPHA_
#ifdef _PPC_
#ifdef _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 */
/*
0 */

    /* Procedure CallSetComplete */

/* 170 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 172 */ NdrFcLong( 0x0 ), /* 0 */
/* 176 */ NdrFcShort( 0x8 ), /* 8 */
#ifdef _ALPHA_
/* 178 */ NdrFcShort( 0x8 ), /* x86, MIPS, PPC Stack
size/offset = 8 */
#else
    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, */
#ifdef _ALPHA_
/* 188 */ NdrFcShort( 0x4 ), /* x86, MIPS, PPC Stack
size/offset = 4 */
#else

```

```

    NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 190 */ 0x8, /* FC_LONG */
0 */
0x0
}
};
static const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString =
{
    0,
    {
        NdrFcShort( 0x0 ), /*
0 */
/* 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 */

FC_STRUCT */
0x15, /*
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 */
FC_END */
0x5b, /*
/*
FC_LONG */
/* 448 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 450 */
0x16, /*
FC_PSTRUCT */
0x3, /*
3 */
/* 452 */ NdrFcShort( 0x8 ), /* 8 */
/* 454 */
0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */
/* 456 */
0x46, /*
FC_NO_REPEAT */
0x5c, /*
FC_PAD */
/* 458 */ NdrFcShort( 0x4 ), /* 4 */
/* 460 */ NdrFcShort( 0x4 ), /* 4 */
/* 462 */ 0x11, 0x0, /* FC_RP */
/* 464 */ NdrFcShort( 0xffffffffd4 ), /* Offset= -
44 (420) */
/* 466 */
0x5b, /*
FC_END */
0x8, /*
FC_LONG */
/* 468 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */
/* 470 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /*
3 */
/* 472 */ NdrFcShort( 0x0 ), /* 0 */
/* 474 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 476 */ NdrFcShort( 0x0 ), /* 0 */
/* 478 */ NdrFcLong( 0xfffffffff ), /* -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( 0xfffffffff ), /* -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 */
0x36, /*
FC_POINTER */
/* 578 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 580 */
0x11, 0x0, /*
FC_RP */
/* 582 */ NdrFcShort( 0xffffffffd4 ), /* Offset= -
44 (538) */
/* 584 */
0x2E, /*
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 */
FC_CARRAY */
0x1b, /*
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 */
FC_VARIABLE_REPEAT */
0x48, /*
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 */
0x0, /*
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 */ 0x1, /* FC_BYTE */
0x5b, /*
FC_END */
/* 684 */
0x15, /*
FC_STRUCT */
0x3, /*
3 */
/* 686 */ NdrFcShort( 0x10 ), /* 16 */
/* 688 */ 0x8, /* FC_LONG */
0x6, /*
FC_SHORT */
/* 690 */ 0x6, /* FC_SHORT */
0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 692 */ 0x0, /* 0 */

```

```

NdrFcShort( 0xffffffff
), /* 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( 0xfffffe8 ), /* Offset= -
24 (684) */
/* 710 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 712 */
0x11, 0x0, /*
FC_RP */
/* 714 */ NdrFcShort( 0xfffff0c ), /* 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( 0xffffffff ), /* Offset= -
24 (716) */
/* 742 */
FC_END */
0x5b, /*
FC_LONG */
/* 744 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */
/* 746 */
FC_CARRAY */
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 */
FC_PSTRUCT */
0x16, /*
3 */
/* 758 */ NdrFcShort( 0x8 ), /* 8 */
/* 760 */
FC_PP */
0x4b, /*
FC_PAD */
/* 762 */
0x5c, /*
FC_NO_REPEAT */
0x46, /*
0x5c, /*
FC_PAD */
/* 764 */ NdrFcShort( 0x4 ), /* 4 */
/* 766 */ NdrFcShort( 0x4 ), /* 4 */
/* 768 */ 0x12, 0x0, /* FC_UP */
/* 770 */ NdrFcShort( 0xffffffff ), /* Offset= -
24 (746) */
/* 772 */
FC_END */
0x5b, /*
FC_LONG */
/* 774 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */
/* 776 */
FC_CARRAY */
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 */
FC_PSTRUCT */
0x16, /*
0x3, /*
3 */
/* 788 */ NdrFcShort( 0x8 ), /* 8 */
/* 790 */
FC_PP */
0x4b, /*
FC_PAD */
0x5c, /*
/* 792 */
FC_NO_REPEAT */
0x46, /*
0x5c, /*
FC_PAD */
/* 794 */ NdrFcShort( 0x4 ), /* 4 */
/* 796 */ NdrFcShort( 0x4 ), /* 4 */
/* 798 */ 0x12, 0x0, /* FC_UP */
/* 800 */ NdrFcShort( 0xffffffff ), /* Offset= -
24 (776) */
/* 802 */
FC_END */
0x5b, /*
FC_LONG */
0x8, /*
/* 804 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */
/* 806 */
FC_CARRAY */
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 */
FC_PSTRUCT */
0x16, /*
0x3, /*
3 */
/* 818 */ NdrFcShort( 0x8 ), /* 8 */
/* 820 */
FC_PP */
0x4b, /*
FC_PAD */
0x5c, /*
/* 822 */

```

```

0x46, /*
FC_NO_REPEAT */
0x5c, /*
FC_PAD */
/* 824 */ NdrFcShort( 0x4 ), /* 4 */
/* 826 */ NdrFcShort( 0x4 ), /* 4 */
/* 828 */ 0x12, 0x0, /* FC_UP */
/* 830 */ NdrFcShort( 0xffffffff ), /* Offset= -
24 (806) */
/* 832 */
FC_END */
0x5b, /*
FC_LONG */
0x8, /*
/* 834 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */
/* 836 */
FC_STRUCT */
0x15, /*
0x3, /*
3 */
/* 838 */ NdrFcShort( 0x8 ), /* 8 */
/* 840 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 842 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 844 */
FC_CARRAY */
0x1b, /*
FC_CARRAY */
0x3, /*
3 */
/* 846 */ NdrFcShort( 0x8 ), /* 8 */
/* 848 */ 0x7, /* Corr desc: FC_USHORT
*/
0x0, /*
*/
/* 850 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 852 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 854 */ NdrFcShort( 0xffffffff ), /* Offset= -
18 (836) */
/* 856 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 858 */
FC_BOGUS_STRUCT */
0x1a, /*
0x3, /*
3 */
/* 860 */ NdrFcShort( 0x28 ), /* 40 */
/* 862 */ NdrFcShort( 0xffffffff ), /* Offset= -
18 (844) */
/* 864 */ NdrFcShort( 0x0 ), /* Offset= 0 (864) */
/* 866 */ 0x6, /* FC_SHORT */
0x6, /*
FC_SHORT */
/* 868 */ 0x38, /* FC_ALIGN4 */

```

```

0x8, /*
FC_LONG */
/* 870 */ 0x8, /* FC_LONG */
0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 872 */ 0x0, /* 0 */
NdrFcShort( 0xfffffdf7
), /* Offset= -521 (352) */
0x5b, /*
FC_END */
/* 876 */
0x12, 0x0, /*
FC_UP */
/* 878 */ NdrFcShort( 0xfffffef6 ), /* Offset= -
266 (612) */
/* 880 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 882 */ 0x1, /* FC_BYTE */
0x5c, /*
FC_PAD */
/* 884 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 886 */ 0x6, /* FC_SHORT */
0x5c, /*
FC_PAD */
/* 888 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 890 */ 0x8, /* FC_LONG */
0x5c, /*
FC_PAD */
/* 892 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 894 */ 0xa, /* FC_FLOAT */
0x5c, /*
FC_PAD */
/* 896 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 898 */ 0xc, /* FC_DOUBLE */
0x5c, /*
FC_PAD */
/* 900 */
0x12, 0x0, /*
FC_UP */
/* 902 */ NdrFcShort( 0xfffffd90 ), /* Offset= -
624 (278) */
/* 904 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 906 */ NdrFcShort( 0xfffffd92 ), /* Offset= -
622 (284) */
/* 908 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 910 */ NdrFcShort( 0xfffffda6 ), /* Offset= -
602 (308) */
/* 912 */
0x12, 0x10, /*
FC_UP [pointer_deref] */

```

```

/* 914 */ NdrFcShort( 0xfffffdb4 ), /* Offset= -
588 (326) */
/* 916 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 918 */ NdrFcShort( 0xfffffdc2 ), /* Offset= -
574 (344) */
/* 920 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 922 */ NdrFcShort( 0x2 ), /* Offset= 2 (924) */
/* 924 */
0x12, 0x0, /*
FC_UP */
/* 926 */ NdrFcShort( 0x16 ), /* Offset= 22 (948) */
/* 928 */
0x15, /*
FC_STRUCT */
0x7, /*
7 */
/* 930 */ NdrFcShort( 0x10 ), /* 16 */
/* 932 */ 0x6, /* FC_SHORT */
0x1, /*
FC_BYTE */
/* 934 */ 0x1, /* FC_BYTE */
0x38, /*
FC_ALIGNM4 */
/* 936 */ 0x8, /* FC_LONG */
0x39, /*
FC_ALIGNM8 */
/* 938 */ 0xb, /* FC_HYPER */
0x5b, /*
FC_END */
/* 940 */
0x12, 0x0, /*
FC_UP */
/* 942 */ NdrFcShort( 0xfffffff2 ), /* Offset= -
14 (928) */
/* 944 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 946 */ 0x2, /* FC_CHAR */
0x5c, /*
FC_PAD */
/* 948 */
0x1a, /*
FC_BOGUS_STRUCT */
0x7, /*
7 */
/* 950 */ NdrFcShort( 0x20 ), /* 32 */
/* 952 */ NdrFcShort( 0x0 ), /* 0 */
/* 954 */ NdrFcShort( 0x0 ), /* Offset= 0 (954) */
/* 956 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 958 */ 0x6, /* FC_SHORT */
0x6, /*
FC_SHORT */
/* 960 */ 0x6, /* FC_SHORT */
0x6, /*
FC_SHORT */
/* 962 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/

```

```

0x0, /*
0 */
/* 964 */ NdrFcShort( 0xfffffc42 ), /* Offset= -
958 (6) */
/* 966 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 968 */ 0xb4, /* FC_USER_MARSHAL */
0x83, /*
131 */
/* 970 */ NdrFcShort( 0x0 ), /* 0 */
/* 972 */ NdrFcShort( 0x10 ), /* 16 */
/* 974 */ NdrFcShort( 0x0 ), /* 0 */
/* 976 */ NdrFcShort( 0xfffffc32 ), /* Offset= -
974 (2) */
/* 978 */
0x11, 0x4, /*
FC_RP [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 *
_tpc_com_ps_ProxyVtblList[] =
{
( CInterfaceProxyVtbl *) &ITPCCProxyVtbl,
0
};
const CInterfaceStubVtbl *
_tpc_com_ps_StubVtblList[] =
{
( CInterfaceStubVtbl *) &ITPCCStubVtbl,
0
};
PCInterfaceName const
_tpc_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)*/

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

/* this ALWAYS GENERATED file contains the proxy stub
code */

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:08 2001
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=i2), 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_HEADERING( )

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

```

```

/* verify that the <rpcproxy.h> version is high
enough to compile this file*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 475
#endif

#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of
<rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 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,{0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0
x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFE6E6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0
x4F,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 */
        FC_AUTO_HANDLE * 0x33, /* */
        Old Flags: object, Oi2 /* */
        /* 2 */ NdrFcLong( 0x0 ), /* 0 */
        /* 6 */ NdrFcShort( 0x3 ), /* 3 */
        #ifndef _ALPHA_
        /* 8 */ NdrFcShort( 0x38 ), /* ia64 Stack
size/offset = 56 */
        #else

```

```

NdrFcShort( 0x30 ), /*
axp64 Stack size/offset = 48 */
#endif
/* 10 */ NdrFcShort( 0x0 ), /* 0 */
/* 12 */ NdrFcShort( 0x8 ), /* 8 */
/* 14 */ 0x47, /* Oi2 Flags: srv must
size, clt must size, has return, has ext, */
3 /*
/* 16 */ 0xa, /* 10 */
/* 18 */ 0x7, /*
Ext Flags: new corr desc, clt corr check, srv corr
check, */
/* 18 */ NdrFcShort( 0x20 ), /* 32 */
/* 20 */ NdrFcShort( 0x20 ), /* 32 */
/* 22 */ NdrFcShort( 0x0 ), /* 0 */
/* 24 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 26 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#ifndef _ALPHA_
/* 28 */ NdrFcShort( 0x10 ), /* ia64 Stack
size/offset = 16 */
#else
NdrFcShort( 0x8 ), /*
axp64 Stack size/offset = 8 */
#endif
/* 30 */ NdrFcShort( 0x3b6 ), /* Type
Offset=950 */

/* Parameter txn_out */

/* 32 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
#ifndef _ALPHA_
/* 34 */ NdrFcShort( 0x28 ), /* ia64 Stack
size/offset = 40 */
#else
NdrFcShort( 0x20 ), /*
axp64 Stack size/offset = 32 */
#endif
/* 36 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Return value */

/* 38 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_
/* 40 */ NdrFcShort( 0x30 ), /* ia64 Stack
size/offset = 48 */
#else
NdrFcShort( 0x28 ), /*
axp64 Stack size/offset = 40 */
#endif
/* 42 */ 0x8, /* FC_LONG */
0 /*
/* Procedure Payment */

```

```

/* 44 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 46 */ NdrFcLong( 0x0 ), /* 0 */
/* 50 */ NdrFcShort( 0x4 ), /* 4 */
#ifndef _ALPHA_
/* 52 */ NdrFcShort( 0x38 ), /* ia64 Stack
size/offset = 56 */
#else
NdrFcShort( 0x30 ), /*
axp64 Stack size/offset = 48 */
#endif
/* 54 */ NdrFcShort( 0x0 ), /* 0 */
/* 56 */ NdrFcShort( 0x8 ), /* 8 */
/* 58 */ 0x47, /* Oi2 Flags: srv must
size, clt must size, has return, has ext, */
3 /*
/* 60 */ 0xa, /* 10 */
/* 62 */ 0x7, /*
Ext Flags: new corr desc, clt corr check, srv corr
check, */
/* 62 */ NdrFcShort( 0x20 ), /* 32 */
/* 64 */ NdrFcShort( 0x20 ), /* 32 */
/* 66 */ NdrFcShort( 0x0 ), /* 0 */
/* 68 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 70 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#ifndef _ALPHA_
/* 72 */ NdrFcShort( 0x10 ), /* ia64 Stack
size/offset = 16 */
#else
NdrFcShort( 0x8 ), /*
axp64 Stack size/offset = 8 */
#endif
/* 74 */ NdrFcShort( 0x3b6 ), /* Type
Offset=950 */

/* Parameter txn_out */

/* 76 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
#ifndef _ALPHA_
/* 78 */ NdrFcShort( 0x28 ), /* ia64 Stack
size/offset = 40 */
#else
NdrFcShort( 0x20 ), /*
axp64 Stack size/offset = 32 */
#endif
/* 80 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Return value */

/* 82 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_

```

```

/* 84 */ NdrFcShort( 0x30 ), /* ia64 Stack
size/offset = 48 */
#else
NdrFcShort( 0x28 ), /*
axp64 Stack size/offset = 40 */
#endif
/* 86 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure Delivery */

/* 88 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 90 */ NdrFcLong( 0x0 ), /* 0 */
/* 94 */ NdrFcShort( 0x5 ), /* 5 */
#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 */
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 */
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 */
FC_STRUCT /*
0x15, /*
7 */
0x7, /*
/* 282 */ NdrFcShort( 0x8 ), /* 8 */
/* 284 */ 0xb, /* FC_HYPER */
0x5b, /*
FC_END */
/* 286 */
0x12, 0x0, /*
FC_UP */
/* 288 */ NdrFcShort( 0xe ), /* Offset= 14 (302) */
/* 290 */
0x1b, /*
FC_CARRAY */
0x1, /*
1 */
/* 292 */ NdrFcShort( 0x2 ), /* 2 */
/* 294 */ 0x9, /* Corr desc: FC_ULONG
*/

```

```

0x0, /*
*/
/* 296 */ NdrFcShort( 0xfffc ), /* -4 */
/* 298 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 300 */ 0x6, /* FC_SHORT */
0x5b, /*
FC_END */
/* 302 */
0x17, /*
FC_CSTRUCT */
0x3, /*
3 */
/* 304 */ NdrFcShort( 0x8 ), /* 8 */
/* 306 */ NdrFcShort( 0xfffff0 ), /* Offset= -
16 (290) */
/* 308 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 310 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 312 */
0x2E, /*
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 */
0x2E, /*
0x5a, /*
FC_CONSTANT_IID */
/* 332 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 336 */ NdrFcShort( 0x0 ), /* 0 */
/* 338 */ NdrFcShort( 0x0 ), /* 0 */
/* 340 */ 0xc0, /* 192 */
0x0, /*
0 */
/* 342 */ 0x0, /* 0 */
0x0, /*
0 */
/* 344 */ 0x0, /* 0 */
0x0, /*
0 */
/* 346 */ 0x0, /* 0 */
0x46, /*
70 */
/* 348 */

```

```

0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 350 */ NdrFcShort( 0x2 ), /* Offset= 2 (352) */
/* 352 */
0x12, 0x0, /*
FC_UP */
/* 354 */ NdrFcShort( 0x1e6 ), /* Offset=
486 (840) */
/* 356 */
0x2a, /*
FC_ENCAPSULATED_UNION */
0x89, /*
137 */
/* 358 */ NdrFcShort( 0x20 ), /* 32 */
/* 360 */ NdrFcShort( 0xa ), /* 10 */
/* 362 */ NdrFcLong( 0x8 ), /* 8 */
/* 366 */ NdrFcShort( 0x50 ), /* Offset= 80 (446) */
/* 368 */ NdrFcLong( 0xd ), /* 13 */
/* 372 */ NdrFcShort( 0x70 ), /* Offset= 112 (484) */
/* 374 */ NdrFcLong( 0x9 ), /* 9 */
/* 378 */ NdrFcShort( 0x90 ), /* Offset= 144 (522) */
/* 380 */ NdrFcLong( 0xc ), /* 12 */
/* 384 */ NdrFcShort( 0xb0 ), /* Offset= 176 (560) */
/* 386 */ NdrFcLong( 0x24 ), /* 36 */
/* 390 */ NdrFcShort( 0x104 ), /* Offset=
260 (650) */
/* 392 */ NdrFcLong( 0x800d ), /* 32781 */
/* 396 */ NdrFcShort( 0x120 ), /* Offset=
288 (684) */
/* 398 */ NdrFcLong( 0x10 ), /* 16 */
/* 402 */ NdrFcShort( 0x13a ), /* Offset=
314 (716) */
/* 404 */ NdrFcLong( 0x2 ), /* 2 */
/* 408 */ NdrFcShort( 0x150 ), /* Offset=
336 (744) */
/* 410 */ NdrFcLong( 0x3 ), /* 3 */
/* 414 */ NdrFcShort( 0x166 ), /* Offset=
358 (772) */
/* 416 */ NdrFcLong( 0x14 ), /* 20 */
/* 420 */ NdrFcShort( 0x17c ), /* Offset=
380 (800) */
/* 422 */ NdrFcShort( 0xffffffff ), /* Offset= -1
(421) */
/* 424 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /*
3 */
/* 426 */ NdrFcShort( 0x0 ), /* 0 */
/* 428 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 430 */ NdrFcShort( 0x0 ), /* 0 */
/* 432 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 434 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 438 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 440 */
0x12, 0x0, /*
FC_UP */
/* 442 */ NdrFcShort( 0xffffffff74 ), /* Offset= -
140 (302) */

```

```

/* 444 */ 0x5c,          /* FC_PAD */
FC_END /*
/* 446 */
FC_BOGUS_STRUCT */
0x1a,          /*
0x3,          /*
3 */
/* 448 */ NdrFcShort( 0x10 ), /* 16 */
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */ NdrFcShort( 0x6 ), /* Offset= 6 (458) */
/* 454 */ 0x8,          /* FC_LONG */
0x39,          /*
FC_ALIGNM8 */
/* 456 */ 0x36,        /* FC_POINTER */
0x5b,          /*
FC_END /*
/* 458 */
0x11, 0x0,     /*
FC_RP */
/* 460 */ NdrFcShort( 0xfffff5dc ), /* 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( 0xfffff58 ), /* Offset= -
168 (312) */
/* 482 */ 0x5c,        /* FC_PAD */
0x5b,          /*
FC_END /*
/* 484 */
0x1a,          /*
FC_BOGUS_STRUCT */
0x3,          /*
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( 0xfffff5dc ), /* 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( 0xfffff44 ), /* Offset= -
188 (330) */
/* 520 */ 0x5c,        /* FC_PAD */
0x5b,          /*
FC_END /*
/* 522 */
0x1a,          /*
FC_BOGUS_STRUCT */
0x3,          /*
0x3,          /*
3 */
/* 524 */ NdrFcShort( 0x10 ), /* 16 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 530 */ 0x8,          /* FC_LONG */
0x39,          /*
FC_ALIGNM8 */
/* 532 */ 0x36,        /* FC_POINTER */
0x5b,          /*
FC_END /*
/* 534 */
0x11, 0x0,     /*
FC_RP */
/* 536 */ NdrFcShort( 0xfffff5dc ), /* Offset= -
36 (500) */
/* 538 */
0x21,          /*
FC_BOGUS_ARRAY */
0x3,          /*
3 */
/* 540 */ NdrFcShort( 0x0 ), /* 0 */
/* 542 */ 0x19,        /* Corr desc: field
pointer, FC_ULONG */
0x0,          /*
*/
/* 544 */ NdrFcShort( 0x0 ), /* 0 */
/* 546 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 548 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 552 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 554 */
0x12, 0x0,     /*
FC_UP */

```

```

/* 556 */ NdrFcShort( 0x176 ), /* Offset=
374 (930) */
/* 558 */ 0x5c,        /* FC_PAD */
0x5b,          /*
FC_END /*
/* 560 */
0x1a,          /*
FC_BOGUS_STRUCT */
0x3,          /*
0x3,          /*
3 */
/* 562 */ NdrFcShort( 0x10 ), /* 16 */
/* 564 */ NdrFcShort( 0x0 ), /* 0 */
/* 566 */ NdrFcShort( 0x6 ), /* Offset= 6 (572) */
/* 568 */ 0x8,          /* FC_LONG */
0x39,          /*
FC_ALIGNM8 */
/* 570 */ 0x36,        /* FC_POINTER */
0x5b,          /*
FC_END /*
/* 572 */
0x11, 0x0,     /*
FC_RP */
/* 574 */ NdrFcShort( 0xfffff5dc ), /* Offset= -
36 (538) */
/* 576 */
0x2f,          /*
FC_IP */
0x5a,          /*
FC_CONSTANT_IID */
/* 578 */ NdrFcLong( 0x2f ), /* 47 */
/* 582 */ NdrFcShort( 0x0 ), /* 0 */
/* 584 */ NdrFcShort( 0x0 ), /* 0 */
/* 586 */ 0xc0,        /* 192 */
0x0,          /*
0 */
/* 588 */ 0x0,        /* 0 */
0x0,          /*
0 */
/* 590 */ 0x0,        /* 0 */
0x0,          /*
0 */
/* 592 */ 0x0,        /* 0 */
0x46,         /*
70 */
/* 594 */
0x1b,          /*
FC_CARRAY */
0x0,          /*
0 */
/* 596 */ NdrFcShort( 0x1 ), /* 1 */
/* 598 */ 0x19,        /* Corr desc: field
pointer, FC_ULONG */
0x0,          /*
*/
/* 600 */ NdrFcShort( 0x4 ), /* 4 */
/* 602 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 604 */ 0x1,        /* FC_BYTE */
0x5b,          /*
FC_END /*
/* 606 */
0x1a,          /*
FC_BOGUS_STRUCT */

```

```

0x3, /*
3 */
/* 608 */ NdrFcShort( 0x18 ), /* 24 */
/* 610 */ NdrFcShort( 0x0 ), /* 0 */
/* 612 */ NdrFcShort( 0xc ), /* Offset= 12 (624) */
/* 614 */ 0x8, /* FC_LONG */
FC_LONG /*
/* 616 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /*
0 */
/* 618 */ NdrFcShort( 0xfffffd6 ), /* Offset= -
42 (576) */
/* 620 */ 0x39, /* FC_ALIGNM8 */
FC_POINTER /*
/* 622 */ 0x5c, /* FC_PAD */
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 */
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 */
FC_ALIGNM8 /*
/* 660 */ 0x36, /* FC_POINTER */
FC_END /*
/* 662 */

```

```

0x11, 0x0, /*
FC_RP /*
/* 664 */ NdrFcShort( 0xfffffdc ), /* Offset= -
36 (628) */
/* 666 */
0x1d, /*
FC_SMPARRAY /*
0x0, /*
0 */
/* 668 */ NdrFcShort( 0x8 ), /* 8 */
/* 670 */ 0x1, /* FC_BYTE */
FC_END /*
/* 672 */
0x15, /*
FC_STRUCT /*
0x3, /*
3 */
/* 674 */ NdrFcShort( 0x10 ), /* 16 */
/* 676 */ 0x8, /* FC_LONG */
FC_SHORT /*
/* 678 */ 0x6, /* FC_SHORT */
FC_EMBEDDED_COMPLEX /*
/* 680 */ 0x0, /* 0 */
NdrFcShort( 0xffffff1
), /* Offset= -15 (666) */
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 */
FC_ALIGNM8 /*
/* 694 */ 0x36, /* FC_POINTER */
FC_EMBEDDED_COMPLEX /*
/* 696 */ 0x0, /* 0 */
NdrFcShort( 0xfffffe7
), /* Offset= -25 (672) */
FC_END /*
/* 700 */
0x11, 0x0, /*
FC_RP /*
/* 702 */ NdrFcShort( 0xfffff10 ), /* 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 */
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 */
FC_ALIGNM8 /*
/* 726 */ 0x36, /* FC_POINTER */
FC_END /*
/* 728 */
0x12, 0x0, /*
FC_UP /*
/* 730 */ NdrFcShort( 0xfffffe6 ), /* 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 */
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 */
FC_ALIGNM8 /*
/* 754 */ 0x36, /* FC_POINTER */
FC_END /*
/* 756 */
0x12, 0x0, /*
FC_UP /*
/* 758 */ NdrFcShort( 0xfffffe6 ), /* 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( 0xffffffffe6 ), /* 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( 0xffffffffe6 ), /* 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( 0xffffffffec ), /* Offset= -
20 (816) */
/* 838 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 840 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 842 */ NdrFcShort( 0x38 ), /* 56 */
/* 844 */ NdrFcShort( 0xffffffffec ), /* 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( 0xffffffff02 ), /* 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( 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_ps_p.c

```

#pragma warning( disable: 4049 ) /* more than 64k
source lines */
/* this ALWAYS GENERATED file contains the proxy stub
code */
/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:08 2001
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=12), 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)
#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,{0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0
x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFE6E6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0x00,0
x4F,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,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.3.280 */
    0,
    UserMarshalRoutines,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* Reserved3 */
    0, /* Reserved4 */
    0 /* Reserved5 */
};

#pragma data_seg(".rdata")

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

#if !defined(__RPC_WIN32__)
#error Invalid build platform for this stub.
#endif

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this
stub because it uses these features:
#error -Oif or -Oicf, [wire_marshal] or
[user_marshal] attribute.
#error However, your C/C++ compilation flags indicate
you intend to run this app on earlier systems.
#error This app will die there with the
RPC_X_WRONG_STUB_VERSION error.
#endif

static const MIDL_PROC_FORMAT_STRING
__MIDL_ProcFormatString =
{
    0,
    {
        /* Procedure NewOrder */

```

```

0x33, /*
FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 2 */ NdrFcLong( 0x0 ), /* 0 */
/* 6 */ NdrFcShort( 0x3 ), /* 3 */
#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( 0x20 ), /*
Alpha 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_
#ifdef _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 */
#ifdef _ALPHA_
#ifdef _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, */

```



```

#ifdef _ALPHA_
#ifdef _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 */
#ifdef _ALPHA_
#ifdef _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, */
#ifdef _ALPHA_
#ifdef _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 */
#ifdef _ALPHA_
#ifdef _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, */
#ifdef _ALPHA_
#ifdef _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 */
#ifdef _ALPHA_
#ifdef _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, */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 120 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
#else
NdrFcShort( 0x8 ), /*
MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 122 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Parameter txn_out */

/* 124 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
#ifdef _ALPHA_
#ifdef _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
#endif

```

```

#else
NdrFcShort( 0x28 ), /*
Alpha Stack size/offset = 40 */
#endif
/* 146 */ NdrFcShort( 0x0 ), /* 0 */
/* 148 */ NdrFcShort( 0x8 ), /* 8 */
/* 150 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

/* Parameter txn_in */

/* 152 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 154 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
#else
NdrFcShort( 0x8 ), /*
MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 156 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Parameter txn_out */

/* 158 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 160 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
#else
NdrFcShort( 0x18 ), /*
MIPS Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
PPC Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
Alpha Stack size/offset = 24 */
#endif
/* 162 */ NdrFcShort( 0x3da ), /* Type
Offset=986 */

/* Return value */

```

```

/* 164 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 166 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
#else
NdrFcShort( 0x1c ), /*
MIPS Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /*
PPC Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x20 ), /*
Alpha Stack size/offset = 32 */
#endif
/* 168 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure CallSetComplete */

/* 170 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 172 */ NdrFcLong( 0x0 ), /* 0 */
/* 176 */ NdrFcShort( 0x8 ), /* 8 */
#ifndef _ALPHA_
/* 178 */ NdrFcShort( 0x8 ), /* x86, MIPS, PPC Stack
size/offset = 8 */
#else
NdrFcShort( 0x10 ), /*
Alpha Stack size/offset = 16 */
#endif
/* 180 */ NdrFcShort( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x8 ), /* 8 */
/* 184 */ 0x4, /* Oi2 Flags: has
return, */
0x1, /*
1 */

/* Return value */

/* 186 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_
/* 188 */ NdrFcShort( 0x4 ), /* x86, MIPS, PPC Stack
size/offset = 4 */
#else
NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 190 */ 0x8, /* FC_LONG */
0x0, /*
0 */

0x0

};

```

```

static const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString =
{
    0,
    {
        0 /*/
        /* 2 */
        FC_UP /*/
        /* 4 */ NdrFcShort( 0x3b0 ), /* Offset=
        944 (948) */
        /* 6 */
        FC_NON_ENCAPSULATED_UNION /*/
        /* 8 */ 0x7, /* Corr desc: FC_USHORT
        */
        /* 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 */
FC_STRUCT /*/
/* 280 */ NdrFcShort( 0x8 ), /* 8 */
/* 282 */ 0xb, /* FC_HYPER */
/* 284 */
FC_END /*/
/* 286 */ NdrFcShort( 0xc ), /* Offset= 12 (298) */
/* 288 */
FC_UP /*/
/* 290 */ NdrFcShort( 0x2 ), /* 2 */
/* 292 */ 0x9, /* Corr desc: FC_ULONG
*/
/* 294 */ NdrFcShort( 0xffffc ), /* -4 */
/* 296 */ 0x6, /* FC_SHORT */
/* 298 */
FC_END /*/
/* 299 */
FC_CSTRUCT /*/
/* 300 */ NdrFcShort( 0x8 ), /* 8 */
/* 302 */ NdrFcShort( 0xffffffff2 ), /* Offset= -
14 (288) */
/* 304 */ 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 */
0 /*
/* 320 */ 0x0, /* 0 */
0 /*
/* 322 */ 0x0, /* 0 */
0 /*
/* 324 */ 0x0, /* 0 */
70 /*
/* 326 */
FC_IP /*
0x2f, /*
0x5a, /*
FC_CONSTANT_IID /*
/* 328 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 332 */ NdrFcShort( 0x0 ), /* 0 */
/* 334 */ NdrFcShort( 0x0 ), /* 0 */
/* 336 */ 0xc0, /* 192 */
0 /*
/* 338 */ 0x0, /* 0 */
0 /*
/* 340 */ 0x0, /* 0 */
0 /*
/* 342 */ 0x0, /* 0 */
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 */
FC_CARRAY /*
0x1b, /*
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 /*
0x8, /*
FC_LONG /*
/* 448 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END /*
/* 450 */

```

```

0x16, /*
FC_PSTRUCT /*
0x3, /*
3 /*
/* 452 */ NdrFcShort( 0x8 ), /* 8 */
/* 454 */
0x4b, /*
FC_PP /*
0x5c, /*
FC_PAD /*
/* 456 */
0x46, /*
FC_NO_REPEAT /*
0x5c, /*
FC_PAD /*
/* 458 */ NdrFcShort( 0x4 ), /* 4 */
/* 460 */ NdrFcShort( 0x4 ), /* 4 */
/* 462 */ 0x11, 0x0, /* FC_RP */
/* 464 */ NdrFcShort( 0xffffffffd4 ), /* Offset= -
44 (420) */
/* 466 */
0x5b, /*
FC_END /*
0x8, /*
FC_LONG /*
/* 468 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END /*
/* 470 */
0x21, /*
FC_BOGUS_ARRAY /*
0x3, /*
3 /*
/* 472 */ NdrFcShort( 0x0 ), /* 0 */
/* 474 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
/*
/* 476 */ NdrFcShort( 0x0 ), /* 0 */
/* 478 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 482 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 /*
/* 484 */ NdrFcShort( 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( 0xffffffe0 ), /* Offset= -
32 (470) */
/* 504 */
                                0x21,          /*
FC_BOGUS_ARRAY */
3 */
/* 506 */ NdrFcShort( 0x0 ), /* 0 */
/* 508 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
                                0x0,          /*
*/
/* 510 */ NdrFcShort( 0x0 ), /* 0 */
/* 512 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 516 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
                                0x0,          /*
0 */
/* 518 */ NdrFcShort( 0xffffff40 ), /* Offset= -
192 (326) */
/* 520 */ 0x5c, /* FC_PAD */
                                0x5b,          /*
FC_END */
/* 522 */
                                0x1a,          /*
FC_BOGUS_STRUCT */
3 */
/* 524 */ NdrFcShort( 0x8 ), /* 8 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 530 */ 0x8, /* FC_LONG */
                                0x36,          /*
FC_POINTER */
/* 532 */ 0x5c, /* FC_PAD */
                                0x5b,          /*
FC_END */
/* 534 */
                                0x11, 0x0,      /*
FC_RP */
/* 536 */ NdrFcShort( 0xffffffe0 ), /* Offset= -
32 (504) */
/* 538 */
                                0x1b,          /*
FC_CARRAY */
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 */
FC_VARIABLE_REPEAT */
                                0x48,          /*
                                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 */
3 */
/* 570 */ NdrFcShort( 0x8 ), /* 8 */
/* 572 */ NdrFcShort( 0x0 ), /* 0 */
/* 574 */ NdrFcShort( 0x6 ), /* Offset= 6 (580) */
/* 576 */ 0x8, /* FC_LONG */
                                0x36,          /*
FC_POINTER */
/* 578 */ 0x5c, /* FC_PAD */
                                0x5b,          /*
FC_END */
/* 580 */
                                0x11, 0x0,      /*
FC_RP */
/* 582 */ NdrFcShort( 0xffffffd4 ), /* 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 */
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( 0xffffffd8 ), /* Offset= -
40 (584) */
/* 626 */ 0x36, /* FC_POINTER */
                                0x5b,          /*
FC_END */
/* 628 */
                                0x12, 0x0,      /*
FC_UP */
/* 630 */ NdrFcShort( 0xffffffe4 ), /* Offset= -
28 (602) */
/* 632 */
                                0x1b,          /*
FC_CARRAY */
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( 0xffffffffd4 ), /* 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( 0xffffffffd4 ), /* Offset= -
44 (632) */
/* 678 */
0x1d, /*
FC_SMPARRAY */
0x0, /*
0 */
/* 680 */ NdrFcShort( 0x8 ), /* 8 */
/* 682 */ 0x1, /* FC_BYTE */
0x5b, /*
FC_END */
/* 684 */
0x15, /*
FC_STRUCT */
0x3, /*
3 */
/* 686 */ NdrFcShort( 0x10 ), /* 16 */
/* 688 */ 0x8, /* FC_LONG */
0x6, /*
FC_SHORT */
/* 690 */ 0x6, /* FC_SHORT */
0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 692 */ 0x0, /* 0 */
NdrFcShort( 0xfffffffff1
), /* 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( 0xffffffffe8 ), /* 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( 0xffffffffe8 ), /* Offset= -
24 (716) */
/* 742 */
0x5b, /*
FC_END */
0x8, /*
FC_LONG */
/* 744 */ 0x8, /* FC_LONG */

```

```

0x5b, /*
FC_END */
/* 746 */
0x1b, /*
FC_CARRAY */
0x1, /*
1 */
/* 748 */ NdrFcShort( 0x2 ), /* 2 */
/* 750 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 752 */ NdrFcShort( 0x0 ), /* 0 */
/* 754 */ 0x6, /* FC_SHORT */
0x5b, /*
FC_END */
/* 756 */
0x16, /*
FC_PSTRUCT */
0x3, /*
3 */
/* 758 */ NdrFcShort( 0x8 ), /* 8 */
/* 760 */
0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */
/* 762 */
0x46, /*
FC_NO_REPEAT */
0x5c, /*
FC_PAD */
/* 764 */ NdrFcShort( 0x4 ), /* 4 */
/* 766 */ NdrFcShort( 0x4 ), /* 4 */
/* 768 */ 0x12, 0x0, /* FC_UP */
/* 770 */ NdrFcShort( 0xffffffffe8 ), /* 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 */
FC_PP /*
0x4b, /*
FC_PAD /*
/* 792 */
0x5c, /*
FC_NO_REPEAT /*
0x46, /*
0x5c, /*
FC_PAD /*
/* 794 */ NdrFcShort( 0x4 ), /* 4 */
/* 796 */ NdrFcShort( 0x4 ), /* 4 */
/* 798 */ 0x12, 0x0, /* FC_UP */
/* 800 */ NdrFcShort( 0xfffffe8 ), /* Offset= -
24 (776) */
/* 802 */
0x5b, /*
FC_END */
0x8, /*
/* 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 */
FC_PP /*
0x4b, /*
FC_PAD /*
/* 822 */
0x5c, /*
FC_NO_REPEAT /*
0x46, /*
0x5c, /*
FC_PAD /*
/* 824 */ NdrFcShort( 0x4 ), /* 4 */
/* 826 */ NdrFcShort( 0x4 ), /* 4 */
/* 828 */ 0x12, 0x0, /* FC_UP */
/* 830 */ NdrFcShort( 0xfffffe8 ), /* Offset= -
24 (806) */
/* 832 */

```

```

0x5b, /*
FC_END /*
0x8, /*
/* 834 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END /*
/* 836 */
0x15, /*
FC_STRUCT /*
0x3, /*
3 */
/* 838 */ NdrFcShort( 0x8 ), /* 8 */
/* 840 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG /*
/* 842 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END /*
/* 844 */
0x1b, /*
FC_CARRAY /*
0x3, /*
3 */
/* 846 */ NdrFcShort( 0x8 ), /* 8 */
/* 848 */ 0x7, /* Corr desc: FC_USHORT
*/
0x0, /*
*/
/* 850 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 852 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 854 */ NdrFcShort( 0xfffffee ), /* Offset= -
18 (836) */
/* 856 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END /*
/* 858 */
0x1a, /*
FC_BOGUS_STRUCT /*
0x3, /*
3 */
/* 860 */ NdrFcShort( 0x28 ), /* 40 */
/* 862 */ NdrFcShort( 0xfffffee ), /* Offset= -
18 (844) */
/* 864 */ NdrFcShort( 0x0 ), /* Offset= 0 (864) */
/* 866 */ 0x6, /* FC_SHORT */
0x6, /*
FC_SHORT /*
/* 868 */ 0x38, /* FC_ALIGNM4 */
0x8, /*
FC_LONG /*
/* 870 */ 0x8, /* FC_LONG */
0x4c, /*
FC_EMBEDDED_COMPLEX /*
/* 872 */ 0x0, /* 0 */
NdrFcShort( 0xffffdf7
), /* Offset= -521 (352) */
0x5b, /*
FC_END /*

```

```

/* 876 */
0x12, 0x0, /*
FC_UP /*
/* 878 */ NdrFcShort( 0xfffffef6 ), /* Offset= -
266 (612) */
/* 880 */
0x12, 0x8, /*
FC_UP [simple_pointer] /*
/* 882 */ 0x1, /* FC_BYTE */
0x5c, /*
FC_PAD /*
/* 884 */
0x12, 0x8, /*
FC_UP [simple_pointer] /*
/* 886 */ 0x6, /* FC_SHORT */
0x5c, /*
FC_PAD /*
/* 888 */
0x12, 0x8, /*
FC_UP [simple_pointer] /*
/* 890 */ 0x8, /* FC_LONG */
0x5c, /*
FC_PAD /*
/* 892 */
0x12, 0x8, /*
FC_UP [simple_pointer] /*
/* 894 */ 0xa, /* FC_FLOAT */
0x5c, /*
FC_PAD /*
/* 896 */
0x12, 0x8, /*
FC_UP [simple_pointer] /*
/* 898 */ 0xc, /* FC_DOUBLE */
0x5c, /*
FC_PAD /*
/* 900 */
0x12, 0x0, /*
FC_UP /*
/* 902 */ NdrFcShort( 0xfffffd90 ), /* Offset= -
624 (278) */
/* 904 */
0x12, 0x10, /*
FC_UP [pointer_deref] /*
/* 906 */ NdrFcShort( 0xfffffd92 ), /* Offset= -
622 (284) */
/* 908 */
0x12, 0x10, /*
FC_UP [pointer_deref] /*
/* 910 */ NdrFcShort( 0xfffffda6 ), /* Offset= -
602 (308) */
/* 912 */
0x12, 0x10, /*
FC_UP [pointer_deref] /*
/* 914 */ NdrFcShort( 0xfffffdb4 ), /* Offset= -
588 (326) */
/* 916 */
0x12, 0x10, /*
FC_UP [pointer_deref] /*
/* 918 */ NdrFcShort( 0xfffffdc2 ), /* Offset= -
574 (344) */
/* 920 */
0x12, 0x10, /*
FC_UP [pointer_deref] /*

```

```

/* 922 */ NdrFcShort( 0x2 ), /* Offset= 2 (924) */
/* 924 */
                                0x12, 0x0, /*
FC_UP */
/* 926 */ NdrFcShort( 0x16 ), /* Offset= 22 (948) */
/* 928 */
                                0x15, /*
FC_STRUCT */
                                0x7, /*
7 */
/* 930 */ NdrFcShort( 0x10 ), /* 16 */
/* 932 */ 0x6, /* FC_SHORT */
                                0x1, /*
FC_BYTE */
/* 934 */ 0x1, /* FC_BYTE */
                                0x38, /*
FC_ALIGNM4 */
/* 936 */ 0x8, /* FC_LONG */
                                0x39, /*
FC_ALIGNM8 */
/* 938 */ 0xb, /* FC_HYPER */
                                0x5b, /*
FC_END */
/* 940 */
                                0x12, 0x0, /*
FC_UP */
/* 942 */ NdrFcShort( 0xffffffff2 ), /* Offset= -
14 (928) */
/* 944 */
                                0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 946 */ 0x2, /* FC_CHAR */
                                0x5c, /*
FC_PAD */
/* 948 */
                                0x1a, /*
FC_BOGUS_STRUCT */
                                0x7, /*
7 */
/* 950 */ NdrFcShort( 0x20 ), /* 32 */
/* 952 */ NdrFcShort( 0x0 ), /* 0 */
/* 954 */ NdrFcShort( 0x0 ), /* Offset= 0 (954) */
/* 956 */ 0x8, /* FC_LONG */
                                0x8, /*
FC_LONG */
/* 958 */ 0x6, /* FC_SHORT */
                                0x6, /*
FC_SHORT */
/* 960 */ 0x6, /* FC_SHORT */
                                0x6, /*
FC_SHORT */
/* 962 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
                                0x0, /*
0 */
/* 964 */ NdrFcShort( 0xfffffc42 ), /* Offset= -
958 (6) */
/* 966 */ 0x5c, /* FC_PAD */
                                0x5b, /*
FC_END */
/* 968 */ 0xb4, /* FC_USER_MARSHAL */
                                0x83, /*
131 */

```

```

/* 970 */ NdrFcShort( 0x0 ), /* 0 */
/* 972 */ NdrFcShort( 0x10 ), /* 16 */
/* 974 */ NdrFcShort( 0x0 ), /* 0 */
/* 976 */ NdrFcShort( 0xfffffc32 ), /* Offset= -
974 (2) */
/* 978 */
                                0x11, 0x4, /*
FC_RP [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 * ) &ITPCCStubVtbl,
    0
};

PCInterfaceName const
_tpcc_com_ps_InterfaceNamesList[] =
{
    "ITPCC",
    0
};

#define _tpcc_com_ps_CHECK_IID(n)
IID_GENERIC_CHECK_IID( _tpcc_com_ps, pIID,
n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID *
pIID, int * pIndex )
{
    if(!_tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }
}

```

```

return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo
=
{
    (PCInterfaceProxyVtblList *) &
    _tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) &
    _tpcc_com_ps_StubVtblList,
    (const PCInterfaceName *) &
    _tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    &_tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0 /* Filler3 */
};

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

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

/* this ALWAYS GENERATED file contains the proxy stub
code */

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:08 2001
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=12), 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 __REQD_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,{0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0
x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEED6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0
x4F,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,
    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
};

```

```

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 */
        FC_AUTO_HANDLE /* 0x33, */
        /* 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 */

```

```

    }
    };
    }
    static const MIDL_TYPE_FORMAT_STRING
    __MIDL_TypeFormatString =
    {
        0,
        {
            0 * /
            /* 2 */
            FC_UP /*
            /* 4 */ NdrFcShort( 0x39e ), /* Offset=
            926 (930) */
            /* 6 */
            FC_NON_ENCAPSULATED_UNION /*
            /* 8 */ 0x7, /* Corr desc: FC_USHORT
            */
            /* 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 */
FC_STRUCT /*
/* 282 */ NdrFcShort( 0x8 ), /* 8 */
/* 284 */ 0xb, /* FC_HYPER */
/* 286 */
FC_END /*
/* 288 */ NdrFcShort( 0xe ), /* Offset= 14 (302) */
/* 290 */
FC_CARRAY /*
/* 292 */ NdrFcShort( 0x2 ), /* 2 */
/* 294 */ 0x9, /* Corr desc: FC_ULONG
*/
/* 296 */ NdrFcShort( 0xfffc ), /* -4 */
/* 298 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 300 */ 0x6, /* FC_SHORT */
/* 302 */
FC_END /*
/* 304 */
FC_CSTRUCT /*

```

```

3 */
/* 304 */ NdrFcShort( 0x8 ), /* 8 */
/* 306 */ NdrFcShort( 0xffffffff0 ), /* Offset= -
16 (290) */
/* 308 */ 0x8, /* FC_LONG */
FC_LONG */
/* 310 */ 0x5c, /* FC_PAD */
FC_END */
/* 312 */
FC_IP */
0x2f,
FC_IP */
0x5a,
FC_CONSTANT_IID */
/* 314 */ NdrFcLong( 0x0 ), /* 0 */
/* 318 */ NdrFcShort( 0x0 ), /* 0 */
/* 320 */ NdrFcShort( 0x0 ), /* 0 */
/* 322 */ 0xc0, /* 192 */
0 */
/* 324 */ 0x0, /* 0 */
0 */
/* 326 */ 0x0, /* 0 */
0 */
/* 328 */ 0x0, /* 0 */
70 */
/* 330 */
FC_IP */
0x2f,
FC_IP */
0x5a,
FC_CONSTANT_IID */
/* 332 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 336 */ NdrFcShort( 0x0 ), /* 0 */
/* 338 */ NdrFcShort( 0x0 ), /* 0 */
/* 340 */ 0xc0, /* 192 */
0 */
/* 342 */ 0x0, /* 0 */
0 */
/* 344 */ 0x0, /* 0 */
0 */
/* 346 */ 0x0, /* 0 */
70 */
/* 348 */
FC_UP [pointer_deref] */
/* 350 */ NdrFcShort( 0x2 ), /* Offset= 2 (352) */
/* 352 */
FC_UP */
/* 354 */ NdrFcShort( 0x1e6 ), /* Offset=
486 (840) */
/* 356 */
FC_ENCAPSULATED_UNION */
0x2a,

```

```

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 */
FC_BOGUS_ARRAY */
0x21,
FC_BOGUS_ARRAY */
0x3,
3 */
/* 426 */ NdrFcShort( 0x0 ), /* 0 */
/* 428 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
*/
/* 430 */ NdrFcShort( 0x0 ), /* 0 */
/* 432 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 434 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 438 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 440 */
FC_UP */
/* 442 */ NdrFcShort( 0xffffffff74 ), /* Offset= -
140 (302) */
/* 444 */ 0x5c, /* FC_PAD */
FC_END */
/* 446 */
FC_BOGUS_STRUCT */
0x1a,
3 */
/* 448 */ NdrFcShort( 0x10 ), /* 16 */
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */ NdrFcShort( 0x6 ), /* Offset= 6 (458) */

```

```

/* 454 */ 0x8, /* FC_LONG */
FC_ALIGNM8 */
/* 456 */ 0x36, /* FC_POINTER */
FC_END */
/* 458 */
FC_RP */
/* 460 */ NdrFcShort( 0xffffffffdc ), /* Offset= -
36 (424) */
/* 462 */
FC_BOGUS_ARRAY */
0x21,
3 */
/* 464 */ NdrFcShort( 0x0 ), /* 0 */
/* 466 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
*/
/* 468 */ NdrFcShort( 0x0 ), /* 0 */
/* 470 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 472 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 476 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 478 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0 */
/* 480 */ NdrFcShort( 0xffffffff58 ), /* Offset= -
168 (312) */
/* 482 */ 0x5c, /* FC_PAD */
FC_END */
/* 484 */
FC_BOGUS_STRUCT */
0x1a,
3 */
/* 486 */ NdrFcShort( 0x10 ), /* 16 */
/* 488 */ NdrFcShort( 0x0 ), /* 0 */
/* 490 */ NdrFcShort( 0x6 ), /* Offset= 6 (496) */
/* 492 */ 0x8, /* FC_LONG */
FC_ALIGNM8 */
/* 494 */ 0x36, /* FC_POINTER */
FC_END */
/* 496 */
FC_RP */
/* 498 */ NdrFcShort( 0xffffffffdc ), /* Offset= -
36 (462) */
/* 500 */
FC_BOGUS_ARRAY */
0x21,
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( 0xfffff44 ), /* Offset= -
188 (330) */
/* 520 */ 0x5c, /* FC_PAD */
*/
FC_END /*
/* 522 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 524 */ NdrFcShort( 0x10 ), /* 16 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 530 */ 0x8, /* FC_LONG */
*/
FC_ALIGNM8 /*
/* 532 */ 0x36, /* FC_POINTER */
*/
FC_END /*
/* 534 */
0x11, 0x0, /*
FC_RP */
/* 536 */ NdrFcShort( 0xfffff4dc ), /* Offset= -
36 (500) */
/* 538 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /*
3 */
/* 540 */ NdrFcShort( 0x0 ), /* 0 */
/* 542 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
/*
/* 544 */ NdrFcShort( 0x0 ), /* 0 */
/* 546 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 548 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 552 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 554 */
0x12, 0x0, /*
FC_UP */
/* 556 */ NdrFcShort( 0x176 ), /* Offset=
374 (930) */
/* 558 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 560 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */

```

```

/* 562 */ NdrFcShort( 0x10 ), /* 16 */
/* 564 */ NdrFcShort( 0x0 ), /* 0 */
/* 566 */ NdrFcShort( 0x6 ), /* Offset= 6 (572) */
/* 568 */ 0x8, /* FC_LONG */
0x39, /*
FC_ALIGNM8 */
/* 570 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 572 */
0x11, 0x0, /*
FC_RP */
/* 574 */ NdrFcShort( 0xfffff4dc ), /* Offset= -
36 (538) */
/* 576 */
0x2E, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 578 */ NdrFcLong( 0x2f ), /* 47 */
/* 582 */ NdrFcShort( 0x0 ), /* 0 */
/* 584 */ NdrFcShort( 0x0 ), /* 0 */
/* 586 */ 0xc0, /* 192 */
0x0, /*
0 */
/* 588 */ 0x0, /* 0 */
0x0, /*
0 */
/* 590 */ 0x0, /* 0 */
0x0, /*
0 */
/* 592 */ 0x0, /* 0 */
0x46, /*
70 */
/* 594 */
0x1b, /*
FC_CARRAY */
0x0, /*
0 */
/* 596 */ NdrFcShort( 0x1 ), /* 1 */
/* 598 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 600 */ NdrFcShort( 0x4 ), /* 4 */
/* 602 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 604 */ 0x1, /* FC_BYTE */
0x5b, /*
FC_END */
/* 606 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 608 */ NdrFcShort( 0x18 ), /* 24 */
/* 610 */ NdrFcShort( 0x0 ), /* 0 */
/* 612 */ NdrFcShort( 0xc ), /* Offset= 12 (624) */
/* 614 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 616 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/

```

```

0x0, /*
0 */
/* 618 */ NdrFcShort( 0xffffffff6 ), /* 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( 0xfffff4dc ), /* Offset= -
36 (628) */
/* 666 */
0x1d, /*
FC_SMPARRAY */
0x0, /*
0 */
/* 668 */ NdrFcShort( 0x8 ), /* 8 */

```

```

/* 670 */ 0x1,          /* FC_BYTE */
FC_END /*
/* 672 */
FC_STRUCT */
          0x15,        /*
          0x3,         /*
3 */
/* 674 */ NdrFcShort( 0x10 ), /* 16 */
/* 676 */ 0x8,         /* FC_LONG */
FC_SHORT */
/* 678 */ 0x6,        /* FC_SHORT */
FC_EMBEDDED_COMPLEX /*
/* 680 */ 0x0,        /* 0 */
NdrFcShort( 0xffffffffl
),          /* Offset= -15 (666) */
FC_END /*
/* 684 */
FC_BOGUS_STRUCT */
          0x1a,        /*
          0x3,         /*
3 */
/* 686 */ NdrFcShort( 0x20 ), /* 32 */
/* 688 */ NdrFcShort( 0x0 ), /* 0 */
/* 690 */ NdrFcShort( 0xa ), /* Offset= 10 (700) */
/* 692 */ 0x8,        /* FC_LONG */
FC_ALIGNM8 /*
/* 694 */ 0x36,       /* FC_POINTER */
FC_EMBEDDED_COMPLEX /*
/* 696 */ 0x0,        /* 0 */
NdrFcShort( 0xffffffffe7
),          /* Offset= -25 (672) */
FC_END /*
/* 700 */
          0x11, 0x0,    /*
FC_RP */
/* 702 */ NdrFcShort( 0xffffffff10 ), /* 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 */
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 */
FC_ALIGNM8 /*
/* 726 */ 0x36,       /* FC_POINTER */
FC_END /*
/* 728 */
          0x12, 0x0,    /*
FC_UP */
/* 730 */ NdrFcShort( 0xffffffffe6 ), /* 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 */
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 */
FC_ALIGNM8 /*
/* 754 */ 0x36,       /* FC_POINTER */
FC_END /*
/* 756 */
          0x12, 0x0,    /*
FC_UP */
/* 758 */ NdrFcShort( 0xffffffffe6 ), /* 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 */
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 */
FC_ALIGNM8 /*
/* 782 */ 0x36,       /* FC_POINTER */
FC_END /*
/* 784 */
          0x12, 0x0,    /*
FC_UP */
/* 786 */ NdrFcShort( 0xffffffffe6 ), /* 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 */
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 */
FC_ALIGNM8 /*
/* 810 */ 0x36,       /* FC_POINTER */
FC_END /*
/* 812 */
          0x12, 0x0,    /*
FC_UP */
/* 814 */ NdrFcShort( 0xffffffffe6 ), /* Offset= -
26 (788) */
/* 816 */
          0x15,        /*
FC_STRUCT */

```

```

0x3, /*
3 */
/* 818 */ NdrFcShort( 0x8 ), /* 8 */
/* 820 */ 0x8, /* FC_LONG */
/* FC_LONG */
/* 822 */ 0x5c, /* FC_PAD */
/* FC_END */
/* 824 */
/* FC_CARRAY */
/* 826 */ NdrFcShort( 0x8 ), /* 8 */
/* 828 */ 0x7, /* Corr desc: FC_USHORT */
/* 830 */ NdrFcShort( 0xffc8 ), /* -56 */
/* 832 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 834 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 836 */ NdrFcShort( 0xfffffec ), /* Offset= -20 (816) */
/* 838 */ 0x5c, /* FC_PAD */
/* FC_END */
/* 840 */
/* FC_BOGUS_STRUCT */
/* 842 */ NdrFcShort( 0x38 ), /* 56 */
/* 844 */ NdrFcShort( 0xfffffec ), /* Offset= -20 (824) */
/* 846 */ NdrFcShort( 0x0 ), /* Offset= 0 (846) */
/* 848 */ 0x6, /* FC_SHORT */
/* FC_SHORT */
/* 850 */ 0x38, /* FC_ALIGNM4 */
/* FC_LONG */
/* 852 */ 0x8, /* FC_LONG */
/* FC_EMBEDDED_COMPLEX */
/* 854 */ 0x4, /* 4 */
/* 856 */ NdrFcShort( 0xffffe0d ), /* Offset= -499 (356) */
/* FC_END */
/* 858 */
/* FC_UP */
/* 860 */ NdrFcShort( 0xfffff02 ), /* Offset= -254 (606) */
/* 862 */
/* FC_UP [simple_pointer] */
/* 864 */ 0x1, /* FC_BYTE */

```

```

FC_PAD */
/* 866 */
FC_UP [simple_pointer] */
/* 868 */ 0x6, /* FC_SHORT */
FC_PAD */
/* 870 */
FC_UP [simple_pointer] */
/* 872 */ 0x8, /* FC_LONG */
FC_PAD */
/* 874 */
FC_UP [simple_pointer] */
/* 876 */ 0xa, /* FC_FLOAT */
FC_PAD */
/* 878 */
FC_UP [simple_pointer] */
/* 880 */ 0xc, /* FC_DOUBLE */
FC_PAD */
/* 882 */
FC_UP */
/* 884 */ NdrFcShort( 0xffffda4 ), /* Offset= -604 (280) */
/* 886 */
FC_UP [pointer_deref] */
/* 888 */ NdrFcShort( 0xffffda6 ), /* Offset= -602 (286) */
/* 890 */
FC_UP [pointer_deref] */
/* 892 */ NdrFcShort( 0xffffdbc ), /* Offset= -580 (312) */
/* 894 */
FC_UP [pointer_deref] */
/* 896 */ NdrFcShort( 0xffffdca ), /* Offset= -566 (330) */
/* 898 */
FC_UP [pointer_deref] */
/* 900 */ NdrFcShort( 0xffffdd8 ), /* Offset= -552 (348) */
/* 902 */
FC_UP [pointer_deref] */
/* 904 */ NdrFcShort( 0x2 ), /* Offset= 2 (906) */
/* 906 */
FC_UP */
/* 908 */ NdrFcShort( 0x16 ), /* Offset= 22 (930) */
/* 910 */
FC_STRUCT */

```

```

0x7, /*
7 */
/* 912 */ NdrFcShort( 0x10 ), /* 16 */
/* 914 */ 0x6, /* FC_SHORT */
/* FC_BYTE */
/* 916 */ 0x1, /* FC_BYTE */
FC_ALIGNM4 */
/* 918 */ 0x8, /* FC_LONG */
FC_ALIGNM8 */
/* 920 */ 0xb, /* FC_HYPER */
FC_END */
/* 922 */
/* FC_UP */
/* 924 */ NdrFcShort( 0xfffff2 ), /* Offset= -14 (910) */
/* 926 */
FC_UP [simple_pointer] */
/* 928 */ 0x2, /* FC_CHAR */
FC_PAD */
/* 930 */
FC_BOGUS_STRUCT */
/* 932 */ NdrFcShort( 0x20 ), /* 32 */
/* 934 */ NdrFcShort( 0x0 ), /* 0 */
/* 936 */ NdrFcShort( 0x0 ), /* Offset= 0 (936) */
/* 938 */ 0x8, /* FC_LONG */
/* FC_LONG */
/* 940 */ 0x6, /* FC_SHORT */
FC_SHORT */
/* 942 */ 0x6, /* FC_SHORT */
FC_SHORT */
/* 944 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 946 */ NdrFcShort( 0xffffc54 ), /* Offset= -940 (6) */
/* 948 */ 0x5c, /* FC_PAD */
FC_END */
/* 950 */ 0xb4, /* FC_USER_MARSHAL */
131 */
/* 952 */ NdrFcShort( 0x0 ), /* 0 */
/* 954 */ NdrFcShort( 0x18 ), /* 24 */
/* 956 */ NdrFcShort( 0x0 ), /* 0 */
/* 958 */ NdrFcShort( 0xffffc44 ), /* Offset= -956 (2) */
/* 960 */
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_ps.h



---


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

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

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:08 2001
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=i2), 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*/
#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("FEEB6AA2-84B1-11d2-BA47-
00C04PBF08B")
    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,

```

```

0;          /* [out] */ VARIANT __RPC_FAR *txn_out) =
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 STDMETHODCALLTYPE 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 STDMETHODCALLTYPE 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 STDMETHODCALLTYPE 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 STDMETHODCALLTYPE 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 STDMETHODCALLTYPE 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_p.c

```

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

/* this ALWAYS GENERATED file contains the proxy stub
code */

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:08 2001
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), 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_HEADER( )

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

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

#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of
<rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 997
#define PROC_FORMAT_STRING_SIZE 193
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 1

typedef struct _MIDL_TYPE_FORMAT_STRING
{
    short         Pad;
    unsigned char Format[ TYPE_FORMAT_STRING_SIZE ];
} MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
{
    short         Pad;
    unsigned char Format[ PROC_FORMAT_STRING_SIZE ];
} MIDL_PROC_FORMAT_STRING;

extern const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING
__MIDL_ProcFormatString;

/* Standard interface: __MIDL_itf_tpcc_com_ps_0000,
ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xc0,0x00,0x00,0x00,0
x00,0x00,0x00,0x46}} */

```

```

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEFE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0
x4F,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
};

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 =
{

```



```

/* 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, */
3 */ 0x3, /*
*/

/* 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
#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
#endif
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
#endif
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, */
3 */ 0x3, /*
*/

```

```

/* 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
#endif
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
#endif
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
#endif

```

```

        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 */
#ifdef _ALPHA_
#ifdef _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, */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 120 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
#else
        NdrFcShort( 0x8 ), /*
MIPS Stack size/offset = 8 */
#endif
#else
        NdrFcShort( 0x8 ), /*
PPC Stack size/offset = 8 */
#endif
#else
        NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
#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 */
#ifdef _ALPHA_
#ifdef _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, */
#ifdef _ALPHA_
#ifdef _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 */
#ifdef _ALPHA_
#ifdef _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, */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 154 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
#else
        NdrFcShort( 0x8 ), /*
MIPS Stack size/offset = 8 */
#endif
#else
        NdrFcShort( 0x8 ), /*
PPC Stack size/offset = 8 */
#endif
#else
        NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 156 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Parameter txn_out */

/* 158 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
#ifdef _ALPHA_
#ifdef _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
#endif

```

```

#else
                                NdrFcShort( 0x18 ), /*
Alpha Stack size/offset = 24 */
#endif
/* 162 */ NdrFcShort( 0x3da ), /* Type
Offset=986 */

/* Return value */

/* 164 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 166 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
#else
                                NdrFcShort( 0x1c ), /*
MIPS Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x1c ), /*
PPC Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x20 ), /*
Alpha Stack size/offset = 32 */
#endif
/* 168 */ 0x8, /* FC_LONG */
0 */
                                0x0, /*

/* 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 */
0 */
                                0x0, /*

}
};

static const MIDL_TYPE_FORMAT_STRING
_MIDL_TypeFormatString =
{
    0,
    {
        NdrFcShort( 0x0 ), /*
0 */
/* 2 */
                                0x12, 0x0, /*
FC_UP */
/* 4 */ NdrFcShort( 0x3b0 ), /* Offset=
944 (948) */
/* 6 */
                                0x2b, /*
FC_NON_ENCAPSULATED_UNION */
                                0x9, /*
FC_ULONG */
/* 8 */ 0x7, /* Corr desc: FC_USHORT
*/
                                0x0, /*

/*
/* 10 */ NdrFcShort( 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 */
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( 0xffc ), /* -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 */

```



```

FC_END */
                                0x5b,      /*
/* 448 */ 0x5c,
                                /* FC_PAD */
                                0x5b,      /*
FC_END */
/* 450 */
                                0x16,      /*
FC_PSTRUCT */
                                0x3,        /*
3 */
/* 452 */ NdrFcShort( 0x8 ), /* 8 */
/* 454 */
                                0x4b,      /*
FC_PP */
                                0x5c,      /*
FC_PAD */
/* 456 */
                                0x46,      /*
FC_NO_REPEAT */
                                0x5c,      /*
FC_PAD */
/* 458 */ NdrFcShort( 0x4 ), /* 4 */
/* 460 */ NdrFcShort( 0x4 ), /* 4 */
/* 462 */ 0x11, 0x0, /* FC_RP */
/* 464 */ NdrFcShort( 0xfffffd4 ), /* Offset= -
44 (420) */
/* 466 */
                                0x5b,      /*
FC_END */
                                0x8,        /*
FC_LONG */
/* 468 */ 0x8,
                                /* FC_LONG */
                                0x5b,      /*
FC_END */
/* 470 */
                                0x21,      /*
FC_BOGUS_ARRAY */
                                0x3,        /*
3 */
/* 472 */ NdrFcShort( 0x0 ), /* 0 */
/* 474 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
                                0x0,      /*
*/
/* 476 */ NdrFcShort( 0x0 ), /* 0 */
/* 478 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 482 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
                                0x0,      /*
0 */
/* 484 */ NdrFcShort( 0xfffff50 ), /* 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( 0xffffffe0 ), /* 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( 0xfffff40 ), /* Offset= -
192 (326) */
/* 520 */ 0x5c, /* FC_PAD */
                                0x5b,      /*
FC_END */
/* 522 */
                                0x1a,      /*
FC_BOGUS_STRUCT */
                                0x3,        /*
3 */
/* 524 */ NdrFcShort( 0x8 ), /* 8 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 530 */ 0x8, /* FC_LONG */
                                0x36,      /*
FC_POINTER */
/* 532 */ 0x5c, /* FC_PAD */
                                0x5b,      /*
FC_END */
/* 534 */
                                0x11, 0x0, /*
FC_RP */
/* 536 */ NdrFcShort( 0xffffffe0 ), /* Offset= -
32 (504) */
/* 538 */
                                0x1b,      /*
FC_CARRAY */
                                0x3,        /*
3 */
/* 540 */ NdrFcShort( 0x4 ), /* 4 */

```

```

/* 542 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
                                0x0,      /*
*/
/* 544 */ NdrFcShort( 0x0 ), /* 0 */
/* 546 */
                                0x4b,      /*
FC_PP */
                                0x5c,      /*
FC_PAD */
/* 548 */
                                0x48,      /*
FC_VARIABLE_REPEAT */
                                0x49,      /*
FC_FIXED_OFFSET */
/* 550 */ NdrFcShort( 0x4 ), /* 4 */
/* 552 */ NdrFcShort( 0x0 ), /* 0 */
/* 554 */ NdrFcShort( 0x1 ), /* 1 */
/* 556 */ NdrFcShort( 0x0 ), /* 0 */
/* 558 */ NdrFcShort( 0x0 ), /* 0 */
/* 560 */ 0x12, 0x0, /* FC_UP */
/* 562 */ NdrFcShort( 0x182 ), /* Offset=
386 (948) */
/* 564 */
                                0x5b,      /*
FC_END */
                                0x8,        /*
FC_LONG */
/* 566 */ 0x5c, /* FC_PAD */
                                0x5b,      /*
FC_END */
/* 568 */
                                0x1a,      /*
FC_BOGUS_STRUCT */
                                0x3,        /*
3 */
/* 570 */ NdrFcShort( 0x8 ), /* 8 */
/* 572 */ NdrFcShort( 0x0 ), /* 0 */
/* 574 */ NdrFcShort( 0x6 ), /* Offset= 6 (580) */
/* 576 */ 0x8, /* FC_LONG */
                                0x36,      /*
FC_POINTER */
/* 578 */ 0x5c, /* FC_PAD */
                                0x5b,      /*
FC_END */
/* 580 */
                                0x11, 0x0, /*
FC_RP */
/* 582 */ NdrFcShort( 0xfffffd4 ), /* Offset= -
44 (538) */
/* 584 */
                                0x2E,      /*
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 */

```

```

0 */
/* 598 */ 0x0, /* 0 */
0x0, /*
0 */
/* 600 */ 0x0, /* 0 */
0x46, /*
70 */
/* 602 */
FC_CARRAY */
0x1b, /*
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 */
FC_VARIABLE_REPEAT */
0x48, /*
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 */ 0x1, /* FC_BYTE */
0x5b, /*
FC_END */
/* 684 */
0x15, /*
FC_STRUCT */
0x3, /*
3 */
/* 686 */ NdrFcShort( 0x10 ), /* 16 */
/* 688 */ 0x8, /* FC_LONG */
0x6, /*
FC_SHORT */
/* 690 */ 0x6, /* FC_SHORT */
0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 692 */ 0x0, /* 0 */

```

```

NdrFcShort( 0xffffffff
), /* 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( 0xfffffe8 ), /* Offset= -
24 (684) */
/* 710 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 712 */
0x11, 0x0, /*
FC_RP */
/* 714 */ NdrFcShort( 0xfffff0c ), /* 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( 0xffffffff ), /* Offset= -
24 (716) */
/* 742 */
FC_END */
0x5b, /*
FC_LONG */
/* 744 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */
/* 746 */
FC_CARRAY */
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 */
FC_PSTRUCT */
0x16, /*
3 */
/* 758 */ NdrFcShort( 0x8 ), /* 8 */
/* 760 */
FC_PP */
0x4b, /*
FC_PAD */
/* 762 */
0x5c, /*
FC_NO_REPEAT */
0x46, /*
0x5c, /*
FC_PAD */
/* 764 */ NdrFcShort( 0x4 ), /* 4 */
/* 766 */ NdrFcShort( 0x4 ), /* 4 */
/* 768 */ 0x12, 0x0, /* FC_UP */
/* 770 */ NdrFcShort( 0xffffffff ), /* Offset= -
24 (746) */
/* 772 */
FC_END */
0x5b, /*
FC_LONG */
/* 774 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */
/* 776 */
FC_CARRAY */
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 */
FC_PSTRUCT */
0x16, /*
0x3, /*
3 */
/* 788 */ NdrFcShort( 0x8 ), /* 8 */
/* 790 */
FC_PP */
0x4b, /*
FC_PAD */
0x5c, /*
/* 792 */
FC_NO_REPEAT */
0x46, /*
0x5c, /*
FC_PAD */
/* 794 */ NdrFcShort( 0x4 ), /* 4 */
/* 796 */ NdrFcShort( 0x4 ), /* 4 */
/* 798 */ 0x12, 0x0, /* FC_UP */
/* 800 */ NdrFcShort( 0xffffffff ), /* Offset= -
24 (776) */
/* 802 */
FC_END */
0x5b, /*
FC_LONG */
0x8, /*
/* 804 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */
/* 806 */
FC_CARRAY */
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 */
FC_PSTRUCT */
0x16, /*
0x3, /*
3 */
/* 818 */ NdrFcShort( 0x8 ), /* 8 */
/* 820 */
FC_PP */
0x4b, /*
FC_PAD */
0x5c, /*
/* 822 */

```

```

0x46, /*
FC_NO_REPEAT */
0x5c, /*
FC_PAD */
/* 824 */ NdrFcShort( 0x4 ), /* 4 */
/* 826 */ NdrFcShort( 0x4 ), /* 4 */
/* 828 */ 0x12, 0x0, /* FC_UP */
/* 830 */ NdrFcShort( 0xffffffff ), /* Offset= -
24 (806) */
/* 832 */
FC_END */
0x5b, /*
FC_LONG */
0x8, /*
/* 834 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */
/* 836 */
FC_STRUCT */
0x15, /*
0x3, /*
3 */
/* 838 */ NdrFcShort( 0x8 ), /* 8 */
/* 840 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 842 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 844 */
FC_CARRAY */
0x1b, /*
FC_CARRAY */
0x3, /*
3 */
/* 846 */ NdrFcShort( 0x8 ), /* 8 */
/* 848 */ 0x7, /* Corr desc: FC_USHORT
*/
0x0, /*
*/
/* 850 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 852 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 854 */ NdrFcShort( 0xffffffff ), /* Offset= -
18 (836) */
/* 856 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 858 */
FC_BOGUS_STRUCT */
0x1a, /*
0x3, /*
3 */
/* 860 */ NdrFcShort( 0x28 ), /* 40 */
/* 862 */ NdrFcShort( 0xffffffff ), /* Offset= -
18 (844) */
/* 864 */ NdrFcShort( 0x0 ), /* Offset= 0 (864) */
/* 866 */ 0x6, /* FC_SHORT */
0x6, /*
FC_SHORT */
/* 868 */ 0x38, /* FC_ALIGN4 */

```

```

0x8, /*
FC_LONG */
/* 870 */ 0x8, /* FC_LONG */
0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 872 */ 0x0, /* 0 */
NdrFcShort( 0xfffffd7
), /* Offset= -521 (352) */
0x5b, /*
FC_END */
/* 876 */
0x12, 0x0, /*
FC_UP */
/* 878 */ NdrFcShort( 0xfffffe6 ), /* Offset= -
266 (612) */
/* 880 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 882 */ 0x1, /* FC_BYTE */
0x5c, /*
FC_PAD */
/* 884 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 886 */ 0x6, /* FC_SHORT */
0x5c, /*
FC_PAD */
/* 888 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 890 */ 0x8, /* FC_LONG */
0x5c, /*
FC_PAD */
/* 892 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 894 */ 0xa, /* FC_FLOAT */
0x5c, /*
FC_PAD */
/* 896 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 898 */ 0xc, /* FC_DOUBLE */
0x5c, /*
FC_PAD */
/* 900 */
0x12, 0x0, /*
FC_UP */
/* 902 */ NdrFcShort( 0xffffd90 ), /* Offset= -
624 (278) */
/* 904 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 906 */ NdrFcShort( 0xffffd92 ), /* Offset= -
622 (284) */
/* 908 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 910 */ NdrFcShort( 0xffffda6 ), /* Offset= -
602 (308) */
/* 912 */
0x12, 0x10, /*
FC_UP [pointer_deref] */

```

```

/* 914 */ NdrFcShort( 0xffffdb4 ), /* Offset= -
588 (326) */
/* 916 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 918 */ NdrFcShort( 0xffffdc2 ), /* Offset= -
574 (344) */
/* 920 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 922 */ NdrFcShort( 0x2 ), /* Offset= 2 (924) */
/* 924 */
0x12, 0x0, /*
FC_UP */
/* 926 */ NdrFcShort( 0x16 ), /* Offset= 22 (948) */
/* 928 */
0x15, /*
FC_STRUCT */
0x7, /*
7 */
/* 930 */ NdrFcShort( 0x10 ), /* 16 */
/* 932 */ 0x6, /* FC_SHORT */
0x1, /*
FC_BYTE */
/* 934 */ 0x1, /* FC_BYTE */
0x38, /*
FC_ALIGNM4 */
/* 936 */ 0x8, /* FC_LONG */
0x39, /*
FC_ALIGNM8 */
/* 938 */ 0xb, /* FC_HYPER */
0x5b, /*
FC_END */
/* 940 */
0x12, 0x0, /*
FC_UP */
/* 942 */ NdrFcShort( 0xfffffff2 ), /* Offset= -
14 (928) */
/* 944 */
0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 946 */ 0x2, /* FC_CHAR */
0x5c, /*
FC_PAD */
/* 948 */
0x1a, /*
FC_BOGUS_STRUCT */
0x7, /*
7 */
/* 950 */ NdrFcShort( 0x20 ), /* 32 */
/* 952 */ NdrFcShort( 0x0 ), /* 0 */
/* 954 */ NdrFcShort( 0x0 ), /* Offset= 0 (954) */
/* 956 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 958 */ 0x6, /* FC_SHORT */
0x6, /*
FC_SHORT */
/* 960 */ 0x6, /* FC_SHORT */
0x6, /*
FC_SHORT */
/* 962 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/

```

```

0x0, /*
0 */
/* 964 */ NdrFcShort( 0xfffffc42 ), /* Offset= -
958 (6) */
/* 966 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 968 */ 0xb4, /* FC_USER_MARSHAL */
0x83, /*
131 */
/* 970 */ NdrFcShort( 0x0 ), /* 0 */
/* 972 */ NdrFcShort( 0x10 ), /* 16 */
/* 974 */ NdrFcShort( 0x0 ), /* 0 */
/* 976 */ NdrFcShort( 0xfffffc32 ), /* Offset= -
974 (2) */
/* 978 */
0x11, 0x4, /*
FC_RP [allocated_on_stack] */
/* 980 */ NdrFcShort( 0x6 ), /* Offset= 6 (986) */
/* 982 */
0x13, 0x0, /*
FC_OP */
/* 984 */ NdrFcShort( 0xffffffd4 ), /* 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 *
_tpc_com_ps_ProxyVtblList[] =
{
( CInterfaceProxyVtbl * ) &ITPCCProxyVtbl,
0
};
const CInterfaceStubVtbl *
_tpc_com_ps_StubVtblList[] =
{
( CInterfaceStubVtbl * ) &ITPCCStubVtbl,
0
};
PCInterfaceName const
_tpc_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)*/

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

/* this ALWAYS GENERATED file contains the proxy stub
code */

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:08 2001
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=i2), 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_HEADERING( )

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

```

```

/* verify that the <rpcproxy.h> version is high
enough to compile this file*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 475
#endif

#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of
<rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 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,{0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0
x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFE6E6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0
x4F,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 */
            FC_AUTO_HANDLE *          0x33,          /*
                                                    */
            Old Flags: object, Oi2 /*
            /* 2 */ NdrFcLong( 0x0 ), /* 0 */
            /* 6 */ NdrFcShort( 0x3 ), /* 3 */
            #ifndef _ALPHA_
            /* 8 */ NdrFcShort( 0x38 ), /* ia64 Stack
            size/offset = 56 */
            #else

```

```

NdrFcShort( 0x30 ), /*
axp64 Stack size/offset = 48 */
#endif
/* 10 */ NdrFcShort( 0x0 ), /* 0 */
/* 12 */ NdrFcShort( 0x8 ), /* 8 */
/* 14 */ 0x47, /* Oi2 Flags: srv must
size, clt must size, has return, has ext, */
3 /*
/* 16 */ 0xa, /* 10 */
/* 18 */ 0x7, /*
Ext Flags: new corr desc, clt corr check, srv corr
check, */
/* 18 */ NdrFcShort( 0x20 ), /* 32 */
/* 20 */ NdrFcShort( 0x20 ), /* 32 */
/* 22 */ NdrFcShort( 0x0 ), /* 0 */
/* 24 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 26 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#ifndef _ALPHA_
/* 28 */ NdrFcShort( 0x10 ), /* ia64 Stack
size/offset = 16 */
#else
NdrFcShort( 0x8 ), /*
axp64 Stack size/offset = 8 */
#endif
/* 30 */ NdrFcShort( 0x3b6 ), /* Type
Offset=950 */

/* Parameter txn_out */

/* 32 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
#ifndef _ALPHA_
/* 34 */ NdrFcShort( 0x28 ), /* ia64 Stack
size/offset = 40 */
#else
NdrFcShort( 0x20 ), /*
axp64 Stack size/offset = 32 */
#endif
/* 36 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Return value */

/* 38 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_
/* 40 */ NdrFcShort( 0x30 ), /* ia64 Stack
size/offset = 48 */
#else
NdrFcShort( 0x28 ), /*
axp64 Stack size/offset = 40 */
#endif
/* 42 */ 0x8, /* FC_LONG */
0 /*
0x0, /*
*/

/* Procedure Payment */

```

```

/* 44 */ 0x33, /* FC_AUTO_HANDLE */
/* 46 */ NdrFcLong( 0x0 ), /* 0 */
/* 50 */ NdrFcShort( 0x4 ), /* 4 */
#ifndef _ALPHA_
/* 52 */ NdrFcShort( 0x38 ), /* ia64 Stack
size/offset = 56 */
#else
NdrFcShort( 0x30 ), /*
axp64 Stack size/offset = 48 */
#endif
/* 54 */ NdrFcShort( 0x0 ), /* 0 */
/* 56 */ NdrFcShort( 0x8 ), /* 8 */
/* 58 */ 0x47, /* Oi2 Flags: srv must
size, clt must size, has return, has ext, */
3 /*
/* 60 */ 0xa, /* 10 */
/* 62 */ 0x7, /*
Ext Flags: new corr desc, clt corr check, srv corr
check, */
/* 62 */ NdrFcShort( 0x20 ), /* 32 */
/* 64 */ NdrFcShort( 0x20 ), /* 32 */
/* 66 */ NdrFcShort( 0x0 ), /* 0 */
/* 68 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 70 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#ifndef _ALPHA_
/* 72 */ NdrFcShort( 0x10 ), /* ia64 Stack
size/offset = 16 */
#else
NdrFcShort( 0x8 ), /*
axp64 Stack size/offset = 8 */
#endif
/* 74 */ NdrFcShort( 0x3b6 ), /* Type
Offset=950 */

/* Parameter txn_out */

/* 76 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
#ifndef _ALPHA_
/* 78 */ NdrFcShort( 0x28 ), /* ia64 Stack
size/offset = 40 */
#else
NdrFcShort( 0x20 ), /*
axp64 Stack size/offset = 32 */
#endif
/* 80 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Return value */

/* 82 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_

```

```

/* 84 */ NdrFcShort( 0x30 ), /* ia64 Stack
size/offset = 48 */
#else
NdrFcShort( 0x28 ), /*
axp64 Stack size/offset = 40 */
#endif
/* 86 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure Delivery */

/* 88 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 90 */ NdrFcLong( 0x0 ), /* 0 */
/* 94 */ NdrFcShort( 0x5 ), /* 5 */
#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 */
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 */
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 */
FC_STRUCT /*
0x15, /*
7 */
0x7, /*
/* 282 */ NdrFcShort( 0x8 ), /* 8 */
/* 284 */ 0xb, /* FC_HYPER */
0x5b, /*
FC_END */
/* 286 */
0x12, 0x0, /*
FC_UP */
/* 288 */ NdrFcShort( 0xe ), /* Offset= 14 (302) */
/* 290 */
0x1b, /*
FC_CARRAY */
0x1, /*
1 */
/* 292 */ NdrFcShort( 0x2 ), /* 2 */
/* 294 */ 0x9, /* Corr desc: FC_ULONG
*/

```

```

0x0, /*
*/
/* 296 */ NdrFcShort( 0xfffc ), /* -4 */
/* 298 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 300 */ 0x6, /* FC_SHORT */
0x5b, /*
FC_END */
/* 302 */
0x17, /*
FC_CSTRUCT */
0x3, /*
3 */
/* 304 */ NdrFcShort( 0x8 ), /* 8 */
/* 306 */ NdrFcShort( 0xfffff0 ), /* Offset= -
16 (290) */
/* 308 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 310 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 312 */
0x2E, /*
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 */
0x2E, /*
0x5a, /*
FC_CONSTANT_IID */
/* 332 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 336 */ NdrFcShort( 0x0 ), /* 0 */
/* 338 */ NdrFcShort( 0x0 ), /* 0 */
/* 340 */ 0xc0, /* 192 */
0x0, /*
0 */
/* 342 */ 0x0, /* 0 */
0x0, /*
0 */
/* 344 */ 0x0, /* 0 */
0x0, /*
0 */
/* 346 */ 0x0, /* 0 */
0x46, /*
70 */
/* 348 */

```

```

0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 350 */ NdrFcShort( 0x2 ), /* Offset= 2 (352) */
/* 352 */
0x12, 0x0, /*
FC_UP */
/* 354 */ NdrFcShort( 0x1e6 ), /* Offset=
486 (840) */
/* 356 */
0x2a, /*
FC_ENCAPSULATED_UNION */
0x89, /*
137 */
/* 358 */ NdrFcShort( 0x20 ), /* 32 */
/* 360 */ NdrFcShort( 0xa ), /* 10 */
/* 362 */ NdrFcLong( 0x8 ), /* 8 */
/* 366 */ NdrFcShort( 0x50 ), /* Offset= 80 (446) */
/* 368 */ NdrFcLong( 0xd ), /* 13 */
/* 372 */ NdrFcShort( 0x70 ), /* Offset= 112 (484) */
/* 374 */ NdrFcLong( 0x9 ), /* 9 */
/* 378 */ NdrFcShort( 0x90 ), /* Offset= 144 (522) */
/* 380 */ NdrFcLong( 0xc ), /* 12 */
/* 384 */ NdrFcShort( 0xb0 ), /* Offset= 176 (560) */
/* 386 */ NdrFcLong( 0x24 ), /* 36 */
/* 390 */ NdrFcShort( 0x104 ), /* Offset=
260 (650) */
/* 392 */ NdrFcLong( 0x800d ), /* 32781 */
/* 396 */ NdrFcShort( 0x120 ), /* Offset=
288 (684) */
/* 398 */ NdrFcLong( 0x10 ), /* 16 */
/* 402 */ NdrFcShort( 0x13a ), /* Offset=
314 (716) */
/* 404 */ NdrFcLong( 0x2 ), /* 2 */
/* 408 */ NdrFcShort( 0x150 ), /* Offset=
336 (744) */
/* 410 */ NdrFcLong( 0x3 ), /* 3 */
/* 414 */ NdrFcShort( 0x166 ), /* Offset=
358 (772) */
/* 416 */ NdrFcLong( 0x14 ), /* 20 */
/* 420 */ NdrFcShort( 0x17c ), /* Offset=
380 (800) */
/* 422 */ NdrFcShort( 0xffffffff ), /* Offset= -1
(421) */
/* 424 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /*
3 */
/* 426 */ NdrFcShort( 0x0 ), /* 0 */
/* 428 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 430 */ NdrFcShort( 0x0 ), /* 0 */
/* 432 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 434 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 438 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 440 */
0x12, 0x0, /*
FC_UP */
/* 442 */ NdrFcShort( 0xffffffff74 ), /* Offset= -
140 (302) */

```

```

/* 444 */ 0x5c,          /* FC_PAD */
FC_END /*
/* 446 */
FC_BOGUS_STRUCT */
0x1a,          /*
0x3,          /*
3 */
/* 448 */ NdrFcShort( 0x10 ), /* 16 */
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */ NdrFcShort( 0x6 ), /* Offset= 6 (458) */
/* 454 */ 0x8,          /* FC_LONG */
0x39,          /*
FC_ALIGNM8 */
/* 456 */ 0x36,        /* FC_POINTER */
0x5b,          /*
FC_END /*
/* 458 */
0x11, 0x0,     /*
FC_RP */
/* 460 */ NdrFcShort( 0xfffff5dc ), /* 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( 0xfffffff ), /* -1 */
/* 476 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 478 */ 0x4c,        /* FC_EMBEDDED_COMPLEX
*/
0x0,          /*
0 */
/* 480 */ NdrFcShort( 0xfffff58 ), /* Offset= -
168 (312) */
/* 482 */ 0x5c,        /* FC_PAD */
0x5b,          /*
FC_END /*
/* 484 */
0x1a,          /*
FC_BOGUS_STRUCT */
0x3,          /*
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( 0xfffff5dc ), /* 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( 0xfffffff ), /* -1 */
/* 514 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 516 */ 0x4c,        /* FC_EMBEDDED_COMPLEX
*/
0x0,          /*
0 */
/* 518 */ NdrFcShort( 0xfffff44 ), /* Offset= -
188 (330) */
/* 520 */ 0x5c,        /* FC_PAD */
0x5b,          /*
FC_END /*
/* 522 */
0x1a,          /*
FC_BOGUS_STRUCT */
0x3,          /*
3 */
/* 524 */ NdrFcShort( 0x10 ), /* 16 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 530 */ 0x8,          /* FC_LONG */
0x39,          /*
FC_ALIGNM8 */
/* 532 */ 0x36,        /* FC_POINTER */
0x5b,          /*
FC_END /*
/* 534 */
0x11, 0x0,     /*
FC_RP */
/* 536 */ NdrFcShort( 0xfffff5dc ), /* 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( 0xfffffff ), /* -1 */
/* 552 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 554 */
0x12, 0x0,     /*
FC_UP */

```

```

/* 556 */ NdrFcShort( 0x176 ), /* Offset=
374 (930) */
/* 558 */ 0x5c,        /* FC_PAD */
0x5b,          /*
FC_END /*
/* 560 */
0x1a,          /*
FC_BOGUS_STRUCT */
0x3,          /*
0x3,          /*
3 */
/* 562 */ NdrFcShort( 0x10 ), /* 16 */
/* 564 */ NdrFcShort( 0x0 ), /* 0 */
/* 566 */ NdrFcShort( 0x6 ), /* Offset= 6 (572) */
/* 568 */ 0x8,          /* FC_LONG */
0x39,          /*
FC_ALIGNM8 */
/* 570 */ 0x36,        /* FC_POINTER */
0x5b,          /*
FC_END /*
/* 572 */
0x11, 0x0,     /*
FC_RP */
/* 574 */ NdrFcShort( 0xfffff5dc ), /* Offset= -
36 (538) */
/* 576 */
0x2f,          /*
FC_IP */
0x5a,          /*
FC_CONSTANT_IID */
/* 578 */ NdrFcLong( 0x2f ), /* 47 */
/* 582 */ NdrFcShort( 0x0 ), /* 0 */
/* 584 */ NdrFcShort( 0x0 ), /* 0 */
/* 586 */ 0xc0,        /* 192 */
0x0,          /*
0 */
/* 588 */ 0x0,        /* 0 */
0x0,          /*
0 */
/* 590 */ 0x0,        /* 0 */
0x0,          /*
0 */
/* 592 */ 0x0,        /* 0 */
0x46,         /*
70 */
/* 594 */
0x1b,          /*
FC_CARRAY */
0x0,          /*
0 */
/* 596 */ NdrFcShort( 0x1 ), /* 1 */
/* 598 */ 0x19,        /* Corr desc: field
pointer, FC_ULONG */
0x0,          /*
*/
/* 600 */ NdrFcShort( 0x4 ), /* 4 */
/* 602 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 604 */ 0x1,        /* FC_BYTE */
0x5b,          /*
FC_END /*
/* 606 */
0x1a,          /*
FC_BOGUS_STRUCT */

```

```

0x3, /*
3 */
/* 608 */ NdrFcShort( 0x18 ), /* 24 */
/* 610 */ NdrFcShort( 0x0 ), /* 0 */
/* 612 */ NdrFcShort( 0xc ), /* Offset= 12 (624) */
/* 614 */ 0x8, /* FC_LONG */
FC_LONG /*
/* 616 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /*
0 */
/* 618 */ NdrFcShort( 0xfffffd6 ), /* Offset= -
42 (576) */
/* 620 */ 0x39, /* FC_ALIGNM8 */
FC_POINTER /*
/* 622 */ 0x5c, /* FC_PAD */
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 */
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 */
FC_ALIGNM8 /*
/* 660 */ 0x36, /* FC_POINTER */
FC_END /*
/* 662 */

```

```

0x11, 0x0, /*
FC_RP /*
/* 664 */ NdrFcShort( 0xfffffdc ), /* Offset= -
36 (628) */
/* 666 */
0x1d, /*
FC_SMPARRAY /*
0x0, /*
0 */
/* 668 */ NdrFcShort( 0x8 ), /* 8 */
/* 670 */ 0x1, /* FC_BYTE */
FC_END /*
/* 672 */
0x15, /*
FC_STRUCT /*
0x3, /*
3 */
/* 674 */ NdrFcShort( 0x10 ), /* 16 */
/* 676 */ 0x8, /* FC_LONG */
FC_SHORT /*
/* 678 */ 0x6, /* FC_SHORT */
FC_EMBEDDED_COMPLEX /*
/* 680 */ 0x0, /* 0 */
NdrFcShort( 0xfffffff1
), /* Offset= -15 (666) */
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 */
FC_ALIGNM8 /*
/* 694 */ 0x36, /* FC_POINTER */
FC_EMBEDDED_COMPLEX /*
/* 696 */ 0x0, /* 0 */
NdrFcShort( 0xffffffe7
), /* Offset= -25 (672) */
FC_END /*
/* 700 */
0x11, 0x0, /*
FC_RP /*
/* 702 */ NdrFcShort( 0xfffff10 ), /* 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 */
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 */
FC_ALIGNM8 /*
/* 726 */ 0x36, /* FC_POINTER */
FC_END /*
/* 728 */
0x12, 0x0, /*
FC_UP /*
/* 730 */ NdrFcShort( 0xfffffe6 ), /* 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 */
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 */
FC_ALIGNM8 /*
/* 754 */ 0x36, /* FC_POINTER */
FC_END /*
/* 756 */
0x12, 0x0, /*
FC_UP /*
/* 758 */ NdrFcShort( 0xfffffe6 ), /* 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( 0xffffffffe6 ), /* 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( 0xffffffffe6 ), /* 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( 0xffffffffec ), /* Offset= -
20 (816) */
/* 838 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 840 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 842 */ NdrFcShort( 0x38 ), /* 56 */
/* 844 */ NdrFcShort( 0xffffffffec ), /* 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( 0xffffffff02 ), /* 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( 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_ps_p.c

```

#pragma warning( disable: 4049 ) /* more than 64k
source lines */
/* this ALWAYS GENERATED file contains the proxy stub
code */
/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:08 2001
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=12), 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)
#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={0xFE6E6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0x00,0x4F,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,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.3.280 */
    0,
    UserMarshalRoutines,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* Reserved3 */
    0, /* Reserved4 */
    0 /* Reserved5 */
};

#pragma data_seg(".rdata")

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

#if !defined(__RPC_WIN32__)
#error Invalid build platform for this stub.
#endif

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this
stub because it uses these features:
#error -Oif or -Oicf, [wire_marshal] or
[user_marshal] attribute.
#error However, your C/C++ compilation flags indicate
you intend to run this app on earlier systems.
#error This app will die there with the
RPC_X_WRONG_STUB_VERSION error.
#endif

static const MIDL_PROC_FORMAT_STRING
__MIDL_ProcFormatString =
{
    0,
    {
        /* Procedure NewOrder */

```

```

                                0x33,          /*
FC_AUTO_HANDLE */
                                0x6c,          /*
Old Flags:  object, Oi2 */
/* 2 */ NdrFcLong( 0x0 ), /* 0 */
/* 6 */ NdrFcShort( 0x3 ), /* 3 */
#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( 0x20 ), /*
Alpha 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_
#ifdef _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 */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 58 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
#else
                                NdrFcShort( 0x18 ), /*
MIPS Stack size/offset = 24 */
#endif
#else
                                NdrFcShort( 0x18 ), /*
PPC Stack size/offset = 24 */
#endif
#else
                                NdrFcShort( 0x18 ), /*
Alpha Stack size/offset = 24 */
#endif
/* 60 */ NdrFcShort( 0x3da ), /* Type
Offset=986 */

/* Return value */

/* 62 */ NdrFcShort( 0x70 ), /* Flags:  out, return,
base type, */

```

```

#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 64 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
#else
NdrFcShort( 0x1c ), /*
MIPS Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /*
PPC Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x20 ), /*
Alpha Stack size/offset = 32 */
#endif
/* 66 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure Delivery */

/* 68 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 70 */ NdrFcLong( 0x0 ), /* 0 */
/* 74 */ NdrFcShort( 0x5 ), /* 5 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 76 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
#else
NdrFcShort( 0x20 ), /*
MIPS Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x20 ), /*
PPC Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x28 ), /*
Alpha Stack size/offset = 40 */
#endif
/* 78 */ NdrFcShort( 0x0 ), /* 0 */
/* 80 */ NdrFcShort( 0x8 ), /* 8 */
/* 82 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

/* Parameter txn_in */

/* 84 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 86 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
#else

```

```

NdrFcShort( 0x8 ), /*
MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 88 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Parameter txn_out */

/* 90 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 92 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
#else
NdrFcShort( 0x18 ), /*
MIPS Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
PPC Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
Alpha Stack size/offset = 24 */
#endif
/* 94 */ NdrFcShort( 0x3da ), /* Type
Offset=986 */

/* Return value */

/* 96 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 98 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
#else
NdrFcShort( 0x1c ), /*
MIPS Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /*
PPC Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x20 ), /*
Alpha Stack size/offset = 32 */
#endif
/* 100 */ 0x8, /* FC_LONG */
0x0, /*
0 */

```

```

/* Procedure StockLevel */

/* 102 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 104 */ NdrFcLong( 0x0 ), /* 0 */
/* 108 */ NdrFcShort( 0x6 ), /* 6 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 110 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
#else
NdrFcShort( 0x20 ), /*
MIPS Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x20 ), /*
PPC Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x28 ), /*
Alpha Stack size/offset = 40 */
#endif
/* 112 */ NdrFcShort( 0x0 ), /* 0 */
/* 114 */ NdrFcShort( 0x8 ), /* 8 */
/* 116 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

/* Parameter txn_in */

/* 118 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 120 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
#else
NdrFcShort( 0x8 ), /*
MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 122 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Parameter txn_out */

/* 124 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
#ifndef _ALPHA_
#ifndef _PPC_

```



```

#if !defined(_MIPS_)
/* 126 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
#else
NdrFcShort( 0x18 ), /*
MIPS Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
PPC Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
Alpha Stack size/offset = 24 */
#endif
/* 128 */ NdrFcShort( 0x3da ), /* Type
Offset=986 */

/* Return value */

/* 130 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifdef _ALPHA_
#ifdef _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 */
#ifdef _ALPHA_
#ifdef _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
#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, */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 154 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
#else
NdrFcShort( 0x8 ), /*
MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 156 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Parameter txn_out */

/* 158 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
#ifdef _ALPHA_
#ifdef _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, */
#ifdef _ALPHA_
#ifdef _PPC_
#if !defined(_MIPS_)
/* 166 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
#else
NdrFcShort( 0x1c ), /*
MIPS Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /*
PPC Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x20 ), /*
Alpha Stack size/offset = 32 */
#endif
/* 168 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure CallSetComplete */

/* 170 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 172 */ NdrFcLong( 0x0 ), /* 0 */
/* 176 */ NdrFcShort( 0x8 ), /* 8 */
#ifdef _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, */
#ifdef _ALPHA_
/* 188 */ NdrFcShort( 0x4 ), /* x86, MIPS, PPC Stack
size/offset = 4 */
#else
NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 190 */ 0x8, /* FC_LONG */
0x0, /*
0 */

0x0

};

```

```

static const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString =
{
    0,
    {
        0 /*
        /* 2 */
        FC_UP /*
        /* 4 */ NdrFcShort( 0x3b0 ), /* Offset=
        944 (948) */
        /* 6 */
        FC_NON_ENCAPSULATED_UNION /*
        /* 8 */ 0x7, /* Corr desc: FC_USHORT
        */
        /* 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 */
FC_STRUCT /*
/* 280 */ NdrFcShort( 0x8 ), /* 8 */
/* 282 */ 0xb, /* FC_HYPER */
/* 284 */
FC_END /*
/* 286 */ NdrFcShort( 0xc ), /* Offset= 12 (298) */
/* 288 */
FC_UP /*
/* 290 */ NdrFcShort( 0x2 ), /* 2 */
/* 292 */ 0x9, /* Corr desc: FC_ULONG
*/
/* 294 */ NdrFcShort( 0xffffc ), /* -4 */
/* 296 */ 0x6, /* FC_SHORT */
/* 298 */
FC_END /*
/* 299 */
FC_CSTRUCT /*
/* 300 */ NdrFcShort( 0x8 ), /* 8 */
/* 302 */ NdrFcShort( 0xffffffff2 ), /* Offset= -
14 (288) */
/* 304 */ 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 */
0, /*
/* 320 */ 0x0, /* 0 */
0, /*
/* 322 */ 0x0, /* 0 */
0, /*
/* 324 */ 0x0, /* 0 */
70, /*
/* 326 */
0x2f, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 328 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 332 */ NdrFcShort( 0x0 ), /* 0 */
/* 334 */ NdrFcShort( 0x0 ), /* 0 */
/* 336 */ 0xc0, /* 192 */
0, /*
/* 338 */ 0x0, /* 0 */
0, /*
/* 340 */ 0x0, /* 0 */
0, /*
/* 342 */ 0x0, /* 0 */
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 */
0x8, /*
FC_LONG */
/* 448 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 450 */

```

```

0x16, /*
FC_PSTRUCT */
0x3, /*
3 */
/* 452 */ NdrFcShort( 0x8 ), /* 8 */
/* 454 */
0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */
/* 456 */
0x46, /*
FC_NO_REPEAT */
0x5c, /*
FC_PAD */
/* 458 */ NdrFcShort( 0x4 ), /* 4 */
/* 460 */ NdrFcShort( 0x4 ), /* 4 */
/* 462 */ 0x11, 0x0, /* FC_RP */
/* 464 */ NdrFcShort( 0xffffffffd4 ), /* Offset= -
44 (420) */
/* 466 */
0x5b, /*
FC_END */
0x8, /*
FC_LONG */
/* 468 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */
/* 470 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /*
3 */
/* 472 */ NdrFcShort( 0x0 ), /* 0 */
/* 474 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 476 */ NdrFcShort( 0x0 ), /* 0 */
/* 478 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 482 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 484 */ NdrFcShort( 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( 0xffffffe0 ), /* 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( 0xfffffff ), /* -1 */
/* 516 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 518 */ NdrFcShort( 0xffffff40 ), /* Offset= -
192 (326) */
/* 520 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 522 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 524 */ NdrFcShort( 0x8 ), /* 8 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 530 */ 0x8, /* FC_LONG */
0x36, /*
FC_POINTER */
/* 532 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 534 */
0x11, 0x0, /*
FC_RP */
/* 536 */ NdrFcShort( 0xffffffe0 ), /* Offset= -
32 (504) */
/* 538 */
0x1b, /*
FC_CARRAY */
0x3, /*
3 */
/* 540 */ NdrFcShort( 0x4 ), /* 4 */
/* 542 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 544 */ NdrFcShort( 0x0 ), /* 0 */
/* 546 */
0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */

```

```

/* 548 */
FC_VARIABLE_REPEAT */
0x48, /*
0x49, /*
FC_FIXED_OFFSET */
/* 550 */ NdrFcShort( 0x4 ), /* 4 */
/* 552 */ NdrFcShort( 0x0 ), /* 0 */
/* 554 */ NdrFcShort( 0x1 ), /* 1 */
/* 556 */ NdrFcShort( 0x0 ), /* 0 */
/* 558 */ NdrFcShort( 0x0 ), /* 0 */
/* 560 */ 0x12, 0x0, /* FC_UP */
/* 562 */ NdrFcShort( 0x182 ), /* Offset=
386 (948) */
/* 564 */
0x5b, /*
FC_END */
0x8, /*
FC_LONG */
/* 566 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 568 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 570 */ NdrFcShort( 0x8 ), /* 8 */
/* 572 */ NdrFcShort( 0x0 ), /* 0 */
/* 574 */ NdrFcShort( 0x6 ), /* Offset= 6 (580) */
/* 576 */ 0x8, /* FC_LONG */
0x36, /*
FC_POINTER */
/* 578 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 580 */
0x11, 0x0, /*
FC_RP */
/* 582 */ NdrFcShort( 0xffffffd4 ), /* 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( 0xffffffd8 ), /* Offset= -
40 (584) */
/* 626 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 628 */
0x12, 0x0, /*
FC_UP */
/* 630 */ NdrFcShort( 0xffffffe4 ), /* Offset= -
28 (602) */
/* 632 */
0x1b, /*
FC_CARRAY */
0x3, /*
3 */
/* 634 */ NdrFcShort( 0x4 ), /* 4 */
/* 636 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 638 */ NdrFcShort( 0x0 ), /* 0 */
/* 640 */
0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */
/* 642 */
0x48, /*
FC_VARIABLE_REPEAT */
0x49, /*
FC_FIXED_OFFSET */
/* 644 */ NdrFcShort( 0x4 ), /* 4 */
/* 646 */ NdrFcShort( 0x0 ), /* 0 */
/* 648 */ NdrFcShort( 0x1 ), /* 1 */
/* 650 */ NdrFcShort( 0x0 ), /* 0 */

```

```

/* 652 */ NdrFcShort( 0x0 ), /* 0 */
/* 654 */ 0x12, 0x0, /* FC_UP */
/* 656 */ NdrFcShort( 0xffffffffd4 ), /* 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( 0xffffffffd4 ), /* Offset= -
44 (632) */
/* 678 */
0x1d, /*
FC_SMPARRAY */
0x0, /*
0 */
/* 680 */ NdrFcShort( 0x8 ), /* 8 */
/* 682 */ 0x1, /* FC_BYTE */
0x5b, /*
FC_END */
/* 684 */
0x15, /*
FC_STRUCT */
0x3, /*
3 */
/* 686 */ NdrFcShort( 0x10 ), /* 16 */
/* 688 */ 0x8, /* FC_LONG */
0x6, /*
FC_SHORT */
/* 690 */ 0x6, /* FC_SHORT */
0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 692 */ 0x0, /* 0 */
NdrFcShort( 0xfffffffff1
), /* 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( 0xffffffffe8 ), /* 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( 0xffffffffe8 ), /* Offset= -
24 (716) */
/* 742 */
0x5b, /*
FC_END */
0x8, /*
FC_LONG */
/* 744 */ 0x8, /* FC_LONG */

```

```

0x5b, /*
FC_END */
/* 746 */
0x1b, /*
FC_CARRAY */
0x1, /*
1 */
/* 748 */ NdrFcShort( 0x2 ), /* 2 */
/* 750 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 752 */ NdrFcShort( 0x0 ), /* 0 */
/* 754 */ 0x6, /* FC_SHORT */
0x5b, /*
FC_END */
/* 756 */
0x16, /*
FC_PSTRUCT */
0x3, /*
3 */
/* 758 */ NdrFcShort( 0x8 ), /* 8 */
/* 760 */
0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */
/* 762 */
0x46, /*
FC_NO_REPEAT */
0x5c, /*
FC_PAD */
/* 764 */ NdrFcShort( 0x4 ), /* 4 */
/* 766 */ NdrFcShort( 0x4 ), /* 4 */
/* 768 */ 0x12, 0x0, /* FC_UP */
/* 770 */ NdrFcShort( 0xffffffffe8 ), /* 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 */
FC_PP /*
0x4b, /*
FC_PAD /*
/* 792 */
0x5c, /*
FC_NO_REPEAT /*
0x46, /*
0x5c, /*
FC_PAD /*
/* 794 */ NdrFcShort( 0x4 ), /* 4 */
/* 796 */ NdrFcShort( 0x4 ), /* 4 */
/* 798 */ 0x12, 0x0, /* FC_UP */
/* 800 */ NdrFcShort( 0xfffffe8 ), /* Offset= -
24 (776) */
/* 802 */
0x5b, /*
FC_END /*
0x8, /*
/* 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 */
FC_PP /*
0x4b, /*
FC_PAD /*
/* 822 */
0x5c, /*
FC_NO_REPEAT /*
0x46, /*
0x5c, /*
FC_PAD /*
/* 824 */ NdrFcShort( 0x4 ), /* 4 */
/* 826 */ NdrFcShort( 0x4 ), /* 4 */
/* 828 */ 0x12, 0x0, /* FC_UP */
/* 830 */ NdrFcShort( 0xfffffe8 ), /* Offset= -
24 (806) */
/* 832 */

```

```

0x5b, /*
FC_END /*
0x8, /*
/* 834 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END /*
/* 836 */
0x15, /*
FC_STRUCT /*
0x3, /*
3 */
/* 838 */ NdrFcShort( 0x8 ), /* 8 */
/* 840 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG /*
/* 842 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END /*
/* 844 */
0x1b, /*
FC_CARRAY /*
0x3, /*
3 */
/* 846 */ NdrFcShort( 0x8 ), /* 8 */
/* 848 */ 0x7, /* Corr desc: FC_USHORT
*/
0x0, /*
*/
/* 850 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 852 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 854 */ NdrFcShort( 0xfffffee ), /* Offset= -
18 (836) */
/* 856 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END /*
/* 858 */
0x1a, /*
FC_BOGUS_STRUCT /*
0x3, /*
3 */
/* 860 */ NdrFcShort( 0x28 ), /* 40 */
/* 862 */ NdrFcShort( 0xfffffee ), /* Offset= -
18 (844) */
/* 864 */ NdrFcShort( 0x0 ), /* Offset= 0 (864) */
/* 866 */ 0x6, /* FC_SHORT */
0x6, /*
FC_SHORT /*
/* 868 */ 0x38, /* FC_ALIGNM4 */
0x8, /*
FC_LONG /*
/* 870 */ 0x8, /* FC_LONG */
0x4c, /*
FC_EMBEDDED_COMPLEX /*
/* 872 */ 0x0, /* 0 */
NdrFcShort( 0xffffdf7
), /* Offset= -521 (352) */
0x5b, /*
FC_END /*

```

```

/* 876 */
0x12, 0x0, /*
FC_UP /*
/* 878 */ NdrFcShort( 0xfffffef6 ), /* Offset= -
266 (612) */
/* 880 */
0x12, 0x8, /*
FC_UP [simple_pointer] /*
/* 882 */ 0x1, /* FC_BYTE */
0x5c, /*
FC_PAD /*
/* 884 */
0x12, 0x8, /*
FC_UP [simple_pointer] /*
/* 886 */ 0x6, /* FC_SHORT */
0x5c, /*
FC_PAD /*
/* 888 */
0x12, 0x8, /*
FC_UP [simple_pointer] /*
/* 890 */ 0x8, /* FC_LONG */
0x5c, /*
FC_PAD /*
/* 892 */
0x12, 0x8, /*
FC_UP [simple_pointer] /*
/* 894 */ 0xa, /* FC_FLOAT */
0x5c, /*
FC_PAD /*
/* 896 */
0x12, 0x8, /*
FC_UP [simple_pointer] /*
/* 898 */ 0xc, /* FC_DOUBLE */
0x5c, /*
FC_PAD /*
/* 900 */
0x12, 0x0, /*
FC_UP /*
/* 902 */ NdrFcShort( 0xfffffd90 ), /* Offset= -
624 (278) */
/* 904 */
0x12, 0x10, /*
FC_UP [pointer_deref] /*
/* 906 */ NdrFcShort( 0xfffffd92 ), /* Offset= -
622 (284) */
/* 908 */
0x12, 0x10, /*
FC_UP [pointer_deref] /*
/* 910 */ NdrFcShort( 0xfffffda6 ), /* Offset= -
602 (308) */
/* 912 */
0x12, 0x10, /*
FC_UP [pointer_deref] /*
/* 914 */ NdrFcShort( 0xfffffdb4 ), /* Offset= -
588 (326) */
/* 916 */
0x12, 0x10, /*
FC_UP [pointer_deref] /*
/* 918 */ NdrFcShort( 0xfffffdc2 ), /* Offset= -
574 (344) */
/* 920 */
0x12, 0x10, /*
FC_UP [pointer_deref] /*

```

```

/* 922 */ NdrFcShort( 0x2 ), /* Offset= 2 (924) */
/* 924 */
                                0x12, 0x0, /*
FC_UP */
/* 926 */ NdrFcShort( 0x16 ), /* Offset= 22 (948) */
/* 928 */
                                0x15, /*
FC_STRUCT */
                                0x7, /*
7 */
/* 930 */ NdrFcShort( 0x10 ), /* 16 */
/* 932 */ 0x6, /* FC_SHORT */
                                0x1, /*
FC_BYTE */
/* 934 */ 0x1, /* FC_BYTE */
                                0x38, /*
FC_ALIGNM4 */
/* 936 */ 0x8, /* FC_LONG */
                                0x39, /*
FC_ALIGNM8 */
/* 938 */ 0xb, /* FC_HYPER */
                                0x5b, /*
FC_END */
/* 940 */
                                0x12, 0x0, /*
FC_UP */
/* 942 */ NdrFcShort( 0xffffffff2 ), /* Offset= -
14 (928) */
/* 944 */
                                0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 946 */ 0x2, /* FC_CHAR */
                                0x5c, /*
FC_PAD */
/* 948 */
                                0x1a, /*
FC_BOGUS_STRUCT */
                                0x7, /*
7 */
/* 950 */ NdrFcShort( 0x20 ), /* 32 */
/* 952 */ NdrFcShort( 0x0 ), /* 0 */
/* 954 */ NdrFcShort( 0x0 ), /* Offset= 0 (954) */
/* 956 */ 0x8, /* FC_LONG */
                                0x8, /*
FC_LONG */
/* 958 */ 0x6, /* FC_SHORT */
                                0x6, /*
FC_SHORT */
/* 960 */ 0x6, /* FC_SHORT */
                                0x6, /*
FC_SHORT */
/* 962 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
                                0x0, /*
0 */
/* 964 */ NdrFcShort( 0xfffffc42 ), /* Offset= -
958 (6) */
/* 966 */ 0x5c, /* FC_PAD */
                                0x5b, /*
FC_END */
/* 968 */ 0xb4, /* FC_USER_MARSHAL */
                                0x83, /*
131 */

```

```

/* 970 */ NdrFcShort( 0x0 ), /* 0 */
/* 972 */ NdrFcShort( 0x10 ), /* 16 */
/* 974 */ NdrFcShort( 0x0 ), /* 0 */
/* 976 */ NdrFcShort( 0xfffffc32 ), /* Offset= -
974 (2) */
/* 978 */
                                0x11, 0x4, /*
FC_RP [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 * ) &ITPCCStubVtbl,
    0
};

PCInterfaceName const
_tpcc_com_ps_InterfaceNamesList[] =
{
    "ITPCC",
    0
};

#define _tpcc_com_ps_CHECK_IID(n)
IID_GENERIC_CHECK_IID( _tpcc_com_ps, pIID,
n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID *
pIID, int * pIndex )
{
    if(!_tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }
}

```

```

return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo
=
{
    (PCInterfaceProxyVtblList *) &
    _tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) &
    _tpcc_com_ps_StubVtblList,
    (const PCInterfaceName *) &
    _tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    &_tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0 /* Filler3 */
};

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

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

/* this ALWAYS GENERATED file contains the proxy stub
code */

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:08 2001
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=12), 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 __REQD_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,{0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0
x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0
x4F,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,
    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
};

```

```

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 */
        FC_AUTO_HANDLE /* 0x33, */
        /* 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 ), /*
xpp64 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 */

```

```

    }
    };
    static const MIDL_TYPE_FORMAT_STRING
    __MIDL_TypeFormatString =
    {
        0,
        {
            0 * /
            /* 2 */
            FC_UP /*
            /* 4 */ NdrFcShort( 0x39e ), /* Offset=
            926 (930) */
            /* 6 */
            FC_NON_ENCAPSULATED_UNION /*
            /* 8 */ 0x7, /* Corr desc: FC_USHORT
            */
            /* 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 */
FC_STRUCT /*
/* 282 */ NdrFcShort( 0x8 ), /* 8 */
/* 284 */ 0xb, /* FC_HYPER */
/* 286 */
FC_END /*
/* 288 */ NdrFcShort( 0xe ), /* Offset= 14 (302) */
/* 290 */
/* 292 */ NdrFcShort( 0x2 ), /* 2 */
/* 294 */ 0x9, /* Corr desc: FC_ULONG
*/
/* 296 */ NdrFcShort( 0xffffc ), /* -4 */
/* 298 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 300 */ 0x6, /* FC_SHORT */
/* 302 */
FC_END /*
/* 304 */
/* 306 */ 0x17, /*
*/
FC_CSTRUCT /*

```

```

0x3, /*
3 */
/* 304 */ NdrFcShort( 0x8 ), /* 8 */
/* 306 */ NdrFcShort( 0xffffffff0 ), /* Offset= -
16 (290) */
/* 308 */ 0x8, /* FC_LONG */
FC_LONG /*
/* 310 */ 0x5c, /* FC_PAD */
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 */
0 /*
/* 324 */ 0x0, /* 0 */
0 /*
/* 326 */ 0x0, /* 0 */
0 /*
/* 328 */ 0x0, /* 0 */
70 /*
/* 330 */
0x2f, /*
FC_IP /*
0x5a, /*
FC_CONSTANT_IID /*
/* 332 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 336 */ NdrFcShort( 0x0 ), /* 0 */
/* 338 */ NdrFcShort( 0x0 ), /* 0 */
/* 340 */ 0xc0, /* 192 */
0 /*
/* 342 */ 0x0, /* 0 */
0 /*
/* 344 */ 0x0, /* 0 */
0 /*
/* 346 */ 0x0, /* 0 */
70 /*
/* 348 */
0x12, 0x10, /*
FC_UP [pointer_deref] /*
/* 350 */ NdrFcShort( 0x2 ), /* Offset= 2 (352) */
/* 352 */
0x12, 0x0, /*
FC_UP /*
/* 354 */ NdrFcShort( 0x1e6 ), /* Offset=
486 (840) */
/* 356 */
0x2a, /*
FC_ENCAPSULATED_UNION */

```

```

0x89, /*
137 */
/* 358 */ NdrFcShort( 0x20 ), /* 32 */
/* 360 */ NdrFcShort( 0xa ), /* 10 */
/* 362 */ NdrFcLong( 0x8 ), /* 8 */
/* 366 */ NdrFcShort( 0x50 ), /* Offset= 80 (446) */
/* 368 */ NdrFcLong( 0xd ), /* 13 */
/* 372 */ NdrFcShort( 0x70 ), /* Offset= 112 (484) */
/* 374 */ NdrFcLong( 0x9 ), /* 9 */
/* 378 */ NdrFcShort( 0x90 ), /* Offset= 144 (522) */
/* 380 */ NdrFcLong( 0xc ), /* 12 */
/* 384 */ NdrFcShort( 0xb0 ), /* Offset= 176 (560) */
/* 386 */ NdrFcLong( 0x24 ), /* 36 */
/* 390 */ NdrFcShort( 0x104 ), /* Offset=
260 (650) */
/* 392 */ NdrFcLong( 0x800d ), /* 32781 */
/* 396 */ NdrFcShort( 0x120 ), /* Offset=
288 (684) */
/* 398 */ NdrFcLong( 0x10 ), /* 16 */
/* 402 */ NdrFcShort( 0x13a ), /* Offset=
314 (716) */
/* 404 */ NdrFcLong( 0x2 ), /* 2 */
/* 408 */ NdrFcShort( 0x150 ), /* Offset=
336 (744) */
/* 410 */ NdrFcLong( 0x3 ), /* 3 */
/* 414 */ NdrFcShort( 0x166 ), /* Offset=
358 (772) */
/* 416 */ NdrFcLong( 0x14 ), /* 20 */
/* 420 */ NdrFcShort( 0x17c ), /* Offset=
380 (800) */
/* 422 */ NdrFcShort( 0xffffffff ), /* Offset= -1
(421) */
/* 424 */
0x21, /*
FC_BOGUS_ARRAY /*
0x3, /*
3 */
/* 426 */ NdrFcShort( 0x0 ), /* 0 */
/* 428 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 430 */ NdrFcShort( 0x0 ), /* 0 */
/* 432 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 434 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 438 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 440 */
0x12, 0x0, /*
FC_UP /*
/* 442 */ NdrFcShort( 0xffffffff74 ), /* Offset= -
140 (302) */
/* 444 */ 0x5c, /* FC_PAD */
FC_END /*
/* 446 */
0x1a, /*
FC_BOGUS_STRUCT /*
0x3, /*
3 */
/* 448 */ NdrFcShort( 0x10 ), /* 16 */
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */ NdrFcShort( 0x6 ), /* Offset= 6 (458) */

```

```

/* 454 */ 0x8, /* FC_LONG */
0x39, /*
FC_ALIGNM8 /*
/* 456 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END /*
/* 458 */
0x11, 0x0, /*
FC_RP /*
/* 460 */ NdrFcShort( 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 */
0x5b, /*
FC_END /*
/* 484 */
0x1a, /*
FC_BOGUS_STRUCT /*
0x3, /*
3 */
/* 486 */ NdrFcShort( 0x10 ), /* 16 */
/* 488 */ NdrFcShort( 0x0 ), /* 0 */
/* 490 */ NdrFcShort( 0x6 ), /* Offset= 6 (496) */
/* 492 */ 0x8, /* FC_LONG */
0x39, /*
FC_ALIGNM8 /*
/* 494 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END /*
/* 496 */
0x11, 0x0, /*
FC_RP /*
/* 498 */ NdrFcShort( 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( 0xfffff44 ), /* Offset= -
188 (330) */
/* 520 */ 0x5c, /* FC_PAD */
/* 522 */
FC_END */
/* 522 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 524 */ NdrFcShort( 0x10 ), /* 16 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 530 */ 0x8, /* FC_LONG */
/* 532 */ 0x36, /* FC_ALIGNM8 */
/* 534 */
FC_END */
/* 534 */
0x11, 0x0, /*
FC_RP */
/* 536 */ NdrFcShort( 0xfffffdc ), /* Offset= -
36 (500) */
/* 538 */
FC_BOGUS_ARRAY */
0x21, /*
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 */
FC_END */
/* 560 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */

```

```

/* 562 */ NdrFcShort( 0x10 ), /* 16 */
/* 564 */ NdrFcShort( 0x0 ), /* 0 */
/* 566 */ NdrFcShort( 0x6 ), /* Offset= 6 (572) */
/* 568 */ 0x8, /* FC_LONG */
/* 570 */ 0x36, /* FC_ALIGNM8 */
/* 572 */ 0x5b, /* FC_POINTER */
/* 574 */ 0x11, 0x0, /* FC_RP */
/* 576 */
FC_IP */
0x2E, /*
0x5a, /*
FC_CONSTANT_IID */
/* 578 */ NdrFcLong( 0x2f ), /* 47 */
/* 582 */ NdrFcShort( 0x0 ), /* 0 */
/* 584 */ NdrFcShort( 0x0 ), /* 0 */
/* 586 */ 0xc0, /* 192 */
0 */
/* 588 */ 0x0, /* 0 */
0 */
/* 590 */ 0x0, /* 0 */
0 */
/* 592 */ 0x0, /* 0 */
70 */
/* 594 */
FC_CARRAY */
0x1b, /*
0 */
/* 596 */ NdrFcShort( 0x1 ), /* 1 */
/* 598 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
/* 600 */ NdrFcShort( 0x4 ), /* 4 */
/* 602 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 604 */ 0x1, /* FC_BYTE */
/* 606 */
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 */
/* 616 */ 0x4c, /* FC_LONG */
/* 616 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/

```

```

0x0, /*
0 */
/* 618 */ NdrFcShort( 0xffffffff6 ), /* Offset= -
42 (576) */
/* 620 */ 0x39, /* FC_ALIGNM8 */
/* 622 */ 0x5c, /* FC_POINTER */
/* 624 */ 0x5b, /* FC_PAD */
/* 628 */
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 */
/* 650 */
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 */
/* 660 */ 0x36, /* FC_ALIGNM8 */
/* 662 */ 0x5b, /* FC_POINTER */
/* 664 */
FC_END */
/* 662 */
0x11, 0x0, /*
FC_RP */
/* 664 */ NdrFcShort( 0xfffffdc ), /* Offset= -
36 (628) */
/* 666 */
0x1d, /*
FC_SMPARRAY */
0x0, /*
0 */
/* 668 */ NdrFcShort( 0x8 ), /* 8 */

```

```

/* 670 */ 0x1,          /* FC_BYTE */
FC_END /*
/* 672 */
FC_STRUCT */
          0x15,        /*
          0x3,         /*
3 */
/* 674 */ NdrFcShort( 0x10 ), /* 16 */
/* 676 */ 0x8,         /* FC_LONG */
FC_SHORT */
/* 678 */ 0x6,        /* FC_SHORT */
FC_EMBEDDED_COMPLEX /*
/* 680 */ 0x0,        /* 0 */
NdrFcShort( 0xffffffffl
),          /* Offset= -15 (666) */
FC_END /*
/* 684 */
FC_BOGUS_STRUCT */
          0x1a,        /*
          0x3,         /*
3 */
/* 686 */ NdrFcShort( 0x20 ), /* 32 */
/* 688 */ NdrFcShort( 0x0 ), /* 0 */
/* 690 */ NdrFcShort( 0xa ), /* Offset= 10 (700) */
/* 692 */ 0x8,        /* FC_LONG */
FC_ALIGNM8 /*
/* 694 */ 0x36,       /* FC_POINTER */
FC_EMBEDDED_COMPLEX /*
/* 696 */ 0x0,        /* 0 */
NdrFcShort( 0xffffffffe7
),          /* Offset= -25 (672) */
FC_END /*
/* 700 */
FC_RP /*
/* 702 */ NdrFcShort( 0xffffffff10 ), /* Offset= -
240 (462) */
/* 704 */
FC_CARRY /*
          0x1b,        /*
          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 */
FC_END /*
/* 716 */
FC_BOGUS_STRUCT */
          0x1a,        /*

```

```

          0x3,         /*
3 */
/* 718 */ NdrFcShort( 0x10 ), /* 16 */
/* 720 */ NdrFcShort( 0x0 ), /* 0 */
/* 722 */ NdrFcShort( 0x6 ), /* Offset= 6 (728) */
/* 724 */ 0x8,        /* FC_LONG */
FC_ALIGNM8 /*
/* 726 */ 0x36,       /* FC_POINTER */
FC_END /*
/* 728 */
FC_UP /*
/* 730 */ NdrFcShort( 0xffffffffe6 ), /* Offset= -
26 (704) */
/* 732 */
FC_CARRY /*
          0x1b,        /*
          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 */
FC_END /*
/* 744 */
FC_BOGUS_STRUCT */
          0x1a,        /*
          0x3,         /*
3 */
/* 746 */ NdrFcShort( 0x10 ), /* 16 */
/* 748 */ NdrFcShort( 0x0 ), /* 0 */
/* 750 */ NdrFcShort( 0x6 ), /* Offset= 6 (756) */
/* 752 */ 0x8,        /* FC_LONG */
FC_ALIGNM8 /*
/* 754 */ 0x36,       /* FC_POINTER */
FC_END /*
/* 756 */
FC_UP /*
/* 758 */ NdrFcShort( 0xffffffffe6 ), /* Offset= -
26 (732) */
/* 760 */
FC_CARRY /*
          0x1b,        /*
          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 */
FC_END /*
/* 772 */
FC_BOGUS_STRUCT */
          0x1a,        /*
          0x3,         /*
3 */
/* 774 */ NdrFcShort( 0x10 ), /* 16 */
/* 776 */ NdrFcShort( 0x0 ), /* 0 */
/* 778 */ NdrFcShort( 0x6 ), /* Offset= 6 (784) */
/* 780 */ 0x8,        /* FC_LONG */
FC_ALIGNM8 /*
/* 782 */ 0x36,       /* FC_POINTER */
FC_END /*
/* 784 */
FC_UP /*
/* 786 */ NdrFcShort( 0xffffffffe6 ), /* Offset= -
26 (760) */
/* 788 */
FC_CARRY /*
          0x1b,        /*
          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 */
FC_END /*
/* 800 */
FC_BOGUS_STRUCT */
          0x1a,        /*
          0x3,         /*
3 */
/* 802 */ NdrFcShort( 0x10 ), /* 16 */
/* 804 */ NdrFcShort( 0x0 ), /* 0 */
/* 806 */ NdrFcShort( 0x6 ), /* Offset= 6 (812) */
/* 808 */ 0x8,        /* FC_LONG */
FC_ALIGNM8 /*
/* 810 */ 0x36,       /* FC_POINTER */
FC_END /*
/* 812 */
FC_UP /*
/* 814 */ NdrFcShort( 0xffffffffe6 ), /* Offset= -
26 (788) */
/* 816 */
FC_STRUCT */
          0x15,        /*

```

```

0x3,          /*
3 */
/* 818 */ NdrFcShort( 0x8 ), /* 8 */
/* 820 */ 0x8, /* FC_LONG */
/* FC_LONG */
/* 822 */ 0x5c, /* FC_PAD */
/* FC_END */
/* 824 */
/* FC_CARRAY */
/* FC_CARRAY */
/* FC_CARRAY */
/* FC_CARRAY */
3 */
/* 826 */ NdrFcShort( 0x8 ), /* 8 */
/* 828 */ 0x7, /* Corr desc: FC_USHORT */
/* FC_CARRAY */
/* FC_CARRAY */
/* FC_CARRAY */
/* 830 */ NdrFcShort( 0xffc8 ), /* -56 */
/* 832 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* FC_CARRAY */
/* 834 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* FC_CARRAY */
/* FC_CARRAY */
0 */
/* 836 */ NdrFcShort( 0xfffffec ), /* Offset= -
20 (816) */
/* 838 */ 0x5c, /* FC_PAD */
/* FC_END */
/* 840 */
/* FC_BOGUS_STRUCT */
/* FC_BOGUS_STRUCT */
3 */
/* 842 */ NdrFcShort( 0x38 ), /* 56 */
/* 844 */ NdrFcShort( 0xfffffec ), /* Offset= -
20 (824) */
/* 846 */ NdrFcShort( 0x0 ), /* Offset= 0 (846) */
/* 848 */ 0x6, /* FC_SHORT */
/* FC_SHORT */
/* 850 */ 0x38, /* FC_ALIGNM4 */
/* FC_LONG */
/* 852 */ 0x8, /* FC_LONG */
/* FC_EMBEDDED_COMPLEX */
/* 854 */ 0x4, /* 4 */
/* FC_EMBEDDED_COMPLEX */
/* FC_EMBEDDED_COMPLEX */
/* FC_END */
/* 858 */
/* FC_UP */
/* 860 */ NdrFcShort( 0xfffff02 ), /* Offset= -
254 (606) */
/* 862 */
/* FC_UP [simple_pointer] */
/* 864 */ 0x1, /* FC_BYTE */

```

```

FC_PAD */
/* 866 */
FC_UP [simple_pointer] */
/* 868 */ 0x6, /* FC_SHORT */
FC_PAD */
/* 870 */
FC_UP [simple_pointer] */
/* 872 */ 0x8, /* FC_LONG */
FC_PAD */
/* 874 */
FC_UP [simple_pointer] */
/* 876 */ 0xa, /* FC_FLOAT */
FC_PAD */
/* 878 */
FC_UP [simple_pointer] */
/* 880 */ 0xc, /* FC_DOUBLE */
FC_PAD */
/* 882 */
FC_UP */
/* 884 */ NdrFcShort( 0xffffda4 ), /* Offset= -
604 (280) */
/* 886 */
FC_UP [pointer_deref] */
/* 888 */ NdrFcShort( 0xffffda6 ), /* Offset= -
602 (286) */
/* 890 */
FC_UP [pointer_deref] */
/* 892 */ NdrFcShort( 0xffffdbc ), /* Offset= -
580 (312) */
/* 894 */
FC_UP [pointer_deref] */
/* 896 */ NdrFcShort( 0xffffdca ), /* Offset= -
566 (330) */
/* 898 */
FC_UP [pointer_deref] */
/* 900 */ NdrFcShort( 0xffffdd8 ), /* Offset= -
552 (348) */
/* 902 */
FC_UP [pointer_deref] */
/* 904 */ NdrFcShort( 0x2 ), /* Offset= 2 (906) */
/* 906 */
FC_UP */
/* 908 */ NdrFcShort( 0x16 ), /* Offset= 22 (930) */
/* 910 */
FC_STRUCT */

```

```

0x7,          /*
7 */
/* 912 */ NdrFcShort( 0x10 ), /* 16 */
/* 914 */ 0x6, /* FC_SHORT */
/* FC_BYTE */
/* 916 */ 0x1, /* FC_BYTE */
/* FC_ALIGNM4 */
/* 918 */ 0x8, /* FC_LONG */
/* FC_ALIGNM8 */
/* 920 */ 0xb, /* FC_HYPER */
/* FC_END */
/* 922 */
/* FC_UP */
/* 924 */ NdrFcShort( 0xfffff2 ), /* Offset= -
14 (910) */
/* 926 */
/* FC_UP [simple_pointer] */
/* 928 */ 0x2, /* FC_CHAR */
/* FC_PAD */
/* 930 */
/* FC_BOGUS_STRUCT */
/* FC_BOGUS_STRUCT */
7 */
/* 932 */ NdrFcShort( 0x20 ), /* 32 */
/* 934 */ NdrFcShort( 0x0 ), /* 0 */
/* 936 */ NdrFcShort( 0x0 ), /* Offset= 0 (936) */
/* 938 */ 0x8, /* FC_LONG */
/* FC_LONG */
/* 940 */ 0x6, /* FC_SHORT */
/* FC_SHORT */
/* 942 */ 0x6, /* FC_SHORT */
/* FC_SHORT */
/* 944 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* FC_CARRAY */
0 */
/* 946 */ NdrFcShort( 0xffffc54 ), /* Offset= -
940 (6) */
/* 948 */ 0x5c, /* FC_PAD */
/* FC_END */
/* 950 */ 0xb4, /* FC_USER_MARSHAL */
/* FC_CARRAY */
131 */
/* 952 */ NdrFcShort( 0x0 ), /* 0 */
/* 954 */ NdrFcShort( 0x18 ), /* 24 */
/* 956 */ NdrFcShort( 0x0 ), /* 0 */
/* 958 */ NdrFcShort( 0xffffc44 ), /* Offset= -
956 (2) */
/* 960 */
/* FC_UP [simple_pointer] */

```

```

/* 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_ps_p.c


---


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

/* this ALWAYS GENERATED file contains the proxy stub
code */

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:08 2001
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=i2), 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)
#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,{0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xc0,0x00,0x00,0x00,0
x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0
x4F,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,
    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,
    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 */
        FC_AUTO_HANDLE *          0x33,          /*
        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, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 30 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
#else
                                NdrFcShort( 0x1c ), /*
MIPS Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x1c ), /*
PPC Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x20 ), /*
Alpha Stack size/offset = 32 */
#endif
/* 32 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure Payment */

/* 34 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 36 */ NdrFcLong( 0x0 ), /* 0 */
/* 40 */ NdrFcShort( 0x4 ), /* 4 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 42 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
#else
                                NdrFcShort( 0x20 ), /*
MIPS Stack size/offset = 32 */
#endif
#else
                                NdrFcShort( 0x20 ), /*
PPC Stack size/offset = 32 */
#endif
#else
                                NdrFcShort( 0x28 ), /*
Alpha Stack size/offset = 40 */
#endif
/* 44 */ NdrFcShort( 0x0 ), /* 0 */
/* 46 */ NdrFcShort( 0x8 ), /* 8 */
/* 48 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

/* Parameter txn_in */

```

```

/* 50 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 52 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
#else
                                NdrFcShort( 0x8 ), /*
MIPS Stack size/offset = 8 */
#endif
#else
                                NdrFcShort( 0x8 ), /*
PPC Stack size/offset = 8 */
#endif
#else
                                NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 54 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Parameter txn_out */

/* 56 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 58 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
#else
                                NdrFcShort( 0x18 ), /*
MIPS Stack size/offset = 24 */
#endif
#else
                                NdrFcShort( 0x18 ), /*
PPC Stack size/offset = 24 */
#endif
#else
                                NdrFcShort( 0x18 ), /*
Alpha Stack size/offset = 24 */
#endif
/* 60 */ NdrFcShort( 0x3da ), /* Type
Offset=986 */

/* Return value */

/* 62 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 64 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
#else
                                NdrFcShort( 0x1c ), /*
MIPS Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x1c ), /*
Alpha Stack size/offset = 28 */
#endif

```

```

                                NdrFcShort( 0x1c ), /*
PPC Stack size/offset = 28 */
#endif
#else
                                NdrFcShort( 0x20 ), /*
Alpha Stack size/offset = 32 */
#endif
/* 66 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure Delivery */

/* 68 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 70 */ NdrFcLong( 0x0 ), /* 0 */
/* 74 */ NdrFcShort( 0x5 ), /* 5 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 76 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
#else
                                NdrFcShort( 0x20 ), /*
MIPS Stack size/offset = 32 */
#endif
#else
                                NdrFcShort( 0x20 ), /*
PPC Stack size/offset = 32 */
#endif
#else
                                NdrFcShort( 0x28 ), /*
Alpha Stack size/offset = 40 */
#endif
/* 78 */ NdrFcShort( 0x0 ), /* 0 */
/* 80 */ NdrFcShort( 0x8 ), /* 8 */
/* 82 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

/* Parameter txn_in */

/* 84 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 86 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
#else
                                NdrFcShort( 0x8 ), /*
MIPS Stack size/offset = 8 */
#endif
#else
                                NdrFcShort( 0x8 ), /*
PPC Stack size/offset = 8 */
#endif
#else
                                NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif

```

```

/* 88 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Parameter txn_out */

/* 90 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 92 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
#else
NdrFcShort( 0x18 ), /*
MIPS Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
PPC Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
Alpha Stack size/offset = 24 */
#endif
/* 94 */ NdrFcShort( 0x3da ), /* Type
Offset=986 */

/* Return value */

/* 96 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 98 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
#else
NdrFcShort( 0x1c ), /*
MIPS Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /*
PPC Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x20 ), /*
Alpha Stack size/offset = 32 */
#endif
/* 100 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure StockLevel */

/* 102 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 104 */ NdrFcLong( 0x0 ), /* 0 */
/* 108 */ NdrFcShort( 0x6 ), /* 6 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)

```

```

/* 110 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
#else
NdrFcShort( 0x20 ), /*
MIPS Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x20 ), /*
PPC Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x28 ), /*
Alpha Stack size/offset = 40 */
#endif
/* 112 */ NdrFcShort( 0x0 ), /* 0 */
/* 114 */ NdrFcShort( 0x8 ), /* 8 */
/* 116 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

/* Parameter txn_in */

/* 118 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 120 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
#else
NdrFcShort( 0x8 ), /*
MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 122 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Parameter txn_out */

/* 124 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 126 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
#else
NdrFcShort( 0x18 ), /*
MIPS Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
PPC Stack size/offset = 24 */
#endif

```

```

#else
NdrFcShort( 0x18 ), /*
Alpha Stack size/offset = 24 */
#endif
/* 128 */ NdrFcShort( 0x3da ), /* Type
Offset=986 */

/* Return value */

/* 130 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 132 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
#else
NdrFcShort( 0x1c ), /*
MIPS Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /*
PPC Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x1c ), /*
Alpha Stack size/offset = 32 */
#endif
/* 134 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure OrderStatus */

/* 136 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 138 */ NdrFcLong( 0x0 ), /* 0 */
/* 142 */ NdrFcShort( 0x7 ), /* 7 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 144 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
#else
NdrFcShort( 0x20 ), /*
MIPS Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x20 ), /*
PPC Stack size/offset = 32 */
#endif
#else
NdrFcShort( 0x28 ), /*
Alpha Stack size/offset = 40 */
#endif
/* 146 */ NdrFcShort( 0x0 ), /* 0 */
/* 148 */ NdrFcShort( 0x8 ), /* 8 */
/* 150 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

```

```

/* Parameter txn_in */

/* 152 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 154 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
#else
NdrFcShort( 0x8 ), /*
MIPS Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
PPC Stack size/offset = 8 */
#endif
#else
NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 156 */ NdrFcShort( 0x3c8 ), /* Type
Offset=968 */

/* Parameter txn_out */

/* 158 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 160 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
#else
NdrFcShort( 0x18 ), /*
MIPS Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
PPC Stack size/offset = 24 */
#endif
#else
NdrFcShort( 0x18 ), /*
Alpha Stack size/offset = 24 */
#endif
/* 162 */ NdrFcShort( 0x3da ), /* Type
Offset=986 */

/* Return value */

/* 164 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 166 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
#else
NdrFcShort( 0x1c ), /*
MIPS Stack size/offset = 28 */
#endif
#else

```

```

NdrFcShort( 0x1c ), /*
PPC Stack size/offset = 28 */
#endif
#else
NdrFcShort( 0x20 ), /*
Alpha Stack size/offset = 32 */
#endif
/* 168 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure CallSetComplete */

/* 170 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 172 */ NdrFcLong( 0x0 ), /* 0 */
/* 176 */ NdrFcShort( 0x8 ), /* 8 */
#ifndef _ALPHA_
/* 178 */ NdrFcShort( 0x8 ), /* x86, MIPS, PPC Stack
size/offset = 8 */
#else
NdrFcShort( 0x10 ), /*
Alpha Stack size/offset = 16 */
#endif
/* 180 */ NdrFcShort( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x8 ), /* 8 */
/* 184 */ 0x4, /* Oi2 Flags: has
return, */
0x1, /*
1 */

/* Return value */

/* 186 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
#ifndef _ALPHA_
/* 188 */ NdrFcShort( 0x4 ), /* x86, MIPS, PPC Stack
size/offset = 4 */
#else
NdrFcShort( 0x8 ), /*
Alpha Stack size/offset = 8 */
#endif
/* 190 */ 0x8, /* FC_LONG */
0x0, /*
0 */
0x0

};

static const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString =
{
0,
{
NdrFcShort( 0x0 ), /*
0 */
/* 2 */
0x12, 0x0, /*
FC_UP */
/* 4 */ NdrFcShort( 0x3b0 ), /* Offset=
944 (948) */

```

```

/* 6 */
FC_NON_ENCAPSULATED_UNION /*
0x2b, /*
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 */
FC_STRUCT /*
0x15, /*
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 */
FC_IP /*
0x2f, /*
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 */
FC_CARRAY */ 0x1b, /*
3 */ 0x3, /*
/* 422 */ NdrFcShort( 0x4 ), /* 4 */
/* 424 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
/* 426 */ NdrFcShort( 0x0 ), /* 0 */
/* 428 */
FC_PP */ 0x4b, /*
FC_PAD */ 0x5c, /*
/* 430 */
FC_VARIABLE_REPEAT */ 0x48, /*
FC_FIXED_OFFSET */ 0x49, /*
/* 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 */
FC_END */ 0x5b, /*
FC_LONG */ 0x8, /*
/* 448 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */ 0x16, /*
/* 450 */
FC_PSTRUCT */ 0x3, /*
3 */
/* 452 */ NdrFcShort( 0x8 ), /* 8 */
/* 454 */
FC_PP */ 0x4b, /*
FC_PAD */ 0x5c, /*
/* 456 */
FC_NO_REPEAT */ 0x46, /*

```

```

FC_PAD */ 0x5c, /*
/* 458 */ NdrFcShort( 0x4 ), /* 4 */
/* 460 */ NdrFcShort( 0x4 ), /* 4 */
/* 462 */ 0x11, 0x0, /* FC_RP */
/* 464 */ NdrFcShort( 0xffffffff4 ), /* Offset= -
44 (420) */
/* 466 */
FC_END */ 0x5b, /*
FC_LONG */ 0x8, /*
/* 468 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */ 0x21, /*
/* 470 */
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 */ 0x1a, /*
/* 488 */
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 */ /* FC_PAD */
/* 498 */ 0x5c, 0x5b, /*
FC_END */ 0x11, 0x0, /*
/* 500 */
FC_RP */
/* 502 */ NdrFcShort( 0xffffffffe0 ), /* Offset= -
32 (470) */
/* 504 */
FC_BOGUS_ARRAY */ 0x21, /*
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 */ 0x1a, /*
/* 522 */
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 */ /* FC_PAD */
/* 532 */ 0x5c, 0x5b, /*
FC_END */ 0x11, 0x0, /*
/* 534 */
FC_RP */
/* 536 */ NdrFcShort( 0xffffffffe0 ), /* Offset= -
32 (504) */
/* 538 */
FC_CARRAY */ 0x1b, /*
3 */ 0x3, /*
/* 540 */ NdrFcShort( 0x4 ), /* 4 */
/* 542 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
/* 544 */ NdrFcShort( 0x0 ), /* 0 */
/* 546 */
FC_PP */ 0x4b, /*
FC_PAD */ 0x5c, /*
/* 548 */
FC_VARIABLE_REPEAT */ 0x48, /*
FC_FIXED_OFFSET */ 0x49, /*
/* 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 */
FC_END */
0x5b, /*
/*
FC_LONG */
/* 566 */ 0x5c, /* FC_PAD */
FC_END */
/* 568 */
FC_BOGUS_STRUCT */
0x1a, /*
0x3, /*
3 */
/* 570 */ NdrFcShort( 0x8 ), /* 8 */
/* 572 */ NdrFcShort( 0x0 ), /* 0 */
/* 574 */ NdrFcShort( 0x6 ), /* Offset= 6 (580) */
/* 576 */ 0x8, /* FC_LONG */
FC_POINTER */
/* 578 */ 0x5c, /* FC_PAD */
FC_END */
/* 580 */
0x11, 0x0, /*
FC_RP */
/* 582 */ NdrFcShort( 0xfffffd4 ), /* Offset= -
44 (538) */
/* 584 */
0x2E, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 586 */ NdrFcLong( 0x2f ), /* 47 */
/* 590 */ NdrFcShort( 0x0 ), /* 0 */
/* 592 */ NdrFcShort( 0x0 ), /* 0 */
/* 594 */ 0xc0, /* 192 */
0 */
/* 596 */ 0x0, /* 0 */
0 */
/* 598 */ 0x0, /* 0 */
0 */
/* 600 */ 0x0, /* 0 */
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 */
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 */
FC_LONG */
/* 622 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 624 */ NdrFcShort( 0xfffffd8 ), /* Offset= -
40 (584) */
/* 626 */ 0x36, /* FC_POINTER */
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 */
FC_POINTER */
/* 672 */ 0x5c, /* FC_PAD */
FC_END */
/* 674 */
0x11, 0x0, /*
FC_RP */
/* 676 */ NdrFcShort( 0xfffffd4 ), /* Offset= -
44 (632) */
/* 678 */
0x1d, /*
FC_SMFARRAY */
0x0, /*
0 */
/* 680 */ NdrFcShort( 0x8 ), /* 8 */
/* 682 */ 0x1, /* FC_BYTE */
FC_END */
/* 684 */
0x15, /*
FC_STRUCT */
0x3, /*
3 */
/* 686 */ NdrFcShort( 0x10 ), /* 16 */
/* 688 */ 0x8, /* FC_LONG */
FC_SHORT */
/* 690 */ 0x6, /* FC_SHORT */
FC_EMBEDDED_COMPLEX */
/* 692 */ 0x0, /* 0 */
NdrFcShort( 0xfffff1
), /* Offset= -15 (678) */
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 */
FC_POINTER */
/* 706 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 708 */ NdrFcShort( 0xfffffe8 ), /* Offset= -
24 (684) */
/* 710 */ 0x5c, /* FC_PAD */

```

```

                                0x5b,          /*
FC_END */
/* 712 */
                                0x11, 0x0,      /*
FC_RP */
/* 714 */ NdrFcShort( 0xfffff0c ), /* Offset= -
244 (470) */
/* 716 */
                                0x1b,          /*
FC_CARRAY */
                                0x0,          /*
0 */
/* 718 */ NdrFcShort( 0x1 ), /* 1 */
/* 720 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
                                0x0,          /*
*/
/* 722 */ NdrFcShort( 0x0 ), /* 0 */
/* 724 */ 0x1, /* FC_BYTE */
                                0x5b,          /*
FC_END */
/* 726 */
                                0x16,          /*
FC_PSTRUCT */
                                0x3,          /*
3 */
/* 728 */ NdrFcShort( 0x8 ), /* 8 */
/* 730 */
                                0x4b,          /*
FC_PP */
                                0x5c,          /*
FC_PAD */
/* 732 */
                                0x46,          /*
FC_NO_REPEAT */
                                0x5c,          /*
FC_PAD */
/* 734 */ NdrFcShort( 0x4 ), /* 4 */
/* 736 */ NdrFcShort( 0x4 ), /* 4 */
/* 738 */ 0x12, 0x0, /* FC_UP */
/* 740 */ NdrFcShort( 0xffffffe8 ), /* Offset= -
24 (716) */
/* 742 */
                                0x5b,          /*
FC_END */
                                0x8,          /*
FC_LONG */
/* 744 */ 0x8, /* FC_LONG */
                                0x5b,          /*
FC_END */
/* 746 */
                                0x1b,          /*
FC_CARRAY */
                                0x1,          /*
1 */
/* 748 */ NdrFcShort( 0x2 ), /* 2 */
/* 750 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
                                0x0,          /*
*/
/* 752 */ NdrFcShort( 0x0 ), /* 0 */
/* 754 */ 0x6, /* FC_SHORT */

```

```

                                0x5b,          /*
FC_END */
/* 756 */
                                0x16,          /*
FC_PSTRUCT */
                                0x3,          /*
3 */
/* 758 */ NdrFcShort( 0x8 ), /* 8 */
/* 760 */
                                0x4b,          /*
FC_PP */
                                0x5c,          /*
FC_PAD */
/* 762 */
                                0x46,          /*
FC_NO_REPEAT */
                                0x5c,          /*
FC_PAD */
/* 764 */ NdrFcShort( 0x4 ), /* 4 */
/* 766 */ NdrFcShort( 0x4 ), /* 4 */
/* 768 */ 0x12, 0x0, /* FC_UP */
/* 770 */ NdrFcShort( 0xffffffe8 ), /* Offset= -
24 (746) */
/* 772 */
                                0x5b,          /*
FC_END */
                                0x8,          /*
FC_LONG */
/* 774 */ 0x8, /* FC_LONG */
                                0x5b,          /*
FC_END */
/* 776 */
                                0x1b,          /*
FC_CARRAY */
                                0x3,          /*
3 */
/* 778 */ NdrFcShort( 0x4 ), /* 4 */
/* 780 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
                                0x0,          /*
*/
/* 782 */ NdrFcShort( 0x0 ), /* 0 */
/* 784 */ 0x8, /* FC_LONG */
                                0x5b,          /*
FC_END */
/* 786 */
                                0x16,          /*
FC_PSTRUCT */
                                0x3,          /*
3 */
/* 788 */ NdrFcShort( 0x8 ), /* 8 */
/* 790 */
                                0x4b,          /*
FC_PP */
                                0x5c,          /*
FC_PAD */
/* 792 */
                                0x46,          /*
FC_NO_REPEAT */
                                0x5c,          /*
FC_PAD */
/* 794 */ NdrFcShort( 0x4 ), /* 4 */

```

```

/* 796 */ NdrFcShort( 0x4 ), /* 4 */
/* 798 */ 0x12, 0x0, /* FC_UP */
/* 800 */ NdrFcShort( 0xffffffe8 ), /* Offset= -
24 (776) */
/* 802 */
                                0x5b,          /*
FC_END */
                                0x8,          /*
FC_LONG */
/* 804 */ 0x8, /* FC_LONG */
                                0x5b,          /*
FC_END */
/* 806 */
                                0x1b,          /*
FC_CARRAY */
                                0x7,          /*
7 */
/* 808 */ NdrFcShort( 0x8 ), /* 8 */
/* 810 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
                                0x0,          /*
*/
/* 812 */ NdrFcShort( 0x0 ), /* 0 */
/* 814 */ 0xb, /* FC_HYPER */
                                0x5b,          /*
FC_END */
/* 816 */
                                0x16,          /*
FC_PSTRUCT */
                                0x3,          /*
3 */
/* 818 */ NdrFcShort( 0x8 ), /* 8 */
/* 820 */
                                0x4b,          /*
FC_PP */
                                0x5c,          /*
FC_PAD */
/* 822 */
                                0x46,          /*
FC_NO_REPEAT */
                                0x5c,          /*
FC_PAD */
/* 824 */ NdrFcShort( 0x4 ), /* 4 */
/* 826 */ NdrFcShort( 0x4 ), /* 4 */
/* 828 */ 0x12, 0x0, /* FC_UP */
/* 830 */ NdrFcShort( 0xffffffe8 ), /* Offset= -
24 (806) */
/* 832 */
                                0x5b,          /*
FC_END */
                                0x8,          /*
FC_LONG */
/* 834 */ 0x8, /* FC_LONG */
                                0x5b,          /*
FC_END */
/* 836 */
                                0x15,          /*
FC_STRUCT */
                                0x3,          /*
3 */
/* 838 */ NdrFcShort( 0x8 ), /* 8 */

```



```

/* 840 */ 0x8,          /* FC_LONG */
FC_LONG */
/* 842 */ 0x5c,        /* FC_PAD */
FC_END */
/* 844 */
FC_CARRAY */
3 */
/* 846 */ NdrFcShort( 0x8 ), /* 8 */
/* 848 */ 0x7,          /* Corr desc: FC_USHORT */
/*
/* 850 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 852 */ 0x4c,          /* FC_EMBEDDED_COMPLEX */
/*
0 */
/* 854 */ NdrFcShort( 0xfffffee ), /* Offset= -
18 (836) */
/* 856 */ 0x5c,          /* FC_PAD */
FC_END */
/* 858 */
FC_BOGUS_STRUCT */
3 */
/* 860 */ NdrFcShort( 0x28 ), /* 40 */
/* 862 */ NdrFcShort( 0xfffffee ), /* Offset= -
18 (844) */
/* 864 */ NdrFcShort( 0x0 ), /* Offset= 0 (864) */
/* 866 */ 0x6,          /* FC_SHORT */
FC_SHORT */
/* 868 */ 0x38,          /* FC_ALIGNM4 */
FC_LONG */
/* 870 */ 0x8,          /* FC_LONG */
FC_EMBEDDED_COMPLEX */
/* 872 */ 0x0,          /* 0 */
NdrFcShort( 0xffffdf7
), /* Offset= -521 (352) */
FC_END */
/* 876 */
FC_UP */
/* 878 */ NdrFcShort( 0xffffef6 ), /* Offset= -
266 (612) */
/* 880 */
FC_UP [simple_pointer] */
/* 882 */ 0x1,          /* FC_BYTE */
FC_PAD */
/* 884 */
FC_UP [simple_pointer] */

```

```

/* 886 */ 0x6,          /* FC_SHORT */
FC_PAD */
/* 888 */
FC_UP [simple_pointer] */
/* 890 */ 0x8,          /* FC_LONG */
FC_PAD */
/* 892 */
FC_UP [simple_pointer] */
/* 894 */ 0xa,          /* FC_FLOAT */
FC_PAD */
/* 896 */
FC_UP [simple_pointer] */
/* 898 */ 0xc,          /* FC_DOUBLE */
FC_PAD */
/* 900 */
FC_UP */
/* 902 */ NdrFcShort( 0xfffffd90 ), /* Offset= -
624 (278) */
/* 904 */
FC_UP [pointer_deref] */
/* 906 */ NdrFcShort( 0xfffffd92 ), /* Offset= -
622 (284) */
/* 908 */
FC_UP [pointer_deref] */
/* 910 */ NdrFcShort( 0xfffffda6 ), /* Offset= -
602 (308) */
/* 912 */
FC_UP [pointer_deref] */
/* 914 */ NdrFcShort( 0xfffffdb4 ), /* Offset= -
588 (326) */
/* 916 */
FC_UP [pointer_deref] */
/* 918 */ NdrFcShort( 0xfffffdc2 ), /* Offset= -
574 (344) */
/* 920 */
FC_UP [pointer_deref] */
/* 922 */ NdrFcShort( 0x2 ), /* Offset= 2 (924) */
/* 924 */
FC_UP */
/* 926 */ NdrFcShort( 0x16 ), /* Offset= 22 (948) */
/* 928 */
FC_STRUCT */
7 */
/* 930 */ NdrFcShort( 0x10 ), /* 16 */
/* 932 */ 0x6,          /* FC_SHORT */
FC_BYTE */

```

```

/* 934 */ 0x1,          /* FC_BYTE */
FC_ALIGNM4 */
/* 936 */ 0x8,          /* FC_LONG */
FC_ALIGNM8 */
/* 938 */ 0xb,          /* FC_HYPER */
FC_END */
/* 940 */
FC_UP */
/* 942 */ NdrFcShort( 0xffffffff2 ), /* Offset= -
14 (928) */
/* 944 */
FC_UP [simple_pointer] */
/* 946 */ 0x2,          /* FC_CHAR */
FC_PAD */
/* 948 */
FC_BOGUS_STRUCT */
7 */
/* 950 */ NdrFcShort( 0x20 ), /* 32 */
/* 952 */ NdrFcShort( 0x0 ), /* 0 */
/* 954 */ NdrFcShort( 0x0 ), /* Offset= 0 (954) */
/* 956 */ 0x8,          /* FC_LONG */
FC_LONG */
/* 958 */ 0x6,          /* FC_SHORT */
FC_SHORT */
/* 960 */ 0x6,          /* FC_SHORT */
FC_SHORT */
/* 962 */ 0x4c,          /* FC_EMBEDDED_COMPLEX */
0 */
/* 964 */ NdrFcShort( 0xfffffc42 ), /* Offset= -
958 (6) */
/* 966 */ 0x5c,          /* FC_PAD */
FC_END */
/* 968 */ 0xb4,          /* FC_USER_MARSHAL */
131 */
/* 970 */ NdrFcShort( 0x0 ), /* 0 */
/* 972 */ NdrFcShort( 0x10 ), /* 16 */
/* 974 */ NdrFcShort( 0x0 ), /* 0 */
/* 976 */ NdrFcShort( 0xfffffc32 ), /* Offset= -
974 (2) */
/* 978 */
FC_RP [allocated_on_stack] */
/* 980 */ NdrFcShort( 0x6 ), /* Offset= 6 (986) */
/* 982 */
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 * ) &ITPCCStubVtbl,
    0
};

PCInterfaceName const
_tpcc_com_ps_InterfaceNamesList[] =
{
    "ITPCC",
    0
};

#define _tpcc_com_ps_CHECK_IID(n)
IID_GENERIC_CHECK_IID( _tpcc_com_ps, pIID,
n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID *
pIID, int * pIndex )
{
    if(!_tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }

    return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo
=
{
    (PCInterfaceProxyVtblList *) &
_tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) &
_tpcc_com_ps_StubVtblList,
    (const PCInterfaceName * ) &
_tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    &_tpcc_com_ps_IID_Lookup,

```

```

1,
2,
0, /* table of [async_uuid] interfaces */
0, /* Filler1 */
0, /* Filler2 */
0 /* Filler3 */
};

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

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

/* this ALWAYS GENERATED file contains the proxy stub
code */

/* File created by MIDL compiler version 5.03.0280
*/
/* at Thu Dec 13 23:13:08 2001
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf (OptLev=i2), 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_HEADER( )

#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,{0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xc0,0x00,0x00,0x00,0
x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0
x4F,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,
    __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 */

        FC_AUTO_HANDLE /*
                                0x33,
                                */
        /*
        Old Flags: object, Oi2 */
        /* 2 */ NdrFcLong( 0x0 ), /* 0 */
        /* 6 */ NdrFcShort( 0x3 ), /* 3 */
        #ifndef _ALPHA_
        /* 8 */ NdrFcShort( 0x38 ), /* ia64 Stack
        size/offset = 56 */
        #else
        NdrFcShort( 0x30 ), /*
        axp64 Stack size/offset = 48 */
        #endif
        /* 10 */ NdrFcShort( 0x0 ), /* 0 */
        /* 12 */ NdrFcShort( 0x8 ), /* 8 */
        /* 14 */ 0x47, /* Oi2 Flags: srv must
        size, clt must size, has return, has ext, */
        3 /*
        */
        /* 16 */ 0xa, /* 10 */
        /*
        Ext Flags: new corr desc, clt corr check, srv corr
        check, */
        /* 18 */ NdrFcShort( 0x20 ), /* 32 */
        /* 20 */ NdrFcShort( 0x20 ), /* 32 */
        /* 22 */ NdrFcShort( 0x0 ), /* 0 */
        /* 24 */ NdrFcShort( 0x0 ), /* 0 */

        /* Parameter txn_in */

        /* 26 */ NdrFcShort( 0x8b ), /* Flags: must size,
        must free, in, by val, */
        #ifndef _ALPHA_
        /* 28 */ NdrFcShort( 0x10 ), /* ia64 Stack
        size/offset = 16 */

```

```

#else
        NdrFcShort( 0x8 ), /*
        axp64 Stack size/offset = 8 */
        #endif
        /* 30 */ NdrFcShort( 0x3b6 ), /* Type
        Offset=950 */

        /* Parameter txn_out */

        /* 32 */ NdrFcShort( 0x6113 ), /* Flags:
        must size, must free, out, simple ref, srv alloc
        size=24 */
        #ifndef _ALPHA_
        /* 34 */ NdrFcShort( 0x28 ), /* ia64 Stack
        size/offset = 40 */
        #else
        NdrFcShort( 0x20 ), /*
        axp64 Stack size/offset = 32 */
        #endif
        /* 36 */ NdrFcShort( 0x3c8 ), /* Type
        Offset=968 */

        /* Return value */

        /* 38 */ NdrFcShort( 0x70 ), /* Flags: out, return,
        base type, */
        #ifndef _ALPHA_
        /* 40 */ NdrFcShort( 0x30 ), /* ia64 Stack
        size/offset = 48 */
        #else
        NdrFcShort( 0x28 ), /*
        axp64 Stack size/offset = 40 */
        #endif
        /* 42 */ 0x8, /* FC_LONG */
        /*
        0 */
        /* Procedure Payment */

        /* 44 */ 0x33, /* FC_AUTO_HANDLE */
        /*
        0x6c, */
        /*
        Old Flags: object, Oi2 */
        /* 46 */ NdrFcLong( 0x0 ), /* 0 */
        /* 50 */ NdrFcShort( 0x4 ), /* 4 */
        #ifndef _ALPHA_
        /* 52 */ NdrFcShort( 0x38 ), /* ia64 Stack
        size/offset = 56 */
        #else
        NdrFcShort( 0x30 ), /*
        axp64 Stack size/offset = 48 */
        #endif
        /* 54 */ NdrFcShort( 0x0 ), /* 0 */
        /* 56 */ NdrFcShort( 0x8 ), /* 8 */
        /* 58 */ 0x47, /* Oi2 Flags: srv must
        size, clt must size, has return, has ext, */
        /*
        0x3, */
        3 /*
        */
        /* 60 */ 0xa, /* 10 */
        /*
        0x7, */
        /*
        Ext Flags: new corr desc, clt corr check, srv corr
        check, */
        /* 62 */ NdrFcShort( 0x20 ), /* 32 */
        /* 64 */ NdrFcShort( 0x20 ), /* 32 */

```

```

/* 66 */ NdrFcShort( 0x0 ), /* 0 */
/* 68 */ NdrFcShort( 0x0 ), /* 0 */

    /* Parameter txn_in */

/* 70 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
#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 */
0 */ 0x0, /*
/* 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 */
FC_NON_ENCAPSULATED_UNION */
0x2b, /*
0x9, /*
FC_ULONG */
/* 8 */ 0x7, /* Corr desc: FC_USHORT
*/
0x0, /*
/*
/* 10 */ NdrFcShort( 0xffff ), /* -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 */
FC_STRUCT */
/* 0x15, */
/* 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( 0xffff ), /* -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 */
/* 0x2E, */
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 */
FC_IP */
/* 0x2E, */
/* 0x5a, */
FC_CONSTANT_IID */
/* 332 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 336 */ NdrFcShort( 0x0 ), /* 0 */
/* 338 */ NdrFcShort( 0x0 ), /* 0 */
/* 340 */ 0xc0, /* 192 */
/* 0x0, */
0 */
/* 342 */ 0x0, /* 0 */
/* 0x0, */
0 */
/* 344 */ 0x0, /* 0 */
/* 0x0, */
0 */
/* 346 */ 0x0, /* 0 */
/* 0x46, */
70 */
/* 348 */
/* 0x12, 0x10, */
FC_UP [pointer_deref] */
/* 350 */ NdrFcShort( 0x2 ), /* Offset= 2 (352) */
/* 352 */
/* 0x12, 0x0, */
FC_UP */
/* 354 */ NdrFcShort( 0x1e6 ), /* Offset=
486 (840) */
/* 356 */
/* 0x2a, */
FC_ENCAPSULATED_UNION */
/* 0x89, */
137 */
/* 358 */ NdrFcShort( 0x20 ), /* 32 */
/* 360 */ NdrFcShort( 0xa ), /* 10 */
/* 362 */ NdrFcLong( 0x8 ), /* 8 */
/* 366 */ NdrFcShort( 0x50 ), /* Offset= 80 (446) */
/* 368 */ NdrFcLong( 0xd ), /* 13 */
/* 372 */ NdrFcShort( 0x70 ), /* Offset= 112 (484) */
/* 374 */ NdrFcLong( 0x9 ), /* 9 */
/* 378 */ NdrFcShort( 0x90 ), /* Offset= 144 (522) */
/* 380 */ NdrFcLong( 0xc ), /* 12 */
/* 384 */ NdrFcShort( 0xb0 ), /* Offset= 176 (560) */
/* 386 */ NdrFcLong( 0x24 ), /* 36 */
/* 390 */ NdrFcShort( 0x104 ), /* Offset=
260 (650) */
/* 392 */ NdrFcLong( 0x800d ), /* 32781 */

```

```

/* 396 */ NdrFcShort( 0x120 ), /* Offset=
288 (684) */
/* 398 */ NdrFcLong( 0x10 ), /* 16 */
/* 402 */ NdrFcShort( 0x13a ), /* Offset=
314 (716) */
/* 404 */ NdrFcLong( 0x2 ), /* 2 */
/* 408 */ NdrFcShort( 0x150 ), /* Offset=
336 (744) */
/* 410 */ NdrFcLong( 0x3 ), /* 3 */
/* 414 */ NdrFcShort( 0x166 ), /* Offset=
358 (772) */
/* 416 */ NdrFcLong( 0x14 ), /* 20 */
/* 420 */ NdrFcShort( 0x17c ), /* Offset=
380 (800) */
/* 422 */ NdrFcShort( 0xffffffff ), /* Offset= -1
(421) */
/* 424 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /*
3 */
/* 426 */ NdrFcShort( 0x0 ), /* 0 */
/* 428 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 430 */ NdrFcShort( 0x0 ), /* 0 */
/* 432 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 434 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 438 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 440 */
0x12, 0x0, /*
FC_UP */
/* 442 */ NdrFcShort( 0xfffff74 ), /* Offset= -
140 (302) */
/* 444 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 446 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 448 */ NdrFcShort( 0x10 ), /* 16 */
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */ NdrFcShort( 0x6 ), /* Offset= 6 (458) */
/* 454 */ 0x8, /* FC_LONG */
0x39, /*
FC_ALIGNM8 */
/* 456 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 458 */
0x11, 0x0, /*
FC_RP */
/* 460 */ NdrFcShort( 0xfffff5dc ), /* 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( 0xfffff58 ), /* Offset= -
168 (312) */
/* 482 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 484 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 486 */ NdrFcShort( 0x10 ), /* 16 */
/* 488 */ NdrFcShort( 0x0 ), /* 0 */
/* 490 */ NdrFcShort( 0x6 ), /* Offset= 6 (496) */
/* 492 */ 0x8, /* FC_LONG */
0x39, /*
FC_ALIGNM8 */
/* 494 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 496 */
0x11, 0x0, /*
FC_RP */
/* 498 */ NdrFcShort( 0xfffff5dc ), /* 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( 0xfffff44 ), /* Offset= -
188 (330) */
/* 520 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 522 */

```

```

0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 524 */ NdrFcShort( 0x10 ), /* 16 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ NdrFcShort( 0x6 ), /* Offset= 6 (534) */
/* 530 */ 0x8, /* FC_LONG */
0x39, /*
FC_ALIGNM8 */
/* 532 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 534 */
0x11, 0x0, /*
FC_RP */
/* 536 */ NdrFcShort( 0xfffff5dc ), /* Offset= -
36 (500) */
/* 538 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /*
3 */
/* 540 */ NdrFcShort( 0x0 ), /* 0 */
/* 542 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 544 */ NdrFcShort( 0x0 ), /* 0 */
/* 546 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 548 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 552 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 554 */
0x12, 0x0, /*
FC_UP */
/* 556 */ NdrFcShort( 0x176 ), /* Offset=
374 (930) */
/* 558 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 560 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 562 */ NdrFcShort( 0x10 ), /* 16 */
/* 564 */ NdrFcShort( 0x0 ), /* 0 */
/* 566 */ NdrFcShort( 0x6 ), /* Offset= 6 (572) */
/* 568 */ 0x8, /* FC_LONG */
0x39, /*
FC_ALIGNM8 */
/* 570 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 572 */
0x11, 0x0, /*
FC_RP */
/* 574 */ NdrFcShort( 0xfffff5dc ), /* 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 */
0 /*
/* 588 */ 0x0, /* 0 */
0 /*
/* 590 */ 0x0, /* 0 */
0 /*
/* 592 */ 0x0, /* 0 */
70 /*
/* 594 */
FC_CARRAY */
0x1b, /*
0x0, /*
0 /*
/* 596 */ NdrFcShort( 0x1 ), /* 1 */
/* 598 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 600 */ NdrFcShort( 0x4 ), /* 4 */
/* 602 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 604 */ 0x1, /* FC_BYTE */
0x5b, /*
FC_END */
/* 606 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 608 */ NdrFcShort( 0x18 ), /* 24 */
/* 610 */ NdrFcShort( 0x0 ), /* 0 */
/* 612 */ NdrFcShort( 0xc ), /* Offset= 12 (624) */
/* 614 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 616 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 618 */ NdrFcShort( 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( 0xfffffdc ), /* Offset= -
36 (628) */
/* 666 */
0x1d, /*
FC_SMPARRAY */
0x0, /*
0 */
/* 668 */ NdrFcShort( 0x8 ), /* 8 */
/* 670 */ 0x1, /* FC_BYTE */
0x5b, /*
FC_END */
/* 672 */
0x15, /*
FC_STRUCT */
0x3, /*
3 */
/* 674 */ NdrFcShort( 0x10 ), /* 16 */
/* 676 */ 0x8, /* FC_LONG */
0x6, /*
FC_SHORT */
/* 678 */ 0x6, /* FC_SHORT */
0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 680 */ 0x0, /* 0 */

```

```

NdrFcShort( 0xffffffff
), /* 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( 0xfffffe7
), /* Offset= -25 (672) */
0x5b, /*
FC_END */
/* 700 */
0x11, 0x0, /*
FC_RP */
/* 702 */ NdrFcShort( 0xfffff10 ), /* 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( 0xfffffe6 ), /* Offset= -
26 (704) */

```



```

/* 732 */
FC_CARRAY */
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 */

```

```

FC_END */
/* 784 */
0x5b, /*
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 */
FC_PAD /*
/* 878 */
FC_UP [simple_pointer] */
/* 880 */ 0xc,          /* FC_DOUBLE */
FC_PAD /*
/* 882 */
FC_UP /*
/* 884 */ NdrFcShort( 0xfffffda4 ), /* Offset= -
604 (280) */
/* 886 */
FC_UP [pointer_deref] */
/* 888 */ NdrFcShort( 0xfffffda6 ), /* Offset= -
602 (286) */
/* 890 */
FC_UP [pointer_deref] */
/* 892 */ NdrFcShort( 0xfffffdbc ), /* Offset= -
580 (312) */
/* 894 */
FC_UP [pointer_deref] */
/* 896 */ NdrFcShort( 0xfffffdca ), /* Offset= -
566 (330) */
/* 898 */
FC_UP [pointer_deref] */
/* 900 */ NdrFcShort( 0xfffffdd8 ), /* Offset= -
552 (348) */
/* 902 */
FC_UP [pointer_deref] */
/* 904 */ NdrFcShort( 0x2 ), /* Offset= 2 (906) */
/* 906 */
FC_UP /*
/* 908 */ NdrFcShort( 0x16 ), /* Offset= 22 (930) */
/* 910 */
FC_STRUCT /*
/* 912 */ NdrFcShort( 0x10 ), /* 16 */
/* 914 */ 0x6,          /* FC_SHORT */
FC_BYTE /*
/* 916 */ 0x1,          /* FC_BYTE */
FC_ALIGNM4 /*
/* 918 */ 0x8,          /* FC_LONG */
FC_ALIGNM8 /*
/* 920 */ 0xb,          /* FC_HYPER */
FC_END /*
/* 922 */
FC_UP /*

```

```

/* 924 */ NdrFcShort( 0xfffffff2 ), /* Offset= -
14 (910) */
/* 926 */
FC_UP [simple_pointer] */
/* 928 */ 0x2,          /* FC_CHAR */
FC_PAD /*
/* 930 */
FC_BOGUS_STRUCT /*
/* 932 */ NdrFcShort( 0x20 ), /* 32 */
/* 934 */ NdrFcShort( 0x0 ), /* 0 */
/* 936 */ NdrFcShort( 0x0 ), /* Offset= 0 (936) */
/* 938 */ 0x8,          /* FC_LONG */
FC_LONG /*
/* 940 */ 0x6,          /* FC_SHORT */
FC_SHORT /*
/* 942 */ 0x6,          /* FC_SHORT */
FC_SHORT /*
/* 944 */ 0x4c,          /* FC_EMBEDDED_COMPLEX */
0 /*
/* 946 */ NdrFcShort( 0xfffffc54 ), /* Offset= -
940 (6) */
/* 948 */ 0x5c,          /* FC_PAD */
FC_END /*
/* 950 */ 0xb4,          /* FC_USER_MARSHAL */
131 /*
/* 952 */ NdrFcShort( 0x0 ), /* 0 */
/* 954 */ NdrFcShort( 0x18 ), /* 24 */
/* 956 */ NdrFcShort( 0x0 ), /* 0 */
/* 958 */ NdrFcShort( 0xfffffc44 ), /* Offset= -
956 (2) */
/* 960 */
FC_RP [allocated_on_stack] */
/* 962 */ NdrFcShort( 0x6 ), /* Offset= 6 (968) */
/* 964 */
FC_OP /*
/* 966 */ NdrFcShort( 0xfffffcdc ), /* Offset= -
36 (930) */
/* 968 */ 0xb4,          /* FC_USER_MARSHAL */
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
    {
        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 <sqldb.h>

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

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

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

#define DEFCLPACKSIZE
4096

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

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

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

BOOL APIENTRY DllMain(HMODULE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved)
{
    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:
            DisableThreadLibraryCalls(hModule);
            dbinit(); //
            initialize dblib break;

        case DLL_PROCESS_DETACH: //
            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
 *
 * message number int
 *
 * message state msgstate
 *
 * message severity int
 *
 * message severity severity
 *
 * *msgtext char
 *
 * msgtext printable
message description
 *
 * RETURNS: int
INT_CONTINUE continue if
error is SQLETIME else INT_CANCEL action
 *
 * INT_CANCEL
 *
 * cancel operation
 *
 * COMMENTS: This function also sets the dead
lock dbproc variable if necessary.
 *
 */

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

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

```

CTPCC_DBLIB
*pConn;

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

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

return 0;
}

/* FUNCTION: void UtilStrCpy(char * pDest, char *
pSrc, int n)
*
* PURPOSE:      This function copies n characters
from string pSrc to pDst and places a
*              null character at the
end of the destination string.
*
* ARGUMENTS:   char
               *pDest destination string pointer
               char
               *pSrc  source string pointer
               int
               n
               number of characters to copy
*
* RETURNS:     None
*
* COMMENTS:    Unlike strncpy this function
ensures that the result string is
               always null
terminated.
*/
inline static void UtilStrCpy(char * pDest, const
BYTE * pSrc, int n)
{
    strncpy(pDest, (char *)pSrc, n);
    pDest[n] = '\0';

    return;
}

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

    static SERRORMSG errorMsgs[] =
    {
        { ERR_WRONG_SP_VERSION,
"Wrong version of stored procs on database
server" },

```

```

        { ERR_INVALID_CUST,
"Invalid Customer id,name." },
        { ERR_NO_SUCH_ORDER,
"No orders found for customer." },
        { ERR_RETRIED_TRANS,
"Retries before transaction succeeded." },
        { 0, "" }
    };

}
};

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

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

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_DBLIB_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 dlib ver 6.0 client behavior

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

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

m_dbproc = dbopen(login, szServer);

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

```

```

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

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

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

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

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

DiscardNextResults(2);

// verify that version of stored procs on
server is correct
dbrpcinit(m_dbproc, "tpcc_version", 0);
if (dbrpcexec(m_dbproc) == FAIL)
    ThrowError(CDBLIBERR::eDbRpcExec);

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

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

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

DiscardNextRows(0);
DiscardNextResults(0);
}

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

```

```

delete m_DbLibErr;
if (m_SqlErr != NULL)
    delete m_SqlErr;
}

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

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

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

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

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

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

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

// check for SQL Server error first; if
yes, throw it and ignore any DBLib error.
if (m_SqlErr != NULL)
{
CSQLERR *pSqlErr;

```

```

pSqlErr = m_SqlErr;
m_SqlErr = NULL; // clear our
pointer to instance; catch handler will delete
throw pSqlErr;
}

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

throw pDbLibErr;
}

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

while (TRUE)
{
rc = dbnextrow(m_dbproc);
if (rc == NO_MORE_ROWS)
break;
if (rc == FAIL)
{
if (iExpectedCount >=
0)
ThrowError(CDBLIBERR::eDbNextRow);
else
break;
}
iRowsRead++;

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

// Read and discard results until no more. Throw an
exception if number of result sets read doesn't

```

```

// match number expected. The result set count will
// be ignored if the expected count value
// passed in is negative. A typical use of this
// routine is to verify that there are no more
// result sets to be read.
void CTPCC_DBLIB::DiscardNextResults(int
iExpectedCount)
{
    int          iResultsRead = 0;
    RETCODE     rc;

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

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

        DiscardNextRows(-1);
        iResultsRead++;
    }

    if ((iExpectedCount >= 0) &&
(iExpectedCount != iResultsRead))

        ThrowError(CDBLIBERR::eWrongRowCount);
}

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

    ResetError();

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

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

```

```

        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);

        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

||

== iErrOleDbProvider &&
strchr(e-
>m_msgtext, sErrTimeoutExpired) != NULL) &&
(++iTryCount
<= iMaxRetries))

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

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

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

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

```

```

    DBDATEREC   daterec;

    int          iTryCount =
0;
    const BYTE   *pData;

    ResetError();

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

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

            // check whether any
            order lines are for a remote warehouse

            m_txn.NewOrder.o_all_local = 1;
            for (i = 0; i <
m_txn.NewOrder.o_ol_cnt; i++)
            {
                if
(m_txn.NewOrder.OL[i].ol_supply_w_id !=
m_txn.NewOrder.w_id)

                    {
                        m_txn.NewOrder.o_all_local = 0; // at
                        least one remote warehouse

                        break;
                    }
            }

            dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.o_all_local);

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

```

```

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

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

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

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

        if(pData=dbdata(m_dbproc, 1))
            UtilStrCpy(m_txn.NewOrder.OL[i].ol_i_name,
pData, dbdatlen(m_dbproc, 1));

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

        if(pData=dbdata(m_dbproc, 3))
            UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_ge
neric, pData, dbdatlen(m_dbproc, 3));

        if(pData=dbdata(m_dbproc, 4))
            dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc, 4),
SQLFLT8, (BYTE
*)&m_txn.NewOrder.OL[i].ol_i_price, 8);

        if(pData=dbdata(m_dbproc, 5))

            dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc, 5),
SQLFLT8, (BYTE
*)&m_txn.NewOrder.OL[i].ol_amount, 8);

```

```

        m_txn.NewOrder.total_amount =
m_txn.NewOrder.total_amount +
m_txn.NewOrder.OL[i].ol_amount;

        DiscardNextRows(0);
    }

    // get remaining values
for w_tax, d_tax, o_id, c_last, c_discount, c_credit,
o_entry_d, commit_flag
    if (dbresults(m_dbproc)
!= SUCCEED)
        ThrowError(CDBLIBERR::eDbResults);

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

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

    if
(pData=dbdata(m_dbproc, 1))
        dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc, 1), SQLFLT8, (BYTE
*)&m_txn.NewOrder.w_tax, 8);

    if
(pData=dbdata(m_dbproc, 2))
        dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc, 2), SQLFLT8, (BYTE
*)&m_txn.NewOrder.d_tax, 8);

    if
(pData=dbdata(m_dbproc, 3))
        m_txn.NewOrder.o_id = (*DBINT *) pData);

    if
(pData=dbdata(m_dbproc, 4))
        UtilStrCpy(m_txn.NewOrder.c_last, pData,
dbdatlen(m_dbproc, 4));

    if
(pData=dbdata(m_dbproc, 5))
        dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc, 5), SQLFLT8, (BYTE
*)&m_txn.NewOrder.c_discount, 8);

    if
(pData=dbdata(m_dbproc, 6))

```

```

        UtilStrCpy(m_txn.NewOrder.c_credit, pData,
dbdatlen(m_dbproc, 6));
    if
(pData=dbdata(m_dbproc, 7))
    {
        datetime =
*(DBDATETIME *) pData);
        dbdatecrack(m_dbproc, &daterec, &datetime);
        m_txn.NewOrder.o_entry_d.year =
daterec.year;
        m_txn.NewOrder.o_entry_d.month =
daterec.month;
        m_txn.NewOrder.o_entry_d.day =
daterec.day;
        m_txn.NewOrder.o_entry_d.hour =
daterec.hour;
        m_txn.NewOrder.o_entry_d.minute =
daterec.minute;
        m_txn.NewOrder.o_entry_d.second =
daterec.second;
    }
    if
(pData=dbdata(m_dbproc, 8))
        commit_flag =
(*DBTINYINT *) pData);

        DiscardNextRows(0);
        DiscardNextResults(0);

        if (commit_flag == 1)
        {
            m_txn.NewOrder.total_amount *= ((1 +
m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));

            m_txn.NewOrder.exec_status_code = eOK;
        }
        else
            m_txn.NewOrder.exec_status_code =
eInvalidItem;

        return;
    }
    catch (CSQLERR *e)
    {
        if ((e->m_msgno == 1205
||
== iErrOleDbProvider &&
(e->m_msgno
>m_msgtext, sErrTimeoutExpired) != NULL)) &&

```

```

        (++iTryCount
<= iMaxRetries))
    {
        // hit
        deadlock; backoff for increasingly longer period
        delete e;
        Sleep(10 *
iTryCount);
    }
    else
        throw;
}
// while (TRUE)
//
// if (iTryCount)
//     throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

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

    int              iTryCount =
0;
    const BYTE      *pData;

    ResetError();

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

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

            // if customer id is
            zero, then payment is by name
            if (m_txn.Payment.c_id
== 0)

```

```

            dbrpcparam(m_dbproc, NULL, 0, SQLCHAR, -1,
strlen(m_txn.Payment.c_last), (unsigned char
*)m_txn.Payment.c_last);

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

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

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

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

            if
(pData=dbdata(m_dbproc, 1))
                m_txn.Payment.c_id = *((DBINT *) pData);
            if
(pData=dbdata(m_dbproc, 2))
                UtilStrCpy(m_txn.Payment.c_last, pData,
dbdatlen(m_dbproc, 2));
            if
(pData=dbdata(m_dbproc, 3))
                {
                    datetime =
*((DBDATETIME *) pData);
                    dbdatecrack(m_dbproc, &daterec, &datetime);
                    m_txn.Payment.h_date.year = daterec.year;
                    m_txn.Payment.h_date.month =
daterec.month;
                    m_txn.Payment.h_date.day = daterec.day;
                    m_txn.Payment.h_date.hour = daterec.hour;
                    m_txn.Payment.h_date.minute =
daterec.minute;
                    m_txn.Payment.h_date.second =
daterec.second;
                }
            if
(pData=dbdata(m_dbproc, 4))
                UtilStrCpy(m_txn.Payment.w_street_1, pData,
dbdatlen(m_dbproc, 4));

```

```

            if
(pData=dbdata(m_dbproc, 5))
                UtilStrCpy(m_txn.Payment.w_street_2, pData,
dbdatlen(m_dbproc, 5));
            if
(pData=dbdata(m_dbproc, 6))
                UtilStrCpy(m_txn.Payment.w_city, pData,
dbdatlen(m_dbproc, 6));
            if
(pData=dbdata(m_dbproc, 7))
                UtilStrCpy(m_txn.Payment.w_state, pData,
dbdatlen(m_dbproc, 7));
            if
(pData=dbdata(m_dbproc, 8))
                UtilStrCpy(m_txn.Payment.w_zip, pData,
dbdatlen(m_dbproc, 8));
            if
(pData=dbdata(m_dbproc, 9))
                UtilStrCpy(m_txn.Payment.d_street_1, pData,
dbdatlen(m_dbproc, 9));
            if
(pData=dbdata(m_dbproc, 10))
                UtilStrCpy(m_txn.Payment.d_street_2, pData,
dbdatlen(m_dbproc, 10));
            if
(pData=dbdata(m_dbproc, 11))
                UtilStrCpy(m_txn.Payment.d_city, pData,
dbdatlen(m_dbproc, 11));
            if
(pData=dbdata(m_dbproc, 12))
                UtilStrCpy(m_txn.Payment.d_state, pData,
dbdatlen(m_dbproc, 12));
            if
(pData=dbdata(m_dbproc, 13))
                UtilStrCpy(m_txn.Payment.d_zip, pData,
dbdatlen(m_dbproc, 13));
            if
(pData=dbdata(m_dbproc, 14))
                UtilStrCpy(m_txn.Payment.c_first, pData,
dbdatlen(m_dbproc, 14));
            if
(pData=dbdata(m_dbproc, 15))
                UtilStrCpy(m_txn.Payment.c_middle, pData,
dbdatlen(m_dbproc, 15));
            if
(pData=dbdata(m_dbproc, 16))
                UtilStrCpy(m_txn.Payment.c_street_1, pData,
dbdatlen(m_dbproc, 16));
            if
(pData=dbdata(m_dbproc, 17))

```



```

        UtilStrCpy(m_txn.Payment.c_street_2, pData,
dbdatlen(m_dbproc, 17));
        if
(pData=dbdata(m_dbproc, 18))
        UtilStrCpy(m_txn.Payment.c_city, pData,
dbdatlen(m_dbproc, 18));
        if
(pData=dbdata(m_dbproc, 19))
        UtilStrCpy(m_txn.Payment.c_state, pData,
dbdatlen(m_dbproc, 19));
        if
(pData=dbdata(m_dbproc, 20))
        UtilStrCpy(m_txn.Payment.c_zip, pData,
dbdatlen(m_dbproc, 20));
        if
(pData=dbdata(m_dbproc, 21))
        UtilStrCpy(m_txn.Payment.c_phone, pData,
dbdatlen(m_dbproc, 21));
        if
(pData=dbdata(m_dbproc, 22))
        {
                datetime =
*((DBDATETIME *) pData);
                dbdatecrack(m_dbproc, &daterec, &datetime);
                m_txn.Payment.c_since.year =
daterec.year;
                m_txn.Payment.c_since.month =
daterec.month;
                m_txn.Payment.c_since.day = daterec.day;
                m_txn.Payment.c_since.hour =
daterec.hour;
                m_txn.Payment.c_since.minute =
daterec.minute;
                m_txn.Payment.c_since.second =
daterec.second;
        }
        if(pData=dbdata(m_dbproc, 23))
        UtilStrCpy(m_txn.Payment.c_credit, pData,
dbdatlen(m_dbproc, 23));
        if(pData=dbdata(m_dbproc, 24))
        dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,24), SQLFLT8, (BYTE
*)&m_txn.Payment.c_credit_lim, 8);
        if(pData=dbdata(m_dbproc, 25))
        dbconvert(m_dbproc, SQLNUMERIC,

```

```

(LPCBYTE)pData, dbdatlen(m_dbproc,25), SQLFLT8, (BYTE
*)&m_txn.Payment.c_discount, 8);
        if(pData=dbdata(m_dbproc, 26))
        dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,26), SQLFLT8, (BYTE
*)&m_txn.Payment.c_balance, 8);
        if(pData=dbdata(m_dbproc, 27))
        UtilStrCpy(m_txn.Payment.c_data, pData,
dbdatlen(m_dbproc, 27));
        DiscardNextRows(0);
        DiscardNextResults(0);
        if (m_txn.Payment.c_id
== 0)
                throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
        else
                m_txn.Payment.exec_status_code = eOK;
        return;
    }
    catch (CSQLERR *e)
    {
        if ((e->m_msgno == 1205
||
== iErrOleDbProvider &&
(e->m_msgno
>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
    DBDATETIME
    DBDATERECDaterec;
    datetime;

```

```

        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);
NO_MORE_ROWS)
break;
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.minut
e = daterec.minute;
m_txn.OrderStatus.OL[i].ol_delivery_d.secon
d = daterec.second;
}
i++;
}

```

```

m_txn.OrderStatus.o_ol_cnt = i;
if (dbresults(m_dbproc)
!= SUCCEEDED)
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)
!= SUCCEEDED)

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

```

```

*
* PURPOSE: Header file for TPC-C txn class
* implementation.
*
* Change history:
* 4.20.000 - updated rev number to
match kit
*/
#pragma once

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

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

class CSQLERR : public CBaseErr
{
public:
    CSQLERR(void)
    {
        m_msgno = 0;
        m_msgstate = 0;
        m_severity = 0;
        m_msgtext = NULL;
    };

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

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

    int ErrorType() {return
ERR_TYPE_SQL;};

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

class CDBLIBERR : public CBaseErr
{
public:
    enum ACTION
    {
        eNone,
        eUnknown,
        eLogin,
        // error from dblogin
        eDbOpen,
        // error from dbopen
    };
};

```

```

        eDbUse,
// error from dbuse
        eDbSqlExec,
// error from dbsqlexec
        eDbSet,
// error from one of the dbset*
routines
        eDbNextRow,
// error from dbnextrow
        eWrongRowCount,
// more or less rows returned than expected
        eWrongNumCols,
// more or less columns returned than
expected
        eDbResults,
// error from dbresults
        eDbRpcExec,
// error from dbrpcexec
        eDbSetMaxProcs,
// error from dbsetmaxprocs
        eDbProcHandler
// error from either dbprocerrhandle or
dbprocmsghandle
    };

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

        m_dberrstr = NULL;
        m_oserrstr = NULL;
    };

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

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

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

class CTPCC_DBLIB_ERR : public CBaseErr
{
public:
    enum CTPCC_DBLIB_ERRS

```

```

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

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

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

    int        m_errno;
    int        m_iTryCount;

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

    char *ErrorText();
};

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

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

    union
    {
        NEW_ORDER_DATA
        Payment;
        DELIVERY_DATA
        Delivery;
    };
};

```

```

        STOCK_LEVEL_DATA
        StockLevel;
        ORDER_STATUS_DATA
        OrderStatus;
    };
    m_txn;
public:
    CTPCC_DBLIB(LPCSTR szServer,
LPCSTR szUser, LPCSTR szPassword, LPCSTR szHost,
LPCSTR szDatabase );
    ~CTPCC_DBLIB(void);

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

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

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

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

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

tpcc_odbc.cpp
/* FILE: TPCC_ODBC.CPP
* Microsoft
TPC-C Kit Ver. 4.20.000

```

```

*
* Copyright
Microsoft, 1999
* All Rights Reserved
*
* Version
4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99
*
* PURPOSE: Implements ODBC calls for TPC-C
txns.
* Contact: Charles Levine
(clevine@microsoft.com)
*
* Change history:
* 4.20.000 - updated rev number to
match kit
* 4.10.001 - not deleting error
class in catch handler on deadlock retry;
* not a
functional bug, but a memory leak
*/

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

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

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

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

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

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

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

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

static SQLHENV henv = SQL_NULL_HENV;
// ODBC environment handle

BOOL WINAPI DllMain(HMODULE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved)
{
switch( ul_reason_for_call )
{
case DLL_PROCESS_ATTACH:

```

```

DisableThreadLibraryCalls(hModule);
if (
SQLAllocHandleStd(SQL_HANDLE_ENV, SQL_NULL_HANDLE,
&henv) != SQL_SUCCESS )
return FALSE;
break;
case DLL_PROCESS_DETACH:
if (henv != NULL)
SQLFreeEnv(henv);
break;
default:
/* nothing */;
}
return TRUE;
}

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

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

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

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

```

```

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

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

// initialization
m_hdbc = SQL_NULL_HDBC;
m_hstmt = SQL_NULL_HSTMT;

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

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

if ( SQLAllocHandle(SQL_HANDLE_DBC, henv,
&m_hdbc) != SQL_SUCCESS )

ThrowError(CODBCERR::eAllocHandle);

if ( SQLSetConnectOption(m_hdbc,
SQL_PACKET_SIZE, 4096) != SQL_SUCCESS )

ThrowError(CODBCERR::eConnOption);

{
char
szConnectStr[256];

```

```

        char
        szOutStr[1024];
        SQLSMALLINT
        iOutStrLen;

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

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

        if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)
            ThrowError(CODBCERR::eConnect);
    }
    if (SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmt) != SQL_SUCCESS)

        ThrowError(CODBCERR::eAllocHandle);
    {
        char                buffer[128];

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

            ThrowError(CODBCERR::eExecDirect);

        // verify that version of stored
procs on server is correct
        char db_sp_version[10];
        strcpy(buffer, "{call
tpcc_version}");
        rc = SQLExecDirect(m_hstmt,
(unsigned char *)buffer, SQL_NTS);
        if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

            ThrowError(CODBCERR::eExecDirect);
        if ( SQLBindCol(m_hstmt, 1,
SQL_C_CHAR, &db_sp_version, sizeof(db_sp_version),
NULL) != SQL_SUCCESS )

            ThrowError(CODBCERR::eBindCol);
        if ( SQLFetch(m_hstmt) ==
SQL_ERROR )

            ThrowError(CODBCERR::eFetch);
        if
        (strcmp(db_sp_version,sVersion))

```

```

        throw new
CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_WRONG_SP_VERSION
);

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

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

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

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

    pODBCErr = new CODBCERR();

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

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

```

```

        break;

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

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

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

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

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

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

    SQLFreeStmt(m_hstmt, SQL_CLOSE);
    throw pODBCErr;
}

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

        ThrowError(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtStockLevel;

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

```

```

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

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

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

    m_hstmt = m_hstmtStockLevel;

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

            if ( SQLFetch(m_hstmt)
== SQL_ERROR )
                ThrowError(CODBCERR::eFetch);

            SQLFreeStmt(m_hstmt,
SQL_CLOSE);

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

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

    // if (iTryCount)

```

```

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

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

    m_hstmt = m_hstmtNewOrder;

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

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
&m_txn.NewOrder.w_id, 0, NULL) != SQL_SUCCESS
    )
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.NewOrder.d_id, 0, NULL) != SQL_SUCCESS
    )
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.c_id, 0, NULL) != SQL_SUCCESS
    )
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.NewOrder.o_ol_cnt, 0, NULL) != SQL_SUCCESS
    )
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.NewOrder.o_all_local, 0, NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindParam);

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

```

```

    )
    ThrowError(CODBCERR::eBindParam);
}

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

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

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

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

```

```

        || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_no_commit_flag, 0, NULL) !=
SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindCol);
}

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

0        1        2        //
//
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++)
        {
            // set the
bind offset value...
            m_BindOffset
= i * sizeof(m_txn.NewOrder.OL[0]);

            if (
SQLFetch(m_hstmt) == SQL_ERROR)

                ThrowError(CODBCERR::eFetch);

            // move to
the next resultset

            if (
SQLMoreResults(m_hstmt) == SQL_ERROR )

                ThrowError(CODBCERR::eMoreResults);

            m_txn.NewOrder.total_amount +=
m_txn.NewOrder.OL[i].ol_amount;
        }

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

            ThrowError(CODBCERR::eSetStmtAttr);

        if ( SQLFetch(m_hstmt)
== SQL_ERROR)

            ThrowError(CODBCERR::eFetch);

        SQLFreeStmt(m_hstmt,
SQL_CLOSE);

```

```

        if (m_no_commit_flag ==
1)
        {
            m_txn.NewOrder.total_amount *= ((1 +
m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));

            m_txn.NewOrder.exec_status_code = eOK;
        }
        else

            m_txn.NewOrder.exec_status_code =
eInvalidItem;

        break;
    }
    catch (CODBCERR *e)
    {
        if (!e->m_bDeadLock)

            throw;

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

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

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

        ThrowError(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtPayment;

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
&m_txn.Payment.w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
&m_txn.Payment.c_w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_DOUBLE, SQL_NUMERIC, 6, 2,
&m_txn.Payment.h_amount, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.Payment.d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.Payment.c_d_id, 0, NULL) != SQL_SUCCESS

```



```

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

    i = 0;
    if ( SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.Payment.c_id, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_street_1,
sizeof(m_txn.Payment.c_street_1), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.Payment.h_date,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.w_street_1,
sizeof(m_txn.Payment.w_street_1), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.w_street_2,
sizeof(m_txn.Payment.w_street_2), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.w_city,
sizeof(m_txn.Payment.w_city), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.w_state,
sizeof(m_txn.Payment.w_state), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.w_zip,
sizeof(m_txn.Payment.w_zip), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.d_street_1,
sizeof(m_txn.Payment.d_street_1), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.d_street_2,
sizeof(m_txn.Payment.d_street_2), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.d_city,
sizeof(m_txn.Payment.d_city), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.d_state,
sizeof(m_txn.Payment.d_state), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.d_zip,
sizeof(m_txn.Payment.d_zip), NULL) !=
SQL_SUCCESS

```

```

        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_first,
sizeof(m_txn.Payment.c_first), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_middle,
sizeof(m_txn.Payment.c_middle), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_street_1,
sizeof(m_txn.Payment.c_street_1), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_street_2,
sizeof(m_txn.Payment.c_street_2), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_city,
sizeof(m_txn.Payment.c_city), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_state,
sizeof(m_txn.Payment.c_state), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_zip,
sizeof(m_txn.Payment.c_zip), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_phone,
sizeof(m_txn.Payment.c_phone), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.Payment.c_since,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_credit,
sizeof(m_txn.Payment.c_credit), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.Payment.c_credit_lim, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.Payment.c_discount, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.Payment.c_balance, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_data,
sizeof(m_txn.Payment.c_data), NULL) !=
SQL_SUCCESS
    )
    ThrowError(CODBCERR::eBindCol);
}

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

    m_hstmt = m_hstmtPayment;

```

```

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

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

            if ( SQLFetch(m_hstmt)
== SQL_ERROR)
                ThrowError(CODBCERR::eFetch);

            SQLFreeStmt(m_hstmt,
SQL_CLOSE);

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

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

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

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

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

```

```

)
ThrowError(CODBCERR::eAllocHandle);

m_hstmt = m_hstmtOrderStatus;

if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )

ThrowError(CODBCERR::eSetStmtAttr);

int i = 0;
if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
&m_txn.OrderStatus.w_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.OrderStatus.d_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.OrderStatus.c_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
sizeof(m_txn.OrderStatus.c_last), 0,
&m_txn.OrderStatus.c_last,
sizeof(m_txn.OrderStatus.c_last), NULL) !=
SQL_SUCCESS
)
ThrowError(CODBCERR::eBindParam);

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

ThrowError(CODBCERR::eSetStmtAttr);

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

```

```

if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols2,
SQL_IS_POINTER ) != SQL_SUCCESS )

ThrowError(CODBCERR::eSetStmtAttr);

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

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

m_hstmt = m_hstmtOrderStatus;

if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )

ThrowError(CODBCERR::eSetStmtAttr);

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

while (TRUE)
{
try
{

```

```

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

ThrowError(CODBCERR::eSetStmtAttr);

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

ThrowError(CODBCERR::eExecDirect);

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

ThrowError(CODBCERR::eSetStmtAttr);

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

ThrowError(CODBCERR::eFetchScroll);

m_txn.OrderStatus.o_ol_cnt =
(short)m_RowsFetched;

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

ThrowError(CODBCERR::eSetStmtAttr);

if (
SQLMoreResults(m_hstmt) == SQL_ERROR )

ThrowError(CODBCERR::eMoreResults);

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

ThrowError(CODBCERR::eFetch);
}

SQLFreeStmt(m_hstmt,
SQL_CLOSE);

```

```

        if
(m_txn.OrderStatus.o_ol_cnt == 0)
            throw new
CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_NO_SUCH_ORDER );
        else if
(m_txn.OrderStatus.c_id == 0 &&
m_txn.OrderStatus.c_last[0] == 0)
            throw new
CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_INVALID_CUST );
        else
            m_txn.OrderStatus.exec_status_code = eOK;
            break;
        }
        catch (COBDCERR *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(COBDCERR::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(COBDCERR::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(COBDCERR::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(COBDCERR::eExecDirect);

            if ( SQLFetch(m_hstmt)
== SQL_ERROR )

                ThrowError(COBDCERR::eFetch);

            SQLFreeStmt(m_hstmt,
SQL_CLOSE);

            m_txn.Delivery.exec_status_code = eOK;
            break;
        }
        catch (COBDCERR *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_odbch


---


/*      FILE:          TPCC_ODBC.H
*
*      TPC-C Kit Ver. 4.20.000
*
*      Microsoft
*      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 COBDCERR : 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
        eSetStmntAttr,
        // error from SQLSetStmntAttr
    };

    COBDCERR(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;
// 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;
        int
m_RowsFetched;
        int
m_no_commit_flag;

        void ThrowError( CODBCERR::ACTION
eAction );

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

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

    public:

```

```

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

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

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

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

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

```

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;
    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];
    double
    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;

```

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
{
    31 BYTE OL_Count; //range 0 to
    31 BYTE OL_Remote_Count; //range 0 to
    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;
    int      DeltaT2;
    int      DeltaT3;
    int      DeltaT4;
    int      RTDelay;
    int      TxnError;
    // error code providing more detail for
TxnStatus
    int      w_id;
    // warehouse ID
    BYTE      d_id;
    // assigned district ID for this thread
    BYTE      d_id_ThisTxn;
    // district ID chosen for this particular
    BYTE      TxnStatus;
    // completion status for txn to indicate
errors
    BYTE      reserved;
    // for word alignment
    TXN_DETAILS      TxnDetails;
    //
} TXN_RECORD_TPCC, *PTXN_RECORD_TPCC;

```

```

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

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

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

////////////////////////////////////
////////////////////////////////////
// The transaction log has a header as the
first 4K block.
//
typedef struct _TXN_LOG_HEADER
{
    char
    EyeCatcher[2]; // signature bytes;
    // should always be "BC"
    int
    LogVersion; // set to
TXN_LOG_VERSION

```

```

JULIAN_TIME
    BeginTxnTS; // timestamp
of first (lowest) txn start
    JULIAN_TIME
    EndTxnTS; // timestamp of last
(highest) txn completion time
    int
    iRecCount; // number of
records in log file
    BOOL
    bLogSorted;
    int
    iFileSize; // file size
in bytes
    // the record map provides a fast
way to get close to a particular timestamp in a
sorted log file.
    struct
    {
        JULIAN_TIME
        TS; // timestamp
of record
        int
        iPos; // byte
position in file
    }
    RecMap[RecMapSize];
#define RecMapSize
200
} TXN_LOG_HEADER, *PTXN_LOG_HEADER;

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

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

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

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

#define TXN_LOG_OS_ERROR 1
#define TXN_LOG_NOT_SORTED 2

```

```

#define SKIP_CTRL_RECS          1

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

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

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

    BOOL
    bLogSorted;
    //
    is log file sorted? applies to both input and output

```

```

    JULIAN_TIME
    BeginTxnTS;
    timestamp of first (lowest) txn start
    JULIAN_TIME
    EndTxnTS;
    // timestamp
    of last (highest) txn completion time
    int
    iRecCount;
    //
    number of records in log file
    BYTE
    *pCurrent;
    //ptr to
    current buffer
    BYTE
    *pBuffer[MAX_NUM_BUFFERS];
    PTXN_RECORD_HEADER *TxnArray;
    //transaction record pointer
    array for sort
    DWORD          dwError;
    HANDLE         hTxnFile;
    //handle to log file
    HANDLE         hMapFile;
    //map file used when
    sorting the log
    HANDLE         hIoComplete;
    //event to signify that
    there are no pending IOs
    HANDLE         hLogFileIo;
    //event to
    signal the IO thread to write the inactive buffer
    Spinlock       Spin;
    //spin lock to protect
    the txn log file buffers
    FILE
    *tmpFile;
    //temp file for merging
    sorted pieces
    PBLOCK_HEADER
    tmpHeaders;
    //sorted
    pointers block header
    BYTE
    **recPointers;
    //record pointer
    buffers for each sorted block
    PTXN_RECORD_HEADER *recBuffers;
    //record buffers for each sorted block
    int
    *PointersRead;
    //# of pointers processed in each block
    BOOL
    *BlockAvailable;
    //whether to check a particular
    block for jmin
    int
    nBlocks;
    int
    jmin;

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

```

```

    int
    iSortedReturnedCount;
    //keeps track of the # of sorted records
    returned through GetSortedRecord()
    int Write(BYTE *ptr, DWORD Size);
    static void LogFileIO(CTxnLog *);
    void LoadBuffers(int j);
    //used in sort/merge to load
    record buffers
    public:
    CTxnLog::CTxnLog(LPCTSTR
    szFileName, DWORD dwOpts);
    ~CTxnLog(void);
    int WriteToLog(PTXN_RECORD_TPCC
    pTxnRcnd);
    int
    WriteToLog(PTXN_RECORD_TPCC_DELIV_DEF pTxnRcnd);
    int
    WriteToLog(PTXN_RECORD_CONTROL pCtrlRec);
    int WriteToLog(PTXN_RECORD_HEADER
    pCtrlRec);
    int WriteCtrlRecToLog(BYTE
    SubType, LPCTSTR lpStr, DWORD dwLen);
    void
    CloseTransactionLogFile(void);
    PTXN_RECORD_HEADER
    GetNextRecord(BOOL bSkipCtrlRecs = FALSE);
    PTXN_RECORD_HEADER
    GetNextRecord(JULIAN_TIME SeekTimeT0, BOOL
    bSkipCtrlRecs = FALSE);
    int Sort(void);
    PTXN_RECORD_HEADER
    GetSortedRecord();
    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:

RunSQLCfg.sql

```
/* TPC-C Benchmark Kit */
/* */
/* RUNSQLCFG.SQL */
/* This script file is used to set runtime server configuration parameters */
/* */

exec sp_configure "show advanced option", 1
go

reconfigure with override
go

/* change this value to approximately the number of connected users */
exec sp_configure "max worker threads",255

/* increase priority of user threads */
exec sp_configure "priority boost",1

/* disable automatic checkpointing */
exec sp_configure "recovery interval",32767

/* change to a mask appropriate for the number of processors on the server */
exec sp_configure "affinity mask",0xf

/* enable fibers */
exec sp_configure "lightweight pooling",1

go

reconfigure with override
go
```

VerifyTpccLoad.sql

```
-- File: VERIFYPCCLOAD.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.22
-- Copyright Microsoft, 2001
-- Purpose: Performs series of TPCC database checks to verify
```

```
-- that database load completed correctly

print " "
select convert(char(30), getdate(),9)
print " "

use tpcc
go

-- *****
-- Check rows per table from SYSINDEXES
-- *****

print 'WAREHOUSE TABLE'

select rows
from sysindexes
where id = object_id("warehouse")
go

print 'DISTRICT TABLE = (10 * No of warehouses)'

select rows
from sysindexes
where id =object_id("district")
go

print 'ITEM TABLE = 100,000'

select rows
from sysindexes
where id =object_id("item")
go

print 'CUSTOMER TABLE = (30,000 * No of warehouses)'

select rows
from sysindexes
where id =object_id("customer")
go

print 'ORDERS TABLE = (30,000 * No of warehouses)'

select rows
from sysindexes
where id =object_id("orders")
go

print 'HISTORY TABLE = (30,000 * No of warehouses)'

select rows
from sysindexes
where id =object_id("history")
go

print 'STOCK TABLE = (100,000 * No of warehouses)'

select rows
from sysindexes
where id =object_id("stock")
go
```

```

print      'ORDER_LINE TABLE = (300,000 * No of warehouses + some change)'

select    rows
from      sysindexes
where     id      =object_id("order_line")
go

print      'NEW_ORDER TABLE = (9000 * No of warehouses)'

select    rows
from      sysindexes
where     id      =object_id("new_order")
go

--      *****
--      Check indices
--      *****

print '*****Index Check*****'

use tpcc
go

sp_helpindex      customer
go

sp_helpindex      stock
go

sp_helpindex      district
go

sp_helpindex      item
go

sp_helpindex      new_order
go

sp_helpindex      orders
go

sp_helpindex      order_line
go

sp_helpindex      warehouse
go

```

backup.sql

```

-- File:      BACKUP.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- 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 tpccback48001, tpccback48002 with init, stats = 1

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

backupdev.sql

```

-- File:      BACKUPDEVB.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates tpcc database Backup Devices

use master
go

-- create backup devices

exec sp_addumpdevice 'disk','tpccback48001','Z:\tpccback48001.dmp'
go
exec sp_addumpdevice 'disk','tpccback48002','Y:\tpccback48002.dmp'
go

```

config.sql

```

-- File:      CONFIG.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.00
--           Copyright Microsoft, 1996
-- Purpose:   Collects SQL Server configuration parameters

print " "
select convert(char(30), getdate(),9)
print " "
go

sp_configure "show advanced",1
go
reconfigure with override
go
exec sp_configure "affinity mask", 15
exec sp_configure "cost threshold for parallelism", 5
exec sp_configure "index create memory", 704
exec sp_configure "lightweight pooling", 1
exec sp_configure "awe enabled", 1
exec sp_configure "locks", 0
exec sp_configure "max degree of parallelism", 1
exec sp_configure "max server memory", 2147483647
exec sp_configure "max worker threads", 215
exec sp_configure "min memory per query", 512
exec sp_configure "min server memory", 0
exec sp_configure "nested triggers", 1
exec sp_configure "network packet size", 2048
exec sp_configure "open objects", 0
exec sp_configure "priority boost", 1
exec sp_configure "recovery interval", 100
exec sp_configure "set working set size", 0

```

```

exec sp_configure "user connections",          0

go

reconfigure with override
go
sp_configure
go

```

createdb.sql

```

-- File:      CREATEDB.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates tpcc database and backup files

use master
go

--           Create temporary table for timing

if exists ( select name from sysobjects where name = 'tpcc_timer' )
drop table tpcc_timer

go

create table tpcc_timer
(
    start_date          char(30),
    end_date            char(30)
)

insert into tpcc_timer values (0,0)
go

--           Store starting time

update tpcc_timer
set start_date          = (select convert(char(30), getdate(),9))
go

-- create main database files

CREATE DATABASE tpcc
ON PRIMARY
(
    NAME              = MSSQL_tpcc_root,
    FILENAME          = "C:\MSSQL_tpcc_root.mdf",
    SIZE              = 8MB,
    FILEGROWTH        =0),
FILEGROUP MSSQL_cs_fg
(
    NAME              = MSSQL_cs1,
    FILENAME          = "F:",
    SIZE              = 78000MB,
    FILEGROWTH        = 0),
(
    NAME              = MSSQL_cs2,
    FILENAME          = "G:",
    SIZE              = 78000MB,
    FILEGROWTH        = 0),
(
    NAME              = MSSQL_cs3,
    FILENAME          = "H:",

```

```

    SIZE              = 78000MB,
    FILEGROWTH        = 0),
(
    NAME              = MSSQL_cs4,
    FILENAME          = "I:",
    SIZE              = 78000MB,
    FILEGROWTH        = 0),
FILEGROUP MSSQL_misc_fg
(
    NAME              = MSSQL_misc1,
    FILENAME          = "J:",
    SIZE              = 149900MB,
    FILEGROWTH        = 0)

LOG ON
(
    NAME              =MSSQL_tpcc_log,
    FILENAME          = "E:",
    SIZE              =120000MB,
    FILEGROWTH        =0)

COLLATE Latin1_General_Bin
go

-- Store ending time
update tpcc_timer
set end_date = (select convert(char(30), getdate(),9))
go

select "Elapsed time (in seconds): ", datediff(second,(select start_date from
tpcc_timer),(select end_date from tpcc_timer))

--           remove temporary table

if exists ( select name from sysobjects where name = 'tpcc_timer' )
drop table tpcc_timer

go

exec sp_dboption tpcc,'torn page',false
go

```

dbopt1.sql

```

-- File:      DBOPT1.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Sets database options for data load

use master
go

exec sp_dboption tpcc,'select into/bulkcopy',true
exec sp_dboption tpcc,'trunc. log on chkpt.',true
go

use tpcc
go

checkpoint
go

```

dbopt2.sql

```
-- File:      DBOPT2.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Resets database options after data load

sp_dboption tpcc,'select into/bulkcopy',FALSE
GO

sp_dboption tpcc,'trunc. log on chkpt.',FALSE
GO

USE tpcc
GO

CHECKPOINT
GO

sp_configure 'allow updates',1
GO

RECONFIGURE WITH OVERRIDE
GO

DECLARE @msg          varchar(50)

--
-- OPTIONS FOR SQL SERVER 8.0
-- Set option values for user-defined indexes
--
SET @msg = ' '
PRINT @msg
SET @msg = 'Setting SQL Server indexoptions'
PRINT @msg
SET @msg = ' '
PRINT @msg

EXEC sp_indexoption 'customer',      'DisallowPageLocks',      TRUE
EXEC sp_indexoption 'district',     'DisallowPageLocks',      TRUE
EXEC sp_indexoption 'warehouse',    'DisallowPageLocks',      TRUE
EXEC sp_indexoption 'stock',        'DisallowPageLocks',      TRUE
EXEC sp_indexoption 'order_line',   'DisallowRowLocks',      TRUE
EXEC sp_indexoption 'orders',      'DisallowRowLocks',      TRUE
EXEC sp_indexoption 'new_order',   'DisallowRowLocks',      TRUE
EXEC sp_indexoption 'item',        'DisallowRowLocks',      TRUE
EXEC sp_indexoption 'item',        'DisallowPageLocks',      TRUE
GO

Print ' '
Print '*****'
Print 'Pre-specified Locking Hierarchy:'
Print '  Lockflag = 0 ==> No pre-specified hierarchy'
Print '  Lockflag = 1 ==> Lock at Page-level then Table-level'
Print '  Lockflag = 2 ==> Lock at Row-level then Table-level'
Print '  Lockflag = 3 ==> Lock at Table-level'
Print ' '

SELECT name,lockflags
FROM sysindexes
WHERE object_id('warehouse') = id OR
```

```
object_id('district') = id OR
object_id('customer') = id OR
object_id('stock') = id OR
object_id('orders') = id OR
object_id('order_line') = id OR
object_id('history') = id OR
object_id('new_order') = id OR
object_id('item') = id

ORDER BY lockflags asc
GO

sp_configure 'allow updates',0
GO

RECONFIGURE WITH OVERRIDE
GO

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

EXEC sp_tableoption 'district', 'pintable',true
EXEC sp_tableoption 'warehouse', 'pintable',true
EXEC sp_tableoption 'new_order', 'pintable',true
EXEC sp_tableoption 'item', 'pintable',true
GO
```

delivery.sql

```
-- File:      DELIVERY.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates delivery transaction stored procedure
--
--           Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_delivery" )
drop procedure tpcc_delivery
go

create proc tpcc_delivery @w_id          smallint,
                        @o_carrier_id  smallint
as

declare @d_id          tinyint,
        @o_id          int,
        @c_id          int,
        @total         numeric(12,2),
        @oid1          int,
        @oid2          int,
        @oid3          int,
        @oid4          int,
        @oid5          int,
        @oid6          int,
        @oid7          int,
        @oid8          int,
        @oid9          int,
        @oid10         int
```

```

select @d_id = 0
begin tran d
    while (@d_id < 10)
    begin
        select      @d_id = @d_id + 1,
                   @total = 0,
                   @o_id = 0

        select      top 1
                   @o_id = no_o_id
        from        new_order (serializable 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 = getdate(),
                   @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
```

getargs.c

```

// File: GETARGS.C
// Microsoft TPC-C Kit Ver. 4.22
// Copyright Microsoft, 1996, 1997, 1998, 1999,
// 2000, 2001
// Purpose: Source file for command line processing

// Includes
#include "tpcc.h"

//=====
//
// Function name: GetArgsLoader
//=====

void GetArgsLoader(int argc, char **argv, TPCC_LDR_ARGS *pargs)
{
    int i;
    char *ptr;

#ifdef DEBUG
    printf("[%ld]DBG: Entering GetArgsLoader()\n", (int) GetCurrentThreadId());
#endif

    /* init args struct with some useful values */
    pargs->server = SERVER;
    pargs->user = USER;
    pargs->password = PASSWORD;
    pargs->database = DATABASE;
    pargs->batch = BATCH;
    pargs->num_warehouses = UNDEF;
    pargs->tables_all = TRUE;
    pargs->table_item = FALSE;

```

```

pargs->table_warehouse      = FALSE;
pargs->table_customer       = FALSE;
pargs->table_orders        = FALSE;
pargs->loader_res_file      = LOADER_RES_FILE;
pargs->pack_size            = DEFPLDPACKSIZE;
pargs->starting_warehouse  = DEF_STARTING_WAREHOUSE;
pargs->build_index          = BUILD_INDEX;
pargs->index_order          = INDEX_ORDER;
pargs->index_script_path    = INDEX_SCRIPT_PATH;
pargs->scale_down           = SCALE_DOWN;

/* check for zero command line args */
if ( argc == 1 )
    GetArgsLoaderUsage();

for ( i = 1; i < argc; ++i )
{
    if ( argv[i][0] != '-' && argv[i][0] != '/' )
    {
        printf("\nUnrecognized command");
        GetArgsLoaderUsage();
        exit(1);
    }

    ptr = argv[i];

    switch ( ptr[1] )
    {
        case 'h': /* Fall throught */
        case 'H':
            GetArgsLoaderUsage();
            break;

        case 'D':
            pargs->database = ptr+2;
            break;

        case 'P':
            pargs->password = ptr+2;
            break;

        case 'S':
            pargs->server = ptr+2;
            break;

        case 'U':
            pargs->user = ptr+2;
            break;

        case 'b':
            pargs->batch = atol(ptr+2);
            break;

        case 'W':
            pargs->num_warehouses = atol(ptr+2);
            break;

        case 's':
            pargs->starting_warehouse = atol(ptr+2);
            break;

        case 't':
            {

```

```

TRUE;
== 0)
TRUE;
== 0)
TRUE;
0)
TRUE;

pargs->tables_all = FALSE;
if ( strcmp(ptr+2,"item") == 0 )
    pargs->table_item =

else if ( strcmp(ptr+2,"warehouse")
        pargs->table_warehouse =

else if ( strcmp(ptr+2,"customer")
        pargs->table_customer =

else if ( strcmp(ptr+2,"orders") ==
        pargs->table_orders =

else
    {
        printf("\nUnrecognized command");
        GetArgsLoaderUsage();
        exit(1);
    }

    break;

}

case 'f':
    pargs->loader_res_file = ptr+2;
    break;

case 'p':
    pargs->pack_size = atol(ptr+2);
    break;

case 'i':
    pargs->build_index = atol(ptr+2);
    break;

case 'o':
    pargs->index_order = atol(ptr+2);
    break;

case 'c':
    pargs->scale_down = atol(ptr+2);
    break;

case 'd':
    pargs->index_script_path = ptr+2;
    break;

default:
    GetArgsLoaderUsage();
    exit(-1);
    break;

}

}

/* check for required args */
if ( pargs->num_warehouses == UNDEF )
{
    printf("Number of Warehouses is required\n");
    exit(-2);
}

```

```

    return;
}

//=====
//
// Function name: GetArgsLoaderUsage
//
//=====

void GetArgsLoaderUsage()
{
#ifdef DEBUG
    printf("[%ld]DBG: Entering GetArgsLoaderUsage()\n", (int) GetCurrentThreadId());
#endif

    printf("TPCCldr:\n\n");
    printf("Parameter                                     Default\n");
    printf("-----\n");
\n");
    printf("-W Number of Warehouses to Load                Required \n");
    printf("-S Server                                           %s\n", SERVER);
    printf("-U Username                                         %s\n", USER);
    printf("-P Password                                         %s\n", PASSWORD);
    printf("-D Database                                         %s\n", DATABASE);
    printf("-b Batch Size                                       %ld\n",
(long) BATCH);
    printf("-p TDS packet size                               %ld\n",
(long) DEFLDPPACKSIZE);
    printf("-f Loader Results Output Filename               %s\n",
LOADER_RES_FILE);
    printf("-s Starting Warehouse                           %ld\n",
(long) DEF_STARTING_WAREHOUSE);
    printf("-i Build Option (data = 0, data and index = 1)   %ld\n",
(long) BUILD_INDEX);
    printf("-o Cluster Index Build Order (before = 1, after = 0) %ld\n",
(long) INDEX_ORDER);
    printf("-c Build Scaled Database (normal = 0, tiny = 1) %ld\n",
(long) SCALE_DOWN);
    printf("-d Index Script Path                             %s\n",
INDEX_SCRIPT_PATH);
    printf("-t Table to Load                                all tables\n");
\n");
    printf("    [item|warehouse|customer|orders]\n");
    printf("    Notes: \n");
    printf("    - the '-t' parameter may be included multiple times to \n");
    printf("    specify multiple tables to be loaded \n");
    printf("    - 'item' loads ITEM table \n");
    printf("    - 'warehouse' loads WAREHOUSE, DISTRICT, and STOCK tables \n");
    printf("    - 'customer' loads CUSTOMER and HISTORY tables \n");
    printf("    - 'orders' load NEW-ORDER, ORDERS, ORDER-LINE tables \n");

    printf("\nNote: Command line switches are case sensitive.\n");

    exit(0);
}

```

idxcuscl.sql

```

-- File:      IDXCUSCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates clustered index on customer table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'customer_cl' )
    drop index customer.customer_cl

create unique clustered index customer_cl on customer(c_w_id, c_d_id, c_id)
    on MSSQL_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxcusnc.sql

```

-- File:      IDXCUSNC.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates non-clustered index on customer table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'customer_nc1' )
    drop index customer.customer_nc1

create unique nonclustered index customer_nc1 on customer(c_w_id, c_d_id, c_last,
c_first, c_id)
    on MSSQL_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxdiscl.sql

```
-- File:      IDXDISCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates clustered index on district table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'district_c1' )
    drop index district.district_c1

create unique clustered index  district_c1 on district(d_w_id, d_id)
    with fillfactor=100 on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxitmcl.sql

```
-- File:      IDXITMCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates clustered index on item table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'item_c1' )
    drop index item.item_c1

create unique clustered index item_c1 on item(i_id)
    on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxnodcl.sql

```
-- File:      IDXNODCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates clustered index on new_order table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'new_order_c1' )
    drop index new_order.new_order_c1

create unique clustered index new_order_c1 on new_order(no_w_id, no_d_id, no_o_id)
    on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxodlcl.sql

```
-- File:      IDXODLCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates clustered index on order_line table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'order_line_c1' )
    drop index order_line.order_line_c1

create unique clustered index order_line_c1 on order_line(ol_w_id, ol_d_id, ol_o_id,
ol_number)
    on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxordcl.sql

```

-- File:      IDXORDCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates clustered index on orders table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'orders_cl' )
    drop index orders.orders_cl

create unique clustered index orders_cl on orders(o_w_id, o_d_id, o_id)
    on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxordnc.sql

```

-- File:      IDXORDNC.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates non-clustered index on orders table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'orders_ncl' )
    drop index orders.orders_ncl

create index orders_ncl on orders(o_w_id, o_d_id, o_c_id, o_id)
    on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxstkcl.sql

```

-- File:      IDXSTKCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22

```

```

--           Copyright Microsoft, 2001
-- Purpose:   Creates clustered index on stock table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'stock_cl' )
    drop index stock.stock_cl

create unique clustered index stock_cl on stock(s_i_id, s_w_id)
    on MSSQL_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

idxwarcl.sql

```

-- File:      IDXWARCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates clustered index on warehouse table

use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = 'warehouse_cl' )
    drop index warehouse.warehouse_cl

create unique clustered index warehouse_cl on warehouse(w_id)
    with fillfactor=100 on MSSQL_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go

```

neword.sql

```

-- File:      NEWORD.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates new order transaction stored procedure

```

```

--
--      Interface Level: 4.10.000

use tpcc
go

if exists ( select name from sysobjects where name = "tpcc_neworder" )
    drop procedure tpcc_neworder
go

create proc tpcc_neworder

        @w_id          smallint,
        @d_id          tinyint,
        @c_id          int,
        @o_ol_cnt      tinyint,
        @o_all_local   tinyint,
        @i_id1 int = 0, @s_w_id1

smallint = 0, @ol_qty1 smallint = 0,
        @i_id2 int = 0, @s_w_id2

smallint = 0, @ol_qty2 smallint = 0,
        @i_id3 int = 0, @s_w_id3

smallint = 0, @ol_qty3 smallint = 0,
        @i_id4 int = 0, @s_w_id4

smallint = 0, @ol_qty4 smallint = 0,
        @i_id5 int = 0, @s_w_id5

smallint = 0, @ol_qty5 smallint = 0,
        @i_id6 int = 0, @s_w_id6

smallint = 0, @ol_qty6 smallint = 0,
        @i_id7 int = 0, @s_w_id7

smallint = 0, @ol_qty7 smallint = 0,
        @i_id8 int = 0, @s_w_id8

smallint = 0, @ol_qty8 smallint = 0,
        @i_id9 int = 0, @s_w_id9

smallint = 0, @ol_qty9 smallint = 0,
        @i_id10 int = 0, @s_w_id10

smallint = 0, @ol_qty10 smallint = 0,
        @i_id11 int = 0, @s_w_id11

smallint = 0, @ol_qty11 smallint = 0,
        @i_id12 int = 0, @s_w_id12

smallint = 0, @ol_qty12 smallint = 0,
        @i_id13 int = 0, @s_w_id13

smallint = 0, @ol_qty13 smallint = 0,
        @i_id14 int = 0, @s_w_id14

smallint = 0, @ol_qty14 smallint = 0,
        @i_id15 int = 0, @s_w_id15

smallint = 0, @ol_qty15 smallint = 0

as
declare  @w_tax          numeric(4,4),
        @d_tax          numeric(4,4),
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     numeric(4,4),
        @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     smallint,
        @li_qty        smallint,
        @ol_number     int,
        @c_id_local    int

begin

begin transaction n

-- get district tax and next available order id and update
-- plus initialize local variables

        update  district
        set      @d_tax          = d_tax,
                 @o_id          = d_next_o_id,
                 d_next_o_id    = d_next_o_id + 1,
                 @o_entry_d     = getdate(),
                 @li_no         = 0,
                 @commit_flag    = 1
        where    d_w_id          = @w_id and
                 d_id           = @d_id

-- process orderlines

        while (@li_no < @o_ol_cnt)
        begin

                select @li_no = @li_no + 1

-- set i_id, s_w_id, and qty for this lineitem

                select  @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 +
           case when
(s_quantity - @li_qty < 10) then 91 else 0 end,
           s_order_cnt   = s_order_cnt + 1,
           s_remote_cnt  = s_remote_cnt + case when
(@li_s_w_id = @w_id) then 0 else 1 end,
           @s_data       = s_data,
           @s_dist       = case @d_id
        when 1 then s_dist_01
        when 2 then s_dist_02
        when 3 then s_dist_03
        when 4 then s_dist_04
        when 5 then s_dist_05
        when 6 then s_dist_06
        when 7 then s_dist_07
        when 8 then s_dist_08
        when 9 then s_dist_09
        when 10 then s_dist_10
        end
    where  s_i_id        = @li_id and
           s_w_id        = @li_s_w_id

-- if there actually is a stock (and item) with these ids, go to work
    if (@@rowcount > 0)

```

```

        begin
-- insert order_line data (using data from item and stock)
        insert into order_line values(@o_id,
                                       @d_id,
                                       @w_id,
                                       @li_no,
                                       @li_id,
                                       @li_s_w_id,
                                       "dec 31, 1899",
                                       @li_qty,
                                       @i_price *
@li_qty,
                                       @s_dist)

-- send line-item data to client
        select @i_name,
               @s_quantity,
               b_g = case when (
(patindex("%ORIGINAL%",@i_data) > 0) and
(patindex("%ORIGINAL%",@s_data) > 0) )
                then "B" else "G" end,
               @i_price,
               @i_price * @li_qty
        end
        else
        begin
-- no item (or stock) found - triggers rollback condition
        select "",0,"",0,0
        select @commit_flag = 0
        end
-- get customer last name, discount, and credit rating
        select @c_last = c_last,
               @c_discount = c_discount,
               @c_credit = c_credit,
               @c_id_local = c_id
        from   customer (repeatableread)
        where  c_id = @c_id and
               c_w_id = @w_id and
               c_d_id = @d_id

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

```

```

-- insert corresponding row into new-order table
        insert into new_order values (      @o_id,
                                           @d_id,
                                           @w_id)

-- select warehouse tax
        select  @w_tax = w_tax
        from    warehouse (repeatableread)
        where   w_id = @w_id

        if (@commit_flag = 1)
            commit transaction n
        else

-- all that work for nuthin!!!
            rollback transaction n

-- return order data to client
        select  @w_tax,
                @d_tax,
                @o_id,
                @c_last,
                @c_discount,
                @c_credit,
                @o_entry_d,
                @commit_flag

end

go

```

null-txns.sql

```

-- TPC-C Null Txn Stored Procs
-- Microsoft TPC-C Kit
-- 8/17/99
--
-- This script will create stored procs which accept the same parameters and return
correctly formed
-- results sets to match the standard TPC-C stored procs. Of course, the advantage
is that these
-- stored procs place almost no load on SQL Server and do not require a database.
--
-- The purpose of these stored procs is to size and test the web client without the
need of a fully
-- scaled database.
--
drop proc tpcc_delivery
drop proc tpcc_neworder
drop proc tpcc_orderstatus
drop proc tpcc_payment
drop proc tpcc_stocklevel
drop proc tpcc_version
drop table order_line_null
go

create proc tpcc_delivery      @w_id          smallint,

```

```

as
                                           @o_carrier_id  smallint

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

declare @delaytime varchar(30)

-- uniform random delay of 0 - 1 second; avg = 0.50
select @delaytime = '00:00:0' + cast(cast((rand()*1.00) as decimal(4,3)) as char(5))
waitfor delay @delaytime

select 3001, 3001, 3001, 3001, 3001, 3001, 3001, 3001, 3001, 3001

GO

create proc tpcc_neworder

        @w_id          smallint,
        @d_id          tinyint,
        @c_id          int,
        @o_ol_cnt      tinyint,
        @o_all_local   tinyint,
        @i_id1         int = 0, @s_w_id1 smallint

= 0, @ol_qty1 smallint = 0,
        @i_id2         int = 0, @s_w_id2 smallint

= 0, @ol_qty2 smallint = 0,
        @i_id3         int = 0, @s_w_id3 smallint

= 0, @ol_qty3 smallint = 0,
        @i_id4         int = 0, @s_w_id4 smallint

= 0, @ol_qty4 smallint = 0,
        @i_id5         int = 0, @s_w_id5 smallint

= 0, @ol_qty5 smallint = 0,
        @i_id6         int = 0, @s_w_id6 smallint

= 0, @ol_qty6 smallint = 0,
        @i_id7         int = 0, @s_w_id7 smallint

= 0, @ol_qty7 smallint = 0,
        @i_id8         int = 0, @s_w_id8 smallint

= 0, @ol_qty8 smallint = 0,
        @i_id9         int = 0, @s_w_id9 smallint

= 0, @ol_qty9 smallint = 0,
        @i_id10        int = 0, @s_w_id10

smallint = 0, @ol_qty10 smallint = 0,
        @i_id11        int = 0, @s_w_id11

smallint = 0, @ol_qty11 smallint = 0,
        @i_id12        int = 0, @s_w_id12

smallint = 0, @ol_qty12 smallint = 0,
        @i_id13        int = 0, @s_w_id13

smallint = 0, @ol_qty13 smallint = 0,
        @i_id14        int = 0, @s_w_id14

smallint = 0, @ol_qty14 smallint = 0,

```

```

smallint = 0, @ol_qty15 smallint = 0

as
declare @w_tax          numeric(4,4),
        @d_tax         numeric(4,4),
        @c_last        char(16),
        @c_credit      char(2),
        @c_discount    numeric(4,4),
        @i_price       numeric(5,2),
        @i_name        char(24),
        @o_entry_d     datetime,
        @li_no         int,
        @o_id          int,
        @commit_flag   tinyint,
        @li_id         int,
        @li_qty        smallint

declare @delaytime varchar(30)

begin
-- uniform random delay of 0 - 0.6 second; avg = 0.3
select @delaytime = '00:00:0' + cast(cast((rand()*0.60) as decimal(4,3)) as
char(5))
waitfor delay @delaytime

-- process orderlines

select @commit_flag = 1, @li_no = 0

while (@li_no < @o_ol_cnt)
begin
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

select @li_no = @li_no + 1
select @i_price = 23.45, @li_qty = @li_no

if (@li_id = 999999)
begin
select '',0,',',0,0
select @commit_flag = 0
end
else
begin

```

```

select 'Item Name blah',17,'G', @i_price, @i_price * @li_qty
end

end

-- return order data to client

select @w_tax = 0.1234,
       @d_tax = 0.0987,
       @o_id = 3001,
       @c_last = 'BAROUGHTABLE',
       @c_discount = 0.2198,
       @c_credit = 'GC',
       @o_entry_d = getdate()

select @w_tax,
       @d_tax,
       @o_id,
       @c_last,
       @c_discount,
       @c_credit,
       @o_entry_d,
       @commit_flag

end

GO

create proc tpcc_orderstatus @w_id          smallint,
                             @d_id        tinyint,
                             @c_id        int,
                             @c_last     char(16) = ''

as
declare @c_balance          numeric(12,2),
        @c_first           char(16),
        @c_middle          char(2),
        @o_id              int,
        @o_entry_d         datetime,
        @o_carrier_id      smallint,
        @ol_cnt            smallint

declare @delaytime varchar(30)

-- uniform random delay of 0 - 0.2 second; avg = 0.1
select @delaytime = '00:00:0' + cast(cast((rand()*0.20) as decimal(4,3)) as
char(5))
waitfor delay @delaytime

select @c_id = 113,
       @c_balance = -10.00,
       @c_first = '8YCodgytqCj8',
       @c_middle = 'OE',
       @c_last = 'OUGHTTOUGHTABLE',
       @o_id = 3456,
       @o_entry_d = getdate(),
       @o_carrier_id = 1

```

```

select @ol_cnt = (rand() * 11) + 5
SET ROWCOUNT @ol_cnt

select
    ol_supply_w_id,
    ol_i_id,
    ol_quantity,
    ol_amount,
    ol_delivery_d
from order_line_null

select @c_id,
    @c_last,
    @c_first,
    @c_middle,
    @o_entry_d,
    @o_carrier_id,
    @c_balance,
    @o_id

GO

create proc tpcc_payment @w_id          smallint,
                                @c_w_id smallint,
                                @h_amount numeric(6,2),
                                @d_id    tinyint,
                                @c_d_id  tinyint,
                                @c_id    int,
                                @c_last  char(16) = ''

as
declare @w_street_1 char(20),
        @w_street_2 char(20),
        @w_city     char(20),
        @w_state    char(2),
        @w_zip      char(9),
        @w_name     char(10),
        @d_street_1 char(20),
        @d_street_2 char(20),
        @d_city     char(20),
        @d_state    char(2),
        @d_zip      char(9),
        @d_name     char(10),
        @c_first    char(16),
        @c_middle   char(2),
        @c_street_1 char(20),
        @c_street_2 char(20),
        @c_city     char(20),
        @c_state    char(2),
        @c_zip      char(9),
        @c_phone    char(16),
        @c_since    datetime,
        @c_credit   char(2),
        @c_credit_lim numeric(12,2),
        @c_balance  numeric(12,2),
        @c_discount numeric(4,4),

```

```

        @data        char(500),
        @c_data      char(500),
        @datetime    datetime,
        @w_ytd       numeric(12,2),
        @d_ytd       numeric(12,2),
        @cnt         smallint,
        @val         smallint,
        @screen_data char(200),
        @d_id_local  tinyint,
        @w_id_local  smallint,
        @c_id_local  int

declare @delaytime varchar(30)

-- uniform random delay of 0 - 0.3 second; avg = 0.15
select @delaytime = '00:00:0' + cast(cast((rand()*0.30) as decimal(4,3)) as
char(5))
waitfor delay @delaytime

select @screen_data = ''

-- get customer info and update balances

select
    @d_street_1 = 'rqSHHakqyV',
    @d_street_2 = 'zZ98nW3BR2s',
    @d_city     = 'ArNr4GNFV9',
    @d_state    = 'aV',
    @d_zip      = '453511111'

-- get warehouse data and update year-to-date

select
    @w_street_1 = 'rqSHHakqyV',
    @w_street_2 = 'zZ98nW3BR2s',
    @w_city     = 'ArNr4GNFV9',
    @w_state    = 'aV',
    @w_zip      = '453511111'

select
    @c_id       = 123,
    @c_balance  = -10000.00,
    @c_first    = 'Kmr03Xureb',
    @c_middle   = 'OE',
    @c_last     = 'BAROUGHTBAR',
    @c_street_1 = 'QpGdOHjv8mR9vNI8V',
    @c_street_2 = 'dzKcObBqbC3yu',
    @c_city     = 'zAKZXdc037FQxq',
    @c_state    = 'QA',
    @c_zip      = '700311111',
    @c_phone    = '2967264064528555',
    @c_credit   = 'GC',
    @c_credit_lim = 50000.00,
    @c_discount = 0.3069,
    @c_since    = getdate(),
    @datetime   = getdate()

-- 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

create proc tpcc_stocklevel @w_id          smallint,
                           @d_id          tinyint,
                           @threshold     smallint
as
declare @delaytime varchar(30)

-- uniform random delay of 0 - 3.6 second; avg = 1.8
select @delaytime = '00:00:0' + cast(cast((rand()*3.60) as decimal(4,3)) as
char(5))
waitfor delay @delaytime

select 49

GO

create proc tpcc_version
as
declare @version char(8)

begin
select @version = '4.10.000'
select @version as 'Version'
end

GO

CREATE TABLE order_line_null (
[ol_i_id] [int] NOT NULL ,
[ol_supply_w_id] [smallint] NOT NULL ,
[ol_delivery_d] [datetime] NOT NULL ,

```

```

[ol_quantity] [smallint] NOT NULL ,
[ol_amount] [numeric](6, 2) NOT NULL
) ON [PRIMARY]
GO

insert into order_line_null values ( 101, 1, getdate(), 1, 123.45 )
insert into order_line_null values ( 102, 1, getdate(), 2, 123.45 )
insert into order_line_null values ( 103, 1, getdate(), 3, 123.45 )
insert into order_line_null values ( 104, 1, getdate(), 4, 123.45 )
insert into order_line_null values ( 105, 1, getdate(), 5, 123.45 )
insert into order_line_null values ( 106, 1, getdate(), 1, 123.45 )
insert into order_line_null values ( 107, 1, getdate(), 2, 123.45 )
insert into order_line_null values ( 108, 1, getdate(), 3, 123.45 )
insert into order_line_null values ( 109, 1, getdate(), 4, 123.45 )
insert into order_line_null values ( 110, 1, getdate(), 5, 123.45 )
insert into order_line_null values ( 111, 1, getdate(), 1, 123.45 )
insert into order_line_null values ( 112, 1, getdate(), 2, 123.45 )
insert into order_line_null values ( 113, 1, getdate(), 3, 123.45 )
insert into order_line_null values ( 114, 1, getdate(), 4, 123.45 )
insert into order_line_null values ( 115, 1, getdate(), 5, 123.45 )

GO

```

ordstat.sql

```

-- File:      ORDSTAT.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates order status transaction stored procedure
--
--           Interface Level: 4.10.000

use tpcc
go

if exists ( select name from sysobjects where name = "tpcc_orderstatus" )
drop procedure tpcc_orderstatus
go

create proc tpcc_orderstatus @w_id          smallint,
                           @d_id          tinyint,
                           @c_id         int,
                           @c_last       char(16) = ""
as
declare @c_balance         numeric(12,2),
        @c_first          char(16),
        @c_middle         char(2),
        @o_id             int,
        @o_entry_d        datetime,
        @o_carrier_id     smallint,
        @cnt              smallint

begin tran o

if (@c_id = 0)
begin
-- get customer id and info using last name

```



```

select  @cnt      = (count(*)+1)/2
from    customer (repeatableread)
where   c_last    = @c_last and
        c_w_id    = @w_id and
        c_d_id    = @d_id

set     rowcount @cnt

select  @c_id      = c_id,
        @c_balance = c_balance,
        @c_first   = c_first,
        @c_last    = c_last,
        @c_middle  = c_middle
from    customer (repeatableread)
where   c_last    = @c_last and
        c_w_id    = @w_id and
        c_d_id    = @d_id
order   by c_w_id, c_d_id, c_last, c_first

set     rowcount 0

end
else
begin
-- get customer info if by id

select  @c_balance = c_balance,
        @c_first   = c_first,
        @c_middle  = c_middle,
        @c_last    = c_last
from    customer (repeatableread)
where   c_id      = @c_id and
        c_d_id    = @d_id and
        c_w_id    = @w_id

select  @cnt      = @@rowcount

end

-- if no such customer
if (@cnt = 0)
begin
raiserror("Customer not found",18,1)
goto custnotfound
end

-- get order info

select  @o_id      = o_id,
        @o_entry_d = o_entry_d,
        @o_carrier_id = o_carrier_id
from    orders (serializable)
where   o_c_id     = @c_id and
        o_d_id     = @d_id and
        o_w_id     = @w_id
order   by o_id asc

-- select order lines for the current order

select  ol_supply_w_id,

```

```

        ol_i_id,
        ol_quantity,
        ol_amount,
        ol_delivery_d
from    order_line (repeatableread)
where   ol_o_id = @o_id and
        ol_d_id = @d_id and
        ol_w_id = @w_id

```

custnotfound:

commit tran o

-- return data to client

```

select  @c_id,
        @c_last,
        @c_first,
        @c_middle,
        @o_entry_d,
        @o_carrier_id,
        @c_balance,
        @o_id

```

go

payment.sql

```

-- File:      PAYMENT.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates payment transaction stored procedure
--
--           Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_payment" )
drop procedure tpcc_payment
go

create proc tpcc_payment      @w_id          smallint,
                             @c_w_id       smallint,
                             @h_amount     numeric(6,2),
                             @d_id         tinyint,
                             @c_d_id       tinyint,
                             @c_id         int,
                             @c_last       char(16) = ""

as
declare @w_street_1 char(20),
        @w_street_2 char(20),
        @w_city     char(20),
        @w_state    char(2),
        @w_zip      char(9),
        @w_name     char(10),
        @d_street_1 char(20),
        @d_street_2 char(20),
        @d_city     char(20),
        @d_state    char(2),

```

```

@d_zip      char(9),
@d_name     char(10),
@c_first   char(16),
@c_middle  char(2),
@c_street_1 char(20),
@c_street_2 char(20),
@c_city    char(20),
@c_state   char(2),
@c_zip     char(9),
@c_phone   char(16),
@c_since   datetime,
@c_credit  char(2),
@c_credit_lim numeric(12,2),
@c_balance numeric(12,2),
@c_discount numeric(4,4),
@data      char(500),
@c_data    char(500),
@datetime  datetime,
@w_ytd     numeric(12,2),
@d_ytd     numeric(12,2),
@cnt       smallint,
@val       smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local smallint,
@c_id_local int

select @screen_data = ""

begin tran p

-- get payment date

select @datetime = getdate()

if (@c_id = 0)
begin

-- get customer id and info using last name

select @cnt = count(*)
from customer (repeatableread)
where c_last = @c_last and
c_w_id = @c_w_id and
c_d_id = @c_d_id

select @val = (@cnt + 1) / 2
set rowcount @val

select @c_id = c_id
from customer (repeatableread)
where c_last = @c_last and
c_w_id = @c_w_id and
c_d_id = @c_d_id

order by c_last, c_first

set rowcount 0
end

-- get customer info and update balances

update customer

```

```

set @c_balance = c_balance = 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

```

random.c

```

//      File:          RANDOM.C
//
//      Microsoft TPC-C Kit Ver. 4.22
//      Copyright Microsoft, 1996, 1997, 1998, 1999,
//      2000, 2001

```

```

//      Purpose:  Random number generation routines for database loader

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

// Defines
#define A          16807
#define M          2147483647
#define Q          127773 /* M div A */
#define R          2836 /* M mod A */
#define Thread    __declspec(thread)

// Globals
long Thread Seed = 0; /* thread local seed */

/*****
 *
 * random -
 * Implements a GOOD pseudo random number generator. This generator
 * will/should? run the complete period before repeating.
 *
 * Copied from:
 * Random Numbers Generators: Good Ones Are Hard to Find.
 * Communications of the ACM - October 1988 Volume 31 Number 10
 *
 * Machine Dependencies:
 * long must be 2 ^ 31 - 1 or greater.
 *
 *****/

/*****
 * seed - load the Seed value used in irand and drand. Should be used before *
 * first call to irand or drand.
 *****/

void seed(long val)
{
#ifdef DEBUG
    printf("[%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;
}

```

removedb.sql

```

-- File:      REMOVEDB.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Removes tpcc database and backup files

```

```

use master
go

```

```

-- remove any existing database and backup files

```

```

exec sp_dbrremove tpcc, dropdev
go

```

```

exec sp_dropdevice 'tpccback48001'
exec sp_dropdevice 'tpccback48002'
go

```

restore.sql

```

-- File:      RESTORE.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Loads database backup from backup files

```

```

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

```

```

load database tpcc from tpccback48001, tpccback48002 with stats = 1, replace

```

```

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

```

```

go

```

```

sp_dboption 'tpcc', 'torn page', false
go

```

sqlshutdown.sql

```

use tpcc
go
checkpoint
go
shutdown
go

```

stocklev.sql

```

-- File:      STOCKLEV.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- Purpose:   Creates stock level transaction stored procedure
--
--           Interface Level: 4.10.000

```

```

use tpcc
go

```

```

if exists (select name from sysobjects where name = "tpcc_stocklevel" )
    drop procedure tpcc_stocklevel

```

```

go

```

```

create proc tpcc_stocklevel    @w_id          smallint,
                               @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

```

```

go

```

strings.c

```

//           File:      STRINGS.C
//           Microsoft TPC-C Kit Ver. 4.22
//           Copyright Microsoft, 1996, 1997, 1998, 1999,
//           2000, 2001
//           Purpose:   Source file for database loader string functions

```

```

// Includes
#include "tpcc.h"
#include <string.h>
#include <ctype.h>

//=====
//
// Function name: MakeAddress
//
//=====
void MakeAddress(char *street_1,
                char *street_2,
                char *city,
                char *state,
                char *zip)
{
#ifdef DEBUG
printf("[%ld]DBG: Entering MakeAddress()\n", (int) GetCurrentThreadId());
#endif

MakeAlphaString (10, 20, ADDRESS_LEN, street_1);
MakeAlphaString (10, 20, ADDRESS_LEN, street_2);
MakeAlphaString (10, 20, ADDRESS_LEN, city);
MakeAlphaString ( 2,  2, STATE_LEN, state);
MakeZipNumberString( 9,  9, ZIP_LEN, zip);

#ifdef DEBUG
printf("[%ld]DBG: MakeAddress: street_1: %s, street_2: %s, city: %s, state: %s,
zip: %s\n",
(int) GetCurrentThreadId(), street_1, street_2, city,
state, zip);
#endif

return;
}

//=====
//
// Function name: LastName
//
//=====
void LastName(int num,
             char *name)
{
static char *n[] =
{
"BAR" , "OUGHT" , "ABLE" , "PRI" , "PRES",
"ESE" , "ANTI" , "CALLY" , "ATION" , "EING"
};

#ifdef DEBUG
printf("[%ld]DBG: Entering LastName()\n", (int) GetCurrentThreadId());
#endif

if ((num >= 0) && (num < 1000))
{

```

```

strcpy(name, n[(num/100)%10]);
strcat(name, n[(num/10)%10]);
strcat(name, n[(num/1)%10]);

if (strlen(name) < LAST_NAME_LEN)
{
PaddString(LAST_NAME_LEN, name);
}
}
else
{
printf("\nError in LastName()... num <%ld> out of range
(0,999)\n", num);
exit(-1);
}

#ifdef DEBUG
printf("[%ld]DBG: LastName: num = [%d] ==> [%d][%d][%d]\n",
(int) GetCurrentThreadId(), num, num/100, (num/10)%10,
num%10);
printf("[%ld]DBG: LastName: String = %s\n", (int) GetCurrentThreadId(),
name);
#endif

return;
}

//=====
//
// Function name: MakeAlphaString
//
//=====
//philipdu 08/13/96 Changed MakeAlphaString to use A-Z, a-z, and 0-9 in
//accordance with spec see below:
//The spec says:
//4.3.2.2 The notation random a-string [x .. y]
//(respectively, n-string [x .. y]) represents a string of random alphanumeric
//(respectively, numeric) characters of a random length of minimum x, maximum y,
//and mean (y+x)/2. Alphanumerics are A..Z, a..z, and 0..9. The only other
//requirement is that the character set used "must be able to represent a minimum
//of 128 different characters". We are using 8-bit chars, so this is a non issue.
//It is completely unreasonable to stuff non-printing chars into the text fields.
//-CLevine 08/13/96

int MakeAlphaString( int x, int y, int z, char *str)
{
int len;
int i;
char cc = 'a';
static char chArray[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
static int chArrayMax = 61;

#ifdef DEBUG
printf("[%ld]DBG: Entering MakeAlphaString()\n", (int) GetCurrentThreadId());
#endif

len= RandomNumber(x, y);

```

```

        for (i=0; i<len; i++)
        {
            cc = chArray[RandomNumber(0, chArrayMax)];
            str[i] = cc;
        }
        if ( len < z )
            memset(str+len, ' ', z - len);
        str[len] = 0;

    return len;
}

//=====
//
// Function name: MakeOriginalAlphaString
//
//=====
int MakeOriginalAlphaString(int x,
                           int y,
                           int z,
                           char *str,
                           int percent)
{
    int len;
    int val;
    int start;

#ifdef DEBUG
    printf("[%ld]DBG: Entering MakeOriginalAlphaString()\n", (int)
GetCurrentThreadId());
#endif

    // verify 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;
}

```

tables.sql

```

-- File: TABLES.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.22
-- Copyright Microsoft, 2001
-- Purpose: Creates TPC-C tables

use tpcc
go

--
-- Remove all existing TPC-C tables

```

```

--
if exists ( select name from sysobjects where name = 'warehouse' )
    drop table warehouse
go
if exists ( select name from sysobjects where name = 'district' )
    drop table district
go
if exists ( select name from sysobjects where name = 'customer' )
    drop table customer
go
if exists ( select name from sysobjects where name = 'history' )
    drop table history
go
if exists ( select name from sysobjects where name = 'new_order' )
    drop table new_order
go
if exists ( select name from sysobjects where name = 'orders' )
    drop table orders
go
if exists ( select name from sysobjects where name = 'order_line' )
    drop table order_line
go
if exists ( select name from sysobjects where name = 'item' )
    drop table item
go
if exists ( select name from sysobjects where name = 'stock' )
    drop table stock
go

--
-- Create new tables
--
create table warehouse
(
    w_id                smallint,
    w_name              char(10),
    w_street_1          char(20),
    w_street_2          char(20),
    w_city              char(20),
    w_state             char(2),
    w_zip              char(9),
    w_tax              numeric(4,4),
    w_ytd              numeric(12,2)
) on MSSQL_misc_fg
go

create table district
(
    d_id                tinyint,
    d_w_id              smallint,
    d_name              char(10),
    d_street_1          char(20),
    d_street_2          char(20),
    d_city              char(20),
    d_state             char(2),
    d_zip              char(9),
    d_tax              numeric(4,4),
    d_ytd              numeric(12,2),
    d_next_o_id         int
) on MSSQL_misc_fg
go

```



```

create table customer
(
    c_id int,
    c_d_id tinyint,
    c_w_id smallint,
    c_first char(16),
    c_middle char(2),
    c_last char(16),
    c_street_1 char(20),
    c_street_2 char(20),
    c_city char(20),
    c_state char(2),
    c_zip char(9),
    c_phone char(16),
    c_since datetime,
    c_credit char(2),
    c_credit_lim numeric(12,2),
    c_discount numeric(4,4),
    c_balance numeric(12,2),
    c_ytd_payment numeric(12,2),
    c_payment_cnt smallint,
    c_delivery_cnt smallint,
    c_data char(500)
) on MSSQL_cs_fg
go

create table history
(
    h_c_id int,
    h_c_d_id tinyint,
    h_c_w_id smallint,
    h_d_id tinyint,
    h_w_id smallint,
    h_date datetime,
    h_amount numeric(6,2),
    h_data char(24)
) on MSSQL_misc_fg
go

create table new_order
(
    no_o_id int,
    no_d_id tinyint,
    no_w_id smallint
) on MSSQL_misc_fg
go

create table orders
(
    o_id int,
    o_d_id tinyint,
    o_w_id smallint,
    o_c_id int,
    o_entry_d datetime,
    o_carrier_id tinyint,
    o_ol_cnt tinyint,
    o_all_local tinyint
) on MSSQL_misc_fg
go

create table order_line
(

```

```

    ol_o_id int,
    ol_d_id tinyint,
    ol_w_id smallint,
    ol_number tinyint,
    ol_i_id int,
    ol_supply_w_id smallint,
    ol_delivery_d datetime,
    ol_quantity smallint,
    ol_amount numeric(6,2),
    ol_dist_info char(24)
) on MSSQL_misc_fg
go

create table item
(
    i_id int,
    i_im_id int,
    i_name char(24),
    i_price numeric(5,2),
    i_data char(50)
) on MSSQL_misc_fg
go

create table stock
(
    s_i_id int,
    s_w_id smallint,
    s_quantity smallint,
    s_dist_01 char(24),
    s_dist_02 char(24),
    s_dist_03 char(24),
    s_dist_04 char(24),
    s_dist_05 char(24),
    s_dist_06 char(24),
    s_dist_07 char(24),
    s_dist_08 char(24),
    s_dist_09 char(24),
    s_dist_10 char(24),
    s_ytd int,
    s_order_cnt smallint,
    s_remote_cnt smallint,
    s_data char(50)
) on MSSQL_cs_fg
go

```

time.c

```

// File: TIME.C
// Microsoft TPC-C Kit Ver. 4.22
// Copyright Microsoft, 1996, 1997, 1998, 1999,
// 2000, 2001
// Purpose: Source file for time functions

// Includes
#include "tpcc.h"

// Globals
static long start_sec;

//=====

```

```

//
// Function name: TimeNow
//
//=====
long TimeNow()
{
    long            time_now;
    struct _timeb  el_time;

#ifdef DEBUG
    printf("[%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;
}

```

tpcc.h

```

// File:                TPCC.H
//                      Microsoft TPC-C Kit Ver. 4.22
//                      Copyright Microsoft, 1996, 1997, 1998, 1999,
//                      2000, 2001
// Purpose: Header file for TPC-C database loader

// Build number of TPC Benchmark Kit
#define TPCKIT_VER "4.22"

// General headers
#include <windows.h>
#include <winbase.h>
#include <stdlib.h>
#include <stdio.h>
#include <process.h>
#include <stddef.h>
#include <stdarg.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <sys\types.h>

// ODBC headers
#include <sql.h>
#include <sqlext.h>
#include <odbcss.h>

// General constants
#define MILLI                1000
#define FALSE                0
#define TRUE                 1
#define UNDEF                -1
#define MINPRINTASCII       32
#define MAXPRINTASCII       126

// Default environment constants

```

```

#define SERVER                ""
#define DATABASE              "tpcc"
#define USER                  "sa"
#define PASSWORD              ""

// Default loader arguments
#define BATCH                  10000
#define DEFLDPACKSIZE        32768
#define LOADER_RES_FILE      "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;
} TPCCCLR_ARGS;

// String length constants
#define SERVER_NAME_LEN      20
#define DATABASE_NAME_LEN   20
#define USER_NAME_LEN       20
#define PASSWORD_LEN        20
#define TABLE_NAME_LEN     20
#define I_DATA_LEN          50
#define I_NAME_LEN          24
#define BRAND_LEN           1
#define LAST_NAME_LEN       16
#define W_NAME_LEN          10
#define ADDRESS_LEN         20
#define STATE_LEN           2
#define ZIP_LEN              9

```

```

#define S_DIST_LEN          24
#define S_DATA_LEN         50
#define D_NAME_LEN         10
#define FIRST_NAME_LEN     16
#define MIDDLE_NAME_LEN   2
#define PHONE_LEN          16
#define CREDIT_LEN         2
#define C_DATA_LEN         500
#define H_DATA_LEN         24
#define DIST_INFO_LEN      24
#define MAX_OL_NEW_ORDER_ITEMS 15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN         25
#define OL_DIST_INFO_LEN   24
#define C_SINCE_LEN        23
#define H_DATE_LEN         23
#define OL_DELIVERY_D_LEN  23
#define O_ENTRY_D_LEN      23

```

```

// Functions in random.c
void seed();
long irand();
double drand();
void WUcreate();
short WURand();
long RandomNumber(long lower, long upper);

```

```

// Functions in getargs.c;
void GetArgsLoader();
void GetArgsLoaderUsage();

```

```

// Functions in time.c
long TimeNow();

```

```

// Functions in strings.c
void MakeAddress();
void LastName();
int MakeAlphaString();
int MakeOriginalAlphaString();
int MakeNumberString();
int MakeZipNumberString();
void InitString();
void InitAddress();
void PaddString();

```

tpccldr.c

```

// File: TPCCLDR.C
// Microsoft TPC-C Kit Ver. 4.22
// Copyright Microsoft, 2000, 2001
// Purpose: Source file for TPC-C database loader

```

```

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

```

```

// Defines
#define MAXITEMS          100000
#define MAXITEMS_SCALE_DOWN 100
#define CUSTOMERS_PER_DISTRICT 3000

```

```

#define CUSTOMERS_SCALE_DOWN 30
#define DISTRICT_PER_WAREHOUSE 10
#define ORDERS_PER_DISTRICT 3000
#define ORDERS_SCALE_DOWN 30
#define MAX_CUSTOMER_THREADS 2
#define MAX_ORDER_THREADS 3
#define MAX_MAIN_THREADS 4

```

```

// Functions declarations

```

```

void HandleErrorDBC (SQLHDBC hdbc1);

```

```

void ChecksQL();
void CheckDataBase();

```

```

long NURand();
void LoadItem();
void LoadWarehouse();

```

```

void Stock();
void District();

```

```

void LoadCustomer();
void CustomerBufInit();
void CustomerBufLoad();
void LoadCustomerTable();
void LoadHistoryTable();

```

```

void LoadOrders();
void OrdersBufInit();
void OrdersBufLoad();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void GetPermutation();
void CheckForCommit();
void OpenConnections();
void BuildIndex();
void FormatDate ();

```

```

// Shared memory structures

```

```

typedef struct
{
    long ol;
    long ol_i_id;
    short ol_supply_w_id;
    short ol_quantity;
    double ol_amount;
    char ol_dist_info[DIST_INFO_LEN+1];
    char ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCT;

```

```

typedef struct
{
    long o_id;
    short o_d_id;
    short o_w_id;
    long o_c_id;
    short o_carrier_id;
    short o_ol_cnt;
    short o_all_local;
}

```

```

        ORDER_LINE_STRUCT  o_ol[15];
    } ORDERS_STRUCT;

typedef struct
{
    long          c_id;
    short         c_d_id;
    short         c_w_id;
    char          c_first[FIRST_NAME_LEN+1];
    char          c_middle[MIDDLE_NAME_LEN+1];
    char          c_last[LAST_NAME_LEN+1];
    char          c_street_1[ADDRESS_LEN+1];
    char          c_street_2[ADDRESS_LEN+1];
    char          c_city[ADDRESS_LEN+1];
    char          c_state[STATE_LEN+1];
    char          c_zip[ZIP_LEN+1];
    char          c_phone[PHONE_LEN+1];
    char          c_credit[CREDIT_LEN+1];
    double        c_credit_lim;
    double        c_discount;
    // fix to avoid ODBC float to numeric conversion problem.
    // double      c_balance;
    char          c_balance[6];

    double        c_ytd_payment;
    short         c_payment_cnt;
    short         c_delivery_cnt;
    char          c_data[C_DATA_LEN+1];
    double        h_amount;
    char          h_data[H_DATA_LEN+1];
} CUSTOMER_STRUCT;

typedef struct
{
    char          c_last[LAST_NAME_LEN+1];
    char          c_first[FIRST_NAME_LEN+1];
    long          c_id;
} CUSTOMER_SORT_STRUCT;

typedef struct
{
    long          time_start;
} LOADER_TIME_STRUCT;

// Global variables
char            szLastError[300];
HENV            henv;
HDBC            v_hdbc; // for SQL
Server version verification
HDBC            i_hdbc1; // for ITEM table
HDBC            w_hdbc1; // for WAREHOUSE,
DISTRICT, STOCK
HDBC            c_hdbc1; // for CUSTOMER
HDBC            c_hdbc2; // for HISTORY
HDBC            o_hdbc1; // for ORDERS
HDBC            o_hdbc2; // for NEW-ORDER
HDBC            o_hdbc3; // for ORDER-LINE

```

```

HSTMT            v_hstmt; // for SQL Server
version verification
HSTMT            i_hstmt1;
HSTMT            w_hstmt1;
HSTMT            c_hstmt1, c_hstmt2;
HSTMT            o_hstmt1, o_hstmt2, o_hstmt3;

ORDERS_STRUCT    orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT  customer_buf[CUSTOMERS_PER_DISTRICT];
long             orders_rows_loaded;
long             new_order_rows_loaded;
long             order_line_rows_loaded;
long             history_rows_loaded;
long             customer_rows_loaded;
long             stock_rows_loaded;
long             district_rows_loaded;
long             item_rows_loaded;
long             warehouse_rows_loaded;
long             main_time_start;
long             main_time_end;
long             max_items;
long             customers_per_district;
long             orders_per_district;
long             first_new_order;
long             last_new_order;

TPCCCLDR_ARGS    *aptr, args;

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

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

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

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

    // process command line arguments
    aptr = &args;
    GetArgsLoader(argc, argv, aptr);

    // verify database and tables exist before attempting to load

```

```

CheckSQL();
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         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, "item");

rc = bcp_init(i_hdbc1, name, NULL, "logs\\item.err", DB_IN);
if (rc != SUCCEEDED)
    HandleErrorDBC(i_hdbc1);

if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    sprintf(bcphint, "tablock, order (i_id), ROWS_PER_BATCH =
100000");
    rc = bcp_control(i_hdbc1, BCPHINTS, (void*) bcphint);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);
}

rc = bcp_bind(i_hdbc1, (BYTE *) &i_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
if (rc != SUCCEEDED)
    HandleErrorDBC(i_hdbc1);

rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 2);
if (rc != SUCCEEDED)
    HandleErrorDBC(i_hdbc1);

rc = bcp_bind(i_hdbc1, (BYTE *) i_name, 0, I_NAME_LEN, NULL, 0, 0, 3);
if (rc != SUCCEEDED)
    HandleErrorDBC(i_hdbc1);

rc = bcp_bind(i_hdbc1, (BYTE *) &i_price, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 4);
if (rc != SUCCEEDED)
    HandleErrorDBC(i_hdbc1);

rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0, I_DATA_LEN, NULL, 0, 0, 5);
if (rc != SUCCEEDED)
    HandleErrorDBC(i_hdbc1);

time_start = (TimeNow() / MILLI);

item_rows_loaded = 0;

for (i_id = 1; i_id <= max_items; i_id++)

```

```

    {
        i_im_id = RandomNumber(1L, 10000L);

        MakeAlphaString(14, 24, I_NAME_LEN, i_name);

        i_price = ((float) RandomNumber(100L, 10000L))/100.0;

        MakeOriginalAlphaString(26, 50, I_DATA_LEN, i_data, 10);

        rc = bcp_sendrow(i_hdbc1);
        if (rc != SUCCEED)
            HandleErrorDBC(i_hdbc1);

        item_rows_loaded++;
        CheckForCommit(i_hdbc1, i_hstmt1, item_rows_loaded, "item",
&time_start);
    }

    rcint = bcp_done(i_hdbc1);
    if (rcint < 0)
        HandleErrorDBC(i_hdbc1);

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

    SQLFreeStmt(i_hstmt1, SQL_DROP);
    SQLDisconnect(i_hdbc1);
    SQLFreeConnect(i_hdbc1);

    // if build index after load
    if ((aptr->build_index == 1) && (aptr->index_order == 0))
        BuildIndex("idxitmcl");
}

//=====
//
// Function   : LoadWarehouse
//
// Loads WAREHOUSE table and loads Stock and District as Warehouses are created
//
//=====

void LoadWarehouse()
{
    short w_id;
    char w_name[W_NAME_LEN+1];
    char w_street_1[ADDRESS_LEN+1];
    char w_street_2[ADDRESS_LEN+1];
    char w_city[ADDRESS_LEN+1];
    char w_state[STATE_LEN+1];
    char w_zip[ZIP_LEN+1];
    double w_tax;
    double w_ytd;
    char name[20];
    long time_start;
    RETCODE rc;
    DBINT rcint;
    char bcp[128];

    // Seed with unique number
    seed(2);

```

```

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

    // if build index before load...
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        BuildIndex("idxwarc1");

    InitString(w_name, W_NAME_LEN+1);
    InitAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

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

    rc = bcp_init(w_hdbc1, name, NULL, "logs\\whouse.err", DB_IN);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcp, "tablock, order (w_id), ROWS_PER_BATCH = %d",
aptr->num_warehouses);
        rc = bcp_control(w_hdbc1, BCP_HINTS, (void*) bcp);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
    }

    rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT2, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

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

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

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

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

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

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

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

    rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 9);

```

```

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

time_start = (TimeNow() / MILLI);
warehouse_rows_loaded = 0;

w_id++)
for (w_id = (short)aptr->starting_warehouse; w_id <= aptr->num_warehouses;
{
    MakeAlphaString(6,10, W_NAME_LEN, w_name);
    MakeAddress(w_street_1, w_street_2, w_city, w_state, w_zip);
    w_tax = ((float) RandomNumber(0L,2000L))/10000.00;
    w_ytd = 300000.00;

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

    warehouse_rows_loaded++;
    CheckForCommit(w_hdbc1, i_hstmt1, warehouse_rows_loaded,
"warehouse", &time_start);
}

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

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

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

stock_rows_loaded = 0;
district_rows_loaded = 0;

District();
Stock();
}

//=====
//
// Function   : District
//
//=====

void District()
{
    short d_id;
    short d_w_id;
    char  d_name[D_NAME_LEN+1];
    char  d_street_1[ADDRESS_LEN+1];
    char  d_street_2[ADDRESS_LEN+1];
    char  d_city[ADDRESS_LEN+1];
    char  d_state[STATE_LEN+1];
    char  d_zip[ZIP_LEN+1];
    double d_tax;

```

```

double      d_ytd;
char        name[20];
long        d_next_o_id;
long        time_start;
int         w_id;
RETCODE    rc;
DBINT      rcint;
char        bcphint[128];

// Seed with unique number
seed(4);

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

// build index before load
if ((aptr->build_index == 1) && (aptr->index_order == 1))
    BuildIndex("idxdiscl");

InitString(d_name, D_NAME_LEN+1);
InitAddress(d_street_1, d_street_2, d_city, d_state, d_zip);
sprintf(name, "%s.%s", aptr->database, "district");

rc = bcp_init(w_hdbc1, name, NULL, "logs\\district.err", DB_IN);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

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

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

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

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

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

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

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

d_ytd = 30000.0;

d_next_o_id = orders_per_district+1;

time_start = (TimeNow() / MILLI);

for (w_id = aptr->starting_warehouse; w_id <= aptr->num_warehouses;
w_id++)
{
    d_w_id = w_id;

    for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
    {
        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;
    short s_w_id;
    short s_quantity;
    char  s_dist_01[S_DIST_LEN+1];
    char  s_dist_02[S_DIST_LEN+1];
    char  s_dist_03[S_DIST_LEN+1];
    char  s_dist_04[S_DIST_LEN+1];
    char  s_dist_05[S_DIST_LEN+1];
    char  s_dist_06[S_DIST_LEN+1];
    char  s_dist_07[S_DIST_LEN+1];
    char  s_dist_08[S_DIST_LEN+1];
    char  s_dist_09[S_DIST_LEN+1];
    char  s_dist_10[S_DIST_LEN+1];
    long  s_ytd;
    short s_order_cnt;
    short s_remote_cnt;
    char  s_data[S_DATA_LEN+1];
    short len;
    char  name[20];
    long  time_start;
    RETCODE rc;
    DBINT rcint;
    char  bcphint[128];

    // Seed with unique number
    seed(3);

    // if build index before load...
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        BuildIndex("idxstkcl");

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

    rc = bcp_init(w_hdbc1, name, NULL, "logs\\stock.err", DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);

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

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

```

```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

```

        HandleErrorDBC(w_hdbc1);

    rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0, S_DATA_LEN, NULL, 0, 0, 17);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    s_ytd = s_order_cnt = s_remote_cnt = 0;

    time_start = (TimeNow() / MILLI);

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

    for (s_i_id=1; s_i_id <= max_items; s_i_id++)
    {
        for (s_w_id = (short)aptr->starting_warehouse; s_w_id <= aptr-
>num_warehouses; s_w_id++)
        {
            s_quantity = (short)RandomNumber(10L,100L);
            len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_01);
            len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_02);
            len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_03);
            len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_04);
            len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_05);
            len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_06);
            len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_07);
            len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_08);
            len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_09);
            len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_10);

            len = MakeOriginalAlphaString(26,50, S_DATA_LEN,
s_data,10);

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

            stock_rows_loaded++;
            CheckForCommit(w_hdbc1, w_hstmt1, stock_rows_loaded,
"stock", &time_start);
        }
    }

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

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

    SQLFreeStmt(w_hstmt1, SQL_DROP);
    SQLDisconnect(w_hdbc1);
    SQLFreeConnect(w_hdbc1);

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

    return;
}

```

```

//=====
//
// Function   : LoadCustomer
//
//=====
void LoadCustomer()
{
    LOADER_TIME_STRUCT    customer_time_start;
    LOADER_TIME_STRUCT    history_time_start;
    short                 w_id;
    short                 d_id;
    DWORD                 dwThreadID[MAX_CUSTOMER_THREADS];
    HANDLE                 hThread[MAX_CUSTOMER_THREADS];
    char                   name[20];
    RETCODE                rc;
    DBINT                  rcint;
    char                   bcphint[128];
    char                   cmd[256];
    char                   rc_l;
    // SQLRETURN
    // SQLSMALLINT
    // SQLCHAR
    // SQLINTEGER
    Msg[SQL_MAX_MESSAGE_LENGTH];
    // SQLINTEGER
    NativeError;

    // Seed with unique number
    seed(5);

    printf("Loading customer and history tables...\n");

    // if build index before load...
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        BuildIndex("idxcuscl");

    // Initialize bulk copy
    sprintf(name, "%s.%s", aptr->database, "customer");

    rc = bcp_init(c_hdbc1, name, NULL, "logs\\customer.err", DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcphint, "tablock, order (c_w_id, c_d_id, c_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
        rc = bcp_control(c_hdbc1, BCPHINTS, (void*) bcphint);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
    }

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

    rc = bcp_init(c_hdbc2, name, NULL, "logs\\history.err", DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc2);

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

    customer_rows_loaded = 0;
    history_rows_loaded = 0;
}

```

```

CustomerBufInit();

customer_time_start.time_start = (TimeNow() / MILLI);
history_time_start.time_start = (TimeNow() / MILLI);

for (w_id = (short)aptr->starting_warehouse; w_id <= aptr->num_warehouses;
w_id++)
{
    for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
    {
        CustomerBufLoad(d_id, w_id);

        // Start parallel loading threads here...

        // Start customer table thread
        printf("...Loading customer table for: d_id = %d, w_id
= %d\n", d_id, w_id);

        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("idxcusc1");

// build non-clustered index
if (aptr->build_index == 1)
    BuildIndex("idxcuscnc");

// Output the NURAND used for the loader into C_FIRST for C_ID = 1,
// C_W_ID = 1, and C_D_ID = 1
sprintf(cmd, "isql -S%s -U%s -P%s -d%s -e -Q\"update customer set c_first
= 'C_LOAD = %d' where c_id = 1 and c_w_id = 1 and c_d_id = 1\" >
logs\\nurand_load.log",

        aptr->server,
        aptr->user,
        aptr->password,
        aptr->database,
        LOADER_NURAND_C);

system(cmd);

SQLFreeStmt(c_hstmt1, SQL_DROP);
SQLDisconnect(c_hdbc1);
SQLFreeConnect(c_hdbc1);

SQLFreeStmt(c_hstmt2, SQL_DROP);
SQLDisconnect(c_hdbc2);
SQLFreeConnect(c_hdbc2);

return;

```

```

}

//=====
//
// Function : CustomerBufInit
//
//=====

void CustomerBufInit()
{
    int i;

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

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

        customer_buf[i].c_credit_lim = 0;
        customer_buf[i].c_discount = (float) 0;

        // fix to avoid ODBC float to numeric conversion 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;
    short c_w_id;
    char c_first[FIRST_NAME_LEN+1];
    char c_middle[MIDDLE_NAME_LEN+1];
    char c_last[LAST_NAME_LEN+1];
    char c_street_1[ADDRESS_LEN+1];
    char c_street_2[ADDRESS_LEN+1];
    char c_city[ADDRESS_LEN+1];
    char c_state[STATE_LEN+1];
    char c_zip[ZIP_LEN+1];
    char c_phone[PHONE_LEN+1];
    char c_credit[CREDIT_LEN+1];
    double c_credit_lim;
    double c_discount;

    // fix to avoid ODBC float to numeric conversion problem.
    // double c_balance;
    char c_balance[6];

    double c_ytd_payment;
    short c_payment_cnt;
    short c_delivery_cnt;
    char c_data[C_DATA_LEN+1];

    char c_since[C_SINCE_LEN+1];
    RETCODE rc;

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 3);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0, FIRST_NAME_LEN, NULL, 0, 0, 4);

```

```

    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0, MIDDLE_NAME_LEN, NULL, 0, 0, 5);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN, NULL, 0, 0, 6);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN, NULL, 0, 0, 7);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN, NULL, 0, 0, 8);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL, 0, 0, 9);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0, 0, 10);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0, 11);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL, 0, 0, 12);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0, C_SINCE_LEN, NULL, 0,
SQLCHARACTER, 13);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL, 0, 0, 14);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 15);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

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

    // fix to avoid ODBC float to numeric conversion problem.

    // rc = bcp_bind(c_hdbc1, (BYTE *) &c_balance, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 17);
    // if (rc != SUCCEEDED)
    //     HandleErrorDBC(c_hdbc1);
rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0, SQLCHARACTER, 17);
    if (rc != SUCCEEDED)

```

```

        HandleErrorDBC(c_hdbc1);

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

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

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

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

for (i = 0; i < customers_per_district; i++)
{
    c_id = customer_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 != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    customer_rows_loaded++;

```

```

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

//=====
//
// Function   : LoadHistoryTable
//
//=====

void LoadHistoryTable(LOADER_TIME_STRUCT *history_time_start)
{
    int         i;
    long        c_id;
    short       c_d_id;
    short       c_w_id;
    double      h_amount;
    char        h_data[H_DATA_LEN+1];
    char        h_date[H_DATE_LEN+1];
    RETCODE     rc;

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
    if (rc != 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, SQLINT2,
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, SQLINT2,
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;
    short               w_id;
    short               d_id;
    DWORD               dwThreadID[MAX_ORDER_THREADS];
    HANDLE              hThread[MAX_ORDER_THREADS];
    char                name[20];
    RETCODE             rc;
    char                bcpHint[128];

    // seed with unique number
    seed(6);

    printf("Loading orders...\n");

    // if build index before load..
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        BuildIndex("idxordc1");
        BuildIndex("idxnodc1");
        BuildIndex("idxodc1");
    }

    // initialize bulk copy
    sprintf(name, "%s..%s", aptr->database, "orders");

    rc = bcp_init(o_hdbc1, name, NULL, "logs\\orders.err", DB_IN);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {

```

```

        sprintf(bcphint, "tablock, order (o_w_id, o_d_id, o_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
rc = bcp_control(o_hdbc1, BCPHINTS, (void*) bcphint);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc1);
    }

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

rc = bcp_init(o_hdbc2, name, NULL, "logs\\neword.err", DB_IN);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc2);

if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    sprintf(bcphint, "tablock, order (no_w_id, no_d_id, no_o_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 9000));
rc = bcp_control(o_hdbc2, BCPHINTS, (void*) bcphint);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc2);
}

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

rc = bcp_init(o_hdbc3, name, NULL, "logs\\ordline.err", DB_IN);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc3);

if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    sprintf(bcphint, "tablock, order (ol_w_id, ol_d_id, ol_o_id,
ol_number), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 300000));
rc = bcp_control(o_hdbc3, BCPHINTS, (void*) bcphint);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc3);
}

orders_rows_loaded      = 0;
new_order_rows_loaded  = 0;
order_line_rows_loaded  = 0;

OrdersBufInit();

orders_time_start.time_start = (TimeNow() / MILLI);
new_order_time_start.time_start = (TimeNow() / MILLI);
order_line_time_start.time_start = (TimeNow() / MILLI);

for (w_id = (short)aptr->starting_warehouse; w_id <= aptr->num_warehouses;
w_id++)
{
    for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
    {
        OrdersBufLoad(d_id, w_id);

        // start parallel loading threads here...

        // start Orders table thread
        printf("...Loading Order Table for: d_id = %d, w_id =
%d\n", d_id, w_id);

        hThread[0] = CreateThread(NULL,

```

```

0,
(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;
    short       o_w_id;
    long        o_c_id;
    short       o_carrier_id;
    short       o_ol_cnt;
    short       o_all_local;
    char        o_entry_d[O_ENTRY_D_LEN+1];
    RETCODE     rc;
    DBINT       rcint;

    // bind ORDER data
rc = bcp_bind(o_hdbc1, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc1);

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

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

```

```

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

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

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

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

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

for (i = 0; i < orders_per_district; i++)
{
    o_id         = orders_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("idxordc1");

        // build non-clustered index
        if (aptr->build_index == 1)
            BuildIndex("idxordnc");
    }

}

//=====
//
// Function   : LoadNewOrderTable
//
//=====

void LoadNewOrderTable(LOADER_TIME_STRUCT *new_order_time_start)
{
    int         i;
    long        o_id;
    short       o_d_id;
    short       o_w_id;
    RETCODE     rc;
    DBINT       rcint;

    // Bind NEW-ORDER data

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

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

    for (i = first_new_order; i < last_new_order; i++)
    {
        o_id   = orders_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("idxmodc1");
    }

}

//=====
//
// Function   : LoadOrderLineTable
//
//=====

void LoadOrderLineTable(LOADER_TIME_STRUCT *order_line_time_start)
{
    int         i, j;
    long        o_id;
    short       o_d_id;
    short       o_w_id;
    long        ol;
    long        ol_i_id;
    short       ol_supply_w_id;
    short       ol_quantity;
    double      ol_amount;
    char        ol_dist_info[DIST_INFO_LEN+1];
    char        ol_delivery_d[OL_DELIVERY_D_LEN+1];
    RETCODE     rc;
    DBINT       rcint;

    // bind ORDER-LINE data
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

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

    rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
3);
    if (rc != 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,
SQLINT2, 6);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0, OL_DELIVERY_D_LEN,
NULL, 0, SQLCHARACTER, 7);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 8);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 9);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc3);

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

for (i = 0; i < orders_per_district; i++)
{
    ol_id = orders_buf[i].ol_id;
    ol_d_id = orders_buf[i].ol_d_id;
    ol_w_id = orders_buf[i].ol_w_id;

    for (j=0; j < orders_buf[i].ol_ol_cnt; j++)
    {
        ol = orders_buf[i].ol_ol[j].ol;
        ol_i_id = orders_buf[i].ol_ol[j].ol_i_id;
        ol_supply_w_id = orders_buf[i].ol_ol[j].ol_supply_w_id;
        ol_quantity = orders_buf[i].ol_ol[j].ol_quantity;
        ol_amount = orders_buf[i].ol_ol[j].ol_amount;

        strcpy(ol_delivery_d, orders_buf[i].ol_ol[j].ol_delivery_d);

        strcpy(ol_dist_info, orders_buf[i].ol_ol[j].ol_dist_info);

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

        order_line_rows_loaded++;
        CheckForCommit(o_hdbc3, o_hstmt3,
order_line_rows_loaded, "order_line", &order_line_time_start->time_start);
    }
}

// rcint = bcp_batch(o_hdbc3);
// if (rcint < 0)
//     HandleErrorDBC(o_hdbc3);

if ((ol_w_id == aptr->num_warehouses) && (ol_d_id == 10))
{
    rcint = bcp_done(o_hdbc3);
    if (rcint < 0)

```

```

        HandleErrorDBC(o_hdbc3);

        SQLFreeStmt(o_hstmt3, SQL_DROP);
        SQLDisconnect(o_hdbc3);
        SQLFreeConnect(o_hdbc3);

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

    }
}

//=====
//
// Function : GetPermutation
//
//=====
void GetPermutation(int perm[], int n)
{
    int i, r, t;

    for (i=1;i<=n;i++)
        perm[i] = i;

    for (i=1;i<=n;i++)
    {
        r = RandomNumber(i,n);
        t = perm[i];
        perm[i] = perm[r];
        perm[r] = t;
    }
}

//=====
//
// Function : CheckForCommit
//
//=====
void CheckForCommit(HDBC hdbc,
                    HSTMT hstmt,
                    int rows_loaded,
                    char *table_name,
                    long *time_start)
{
    long time_end, time_diff;
    // DBINT rcint;

    if ( !(rows_loaded % aptr->batch) )
    {
        // rcint = bcp_batch(hdbc);
        // if (rcint < 0)
        //     HandleErrorDBC(hdbc);

        time_end = (TimeNow() / MILLI);

```

```

        time_diff = time_end - *time_start;

        printf("-> Loaded %ld rows into %s in %ld sec - Total = %d (%.2f
rps)\n",
                aptr->batch,
                table_name,
                time_diff,
                rows_loaded,
                (float) aptr->batch / (time_diff ? time_diff
: 1L));

        *time_start = time_end;
    }
    return;
}

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

    char          szDriverString[300];
    char          szDriverStringOut[1024];
    SQLSMALLINT  cbDriverStringOut;

    SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv );

    SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0 );

    SQLAllocHandle(SQL_HANDLE_DBC, henv , &i_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv , &w_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc2);
    SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc2);
    SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc3);

    SQLSetConnectAttr(i_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(w_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(c_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(c_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );

    // Open connections to SQL Server

```

```

// Connection 1

    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

                aptr->server,
                aptr->user,
                aptr->password,
                aptr->database );

    rc = SQLSetConnectOption (i_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    rc = SQLDriverConnect ( i_hdbc1,

                            NULL,

                            (SQLCHAR*)&szDriverString[0] ,

                            SQL_NTS,

                            (SQLCHAR*)&szDriverStringOut[0],

                            sizeof(szDriverStringOut),

                            &cbDriverStringOut,
                            SQL_DRIVER_NOPROMPT );

    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

// Connection 2

    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

                aptr->server,
                aptr->user,
                aptr->password,
                aptr->database );

    rc = SQLSetConnectOption (w_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    rc = SQLDriverConnect ( w_hdbc1,

                            NULL,

                            (SQLCHAR*)&szDriverString[0] ,

                            SQL_NTS,

                            (SQLCHAR*)&szDriverStringOut[0],

                            sizeof(szDriverStringOut),

                            &cbDriverStringOut,

                            SQL_DRIVER_NOPROMPT );
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

// Connection 3

    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

                aptr->server,
                aptr->user,
                aptr->password,

```

```

aptr->database );

rc = SQLSetConnectOption (c_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = SQLDriverConnect ( c_hdbc1,
                        NULL,
                        (SQLCHAR*)&szDriverString[0] ,
                        SQL_NTS,
                        (SQLCHAR*)&szDriverStringOut[0],
                        sizeof(szDriverStringOut),
                        &cbDriverStringOut,
                        SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

// Connection 4
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );

rc = SQLSetConnectOption (c_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

rc = SQLDriverConnect ( c_hdbc2,
                        NULL,
                        (SQLCHAR*)&szDriverString[0] ,
                        SQL_NTS,
                        (SQLCHAR*)&szDriverStringOut[0],
                        sizeof(szDriverStringOut),
                        &cbDriverStringOut,
                        SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

// Connection 5
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );

rc = SQLSetConnectOption (o_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc1);

```

```

rc = SQLDriverConnect ( o_hdbc1,
                        NULL,
                        (SQLCHAR*)&szDriverString[0] ,
                        SQL_NTS,
                        (SQLCHAR*)&szDriverStringOut[0],
                        sizeof(szDriverStringOut),
                        &cbDriverStringOut,
                        SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc1);

// Connection 6
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );

rc = SQLSetConnectOption (o_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc2);

rc = SQLDriverConnect ( o_hdbc2,
                        NULL,
                        (SQLCHAR*)&szDriverString[0] ,
                        SQL_NTS,
                        (SQLCHAR*)&szDriverStringOut[0],
                        sizeof(szDriverStringOut),
                        &cbDriverStringOut,
                        SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc2);

// Connection 7
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
aptr->server,
aptr->user,
aptr->password,
aptr->database );

rc = SQLSetConnectOption (o_hdbc3, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc3);

rc = SQLDriverConnect ( o_hdbc3,
                        NULL,
                        (SQLCHAR*)&szDriverString[0] ,

```

```

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],
sizeof(szDriverStringOut),
&cbDriverStringOut,
SQL_DRIVER_NOPROMPT );
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc3);
}

//=====
//
// Function name: BuildIndex
//
//=====

void BuildIndex(char          *index_script)
{
    char    cmd[256];

    printf("Starting index creation:  %s\n",index_script);

    sprintf(cmd, "isql -S%s -U%s -P%s -e -i%s\\%s.sql > logs\\%s.log",
            aptr->server,
            aptr->user,
            aptr->password,
            aptr->index_script_path,
            index_script,
            index_script);

    system(cmd);

    printf("Finished index creation:  %s\n",index_script);
}

void HandleErrorDBC (SQLHDBC  hdbc1)
{
    SQLCHAR          SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
    SQLINTEGER  NativeError;
    SQLSMALLINT i, MsgLen;
    SQLRETURN  rc2;
    char        timebuf[128];
    char        datebuf[128];
    FILE        *fp1;

    i = 1;
    while (( rc2 = SQLGetDiagRec(SQL_HANDLE_DBC , hdbc1, i, SqlState ,
    &NativeError,
                                Msg, sizeof(Msg) , &MsgLen )) !=
SQL_NO_DATA )
    {

        sprintf( szLastError , "%s" ,  Msg );

        _strtime(timebuf);
        _strdate(datebuf);
    }
}

```

```

printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

fp1 = fopen("logs\\tpccldr.err","w");
if (fp1 == NULL)
    printf("ERROR:  Unable to open errorlog file.\n");
else
    {
        fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);
        fclose(fp1);
    }
    i++;
}

void HandleErrorSTMT (HSTMT  hstmt1)
{
    SQLCHAR          SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
    SQLINTEGER  NativeError;
    SQLSMALLINT i, MsgLen;
    SQLRETURN  rc2;
    char        timebuf[128];
    char        datebuf[128];
    FILE        *fp1;

    i = 1;
    while (( rc2 = SQLGetDiagRec(SQL_HANDLE_STMT , hstmt1, i, SqlState ,
    &NativeError,
                                Msg, sizeof(Msg) , &MsgLen )) !=
SQL_NO_DATA )
    {

        sprintf( szLastError , "%s" ,  Msg );

        _strtime(timebuf);
        _strdate(datebuf);

        printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

        fp1 = fopen("logs\\tpccldr.err","w");
        if (fp1 == NULL)
            printf("ERROR:  Unable to open errorlog file.\n");
        else
            {
                fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);
                fclose(fp1);
            }
            i++;
    }
}

void FormatDate ( char* szTimeCOutput )
{
    struct tm when;
    time_t now;
}

```

```

time( &now );
when = *localtime( &now );

mktime( &when );

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

return;
}

//=====
//
// Function   : CheckSQL
//
//=====

void CheckSQL()
{
    RETCODE          rc;

    char              szDriverString[300];
    char              szDriverStringOut[1024];
    int               SQLBuildFlag;
    char              resp;

    SQLSMALLINT       cbDriverStringOut;
    SQLCHAR           SQLVersion[19];
    SQLINTEGER        SQLVersionInd;

    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" ,
            aptr->server,
            aptr->user,
            aptr->password );

    if ( SQLSetConnectAttr( v_hdbc, SQL_ATTR_PACKET_SIZE, (SQLPOINTER)aptr->pack_size, SQL_IS_INTEGER ) != 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 ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
        HandleErrorDBC(v_hdbc);

    if ( SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt) != SQL_SUCCESS )
        HandleErrorSTMT(v_hstmt);

    rc = SQLBindCol(v_hstmt, 4, SQL_C_CHAR, &SQLVersion, sizeof(SQLVersion),
&SQLVersionInd);

    // issue SQL Server extended stored procedure (xp_msver) to determine
installed version
    rc = SQLExecDirect(v_hstmt, "EXECUTE xp_msver ProductVersion", SQL_NTS);

    if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
        HandleErrorSTMT(v_hstmt);

    rc = SQLFetch(v_hstmt);

    if (rc != SQL_SUCCESS)
        HandleErrorDBC(v_hdbc);

    // Check build number to ensure 8.00.194 or higher
    SQLBuildFlag = 1;

    // first check the Major version
    if ( SQLVersion[0] == '8' )
    {
        if (( SQLVersion[2] == '0') & ( SQLVersion[3] == '0' )
            {
                if ( SQLVersion[5] == '1' )
                {
                    if ( (SQLVersion[6] == '9') &
(SQLVersion[7] == '4') )
                    {
                        SQLBuildFlag = 0;
                        printf("You are using SQL Server
version = %9s\n\n", SQLVersion);
                    }
                    else
                    {
                        SQLBuildFlag = 1;
                    }
                }
                else
                {
                    if ( SQLVersion[5] == '3' )
                    {
                        if ( (SQLVersion[6] >= 53) &
(SQLVersion[7] >= 48) )
                        {
                            SQLBuildFlag = 0;
                            printf("You are using
SQL Server version = %9s\n\n", SQLVersion);
                        }
                    }
                    else
                    {

```



```

        }
    }
}
}
else
{
    SQLBuildFlag = 1;
}
if ( SQLBuildFlag == 1 )
{
    printf("NOTE:  The SQL Server version you are using is not
supported\n");
    printf("for TPC-C benchmarking.  You currently have SQL Server
version %9s\n",SQLVersion);
    printf("installed.  Please upgrade to Microsoft SQL Server 2000
(8.00.0194) or better.\n");
    printf("and re-run the SETUP program.\n\n");
    printf("Do you wish to continue with setup? (Y/N): ");
    resp = getchar();
    if ( ( resp == 'N' ) || (resp == 'n') )
    {
        printf("\nSetup Aborted!\n");
        exit(1);
    }
}
SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
SQLDisconnect(v_hdbc);
SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

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 LOGS\\ directory for database creation
errors.\n");

            // cleanup database connections and handles
            SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
            SQLDisconnect(v_hdbc);
            SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

            // since there is not a database, exit back to SETUP.CMD
            exit(1);
        }

        if ( SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc, &v_hstmt) != SQL_SUCCESS )
            HandleErrorDBC(v_hdbc);

        if ( SQLBindCol(v_hstmt, 1, SQL_C_ULONG, &TabCount, 0, &TabCountInd) !=
SQL_SUCCESS )
            HandleErrorSTMT(v_hstmt);

        // count the number of user tables from sysobjects
        rc = SQLExecDirect(v_hstmt, "select count(*) from sysobjects where xtype =
'\U'", SQL_NTS);
        if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
            HandleErrorSTMT(v_hstmt);

        if ( SQLFetch(v_hstmt) != SQL_SUCCESS )

```

```

        HandleErrorSTMT(v_hstmt);

// if the number of tables is less than 9, select all the user tables in
TPCC
if (TabCount != 9)
{
    SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);

    SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc, &v_hstmt);

    if ( SQLBindCol(v_hstmt, 1, SQL_C_CHAR, &TabName,
sizeof(TabName), &TabNameInd) != SQL_SUCCESS )
        HandleErrorSTMT(v_hstmt);

    // select the list of user tables into a result set
    rc = SQLExecDirect(v_hstmt, "select * from sysobjects where
xtype = 'U'", SQL_NTS);
    if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
        HandleErrorSTMT(v_hstmt);

    // go through the result set and set the bitmap for each found
table
    // set the bitmap to '1' if the table name is found
    while ((rc = SQLFetch(v_hstmt)) != SQL_NO_DATA)
    {
        switch( TabName[0] )
        {
            case 'w':
                TablesBitMap[0] = '1';
                break;
            case 'd':
                TablesBitMap[1] = '1';
                break;
            case 'c':
                TablesBitMap[2] = '1';
                break;
            case 'h':
                TablesBitMap[3] = '1';
                break;
            case 'n':
                TablesBitMap[4] = '1';
                break;
            case 'o':
                if (TabName[5] = 's')
                    TablesBitMap[5] = '1';
                if (TabName[5] = '_')
                    TablesBitMap[6] = '1';
                break;
            case 'i':
                TablesBitMap[7] = '1';
                break;
            case 's':
                TablesBitMap[8] = '1';
                break;
        }

        // a '0' ExitFlag means do NOT exit the loader early, a '1'
means exit the loader early
        ExitFlag = 0;
    }
}

```

```

// iterate through the bitmap to display which table(s) is
actually missing
for (i = 0; i <= 8; i++)
{
    switch(i)
    {
        case 0:
            if (TablesBitMap[i] == '0')
            {
                printf("The Warehouse table is
missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        case 1:
            if (TablesBitMap[i] == '0')
            {
                printf("The District table is
missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        case 2:
            if (TablesBitMap[i] == '0')
            {
                printf("The Customer table is
missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        case 3:
            if (TablesBitMap[i] == '0')
            {
                printf("The History table is
missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        case 4:
            if (TablesBitMap[i] == '0')
            {
                printf("The New_Order table is
missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        case 5:
            if (TablesBitMap[i] == '0')
            {
                printf("The Orders table is
missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        case 6:
            if (TablesBitMap[i] == '0')
            {
                printf("The Order_Line table is
missing or damaged.\n");
                ExitFlag = 1;
            }
            break;
        case 7:
    }
}

```

```

                                if (TablesBitMap[i] == '0')
                                {
                                    printf("The Item table is missing
or damaged.\n");
                                    ExitFlag = 1;
                                }
                                break;
                                case 8:
                                if (TablesBitMap[i] == '0')
                                {
                                    printf("The Stock table is missing
or damaged.\n");
                                    ExitFlag = 1;
                                }
                                break;
                                }
                                }
                                // if one or more tables are missing, display message and exit
the loader
                                if (ExitFlag = 1)
                                {
                                    printf("\nExiting TPC-C Loader!\n");
                                    printf("\nCheck LOGS\\ directory for database\n");
                                    printf("or table creation errors.\n");

                                    // cleanup database connections and handles
                                    SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
                                    SQLDisconnect(v_hdbc);
                                    SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

                                    exit(1);
                                }
                                }

                                // cleanup database connections and handles
                                SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
                                SQLDisconnect(v_hdbc);
                                SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

                                return;
}

```

version.sql

```

-- File:      VERSION.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.22
--           Copyright Microsoft, 2001
-- 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.
--
-- The value returned by this SP defines the "interface level",
-- which must match between the stored procs and the client code.
-- The interface level may be down rev from the current kit. This
-- indicates that the interface hasn't changed since that version.

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

```

Appendix C: Tunable Parameters

Microsoft SQL Server 2000 Startup Parameters

```
C:\Program Files\Microsoft SQL
Server\MSSQL\BINN\sqlservr.exe
-eC:\Program Files\Microsoft SQL
Server\MSSQL\LOG\ERRORLOG -x -c -t3502
-g100
```

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
- g100 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.

Boot.ini Parameters

```
[boot loader]
timeout=30
```

```
default=multi(0)disk(0)rdisk(0)partition(2)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(2)\WINNT="Microsoft
Windows 2000 Server" /pae /fastdetect
```

Microsoft SQL Server 2000 Configuration Parameters

```
1> 2> 3> DBCC execution completed. If DBCC printed
error messages, contact your system administrator.
Configuration option 'show advanced options' changed
from 1 to 1. Run the RECONFIGURE statement to
install.
```

```
sp_configure "show advanced",1
1> 2> reconfigure with override
1> 2> sp_configure
name
minimum maximum config_value run_value
-----
-----
affinity mask
-2147483648 2147483647 15 15
allow updates
0 1 0 0
awe enabled
0 1 1 1
c2 audit mode
0 1 0 0
cost threshold for parallelism
0 32767 5 5
Cross DB Ownership Chaining
0 1 0 0
cursor threshold
-1 2147483647 -1 -1
default full-text language
0 2147483647 1033 1033
default language
0 9999 0 0
fill factor (%)
0 100 0 0
index create memory (KB)
704 2147483647 704 704
lightweight pooling
0 1 1 1
locks
5000 2147483647 0 0
max degree of parallelism
0 32 1 1
```

```
max server memory (MB)
4 2147483647 2147483647 2147483647
max text repl size (B)
0 2147483647 65536 65536
max worker threads
32 32767 500 500
media retention
0 365 0 0
min memory per query (KB)
512 2147483647 512 512
min server memory (MB)
0 2147483647 0 0
nested triggers
0 1 1 1
network packet size (B)
512 65536 2048 2048
open objects
0 2147483647 0 0
priority boost
0 1 1 1
query governor cost limit
0 2147483647 0 0
query wait (s)
-1 2147483647 -1 -1
recovery interval (min)
0 32767 116 116
remote access
0 1 1 1
remote login timeout (s)
0 2147483647 20 20
remote proc trans
0 1 0 0
remote query timeout (s)
0 2147483647 600 600
scan for startup procs
0 1 0 0
set working set size
0 1 0 0
show advanced options
0 1 1 1
two digit year cutoff
1753 9999 2049 2049
user connections
0 32767 0 0
user options
0 32767 0 0

1>
```

Benchcraft Profile

```
Profile: vy_4800wh-2clients
File Path: C:\Benchcraft\vy_4800wh-
2clients.pro
Version: 3
```

Number of Engines: 4

Name: c18a

Description:
 Directory: c:\blog\cl8a.log
 Machine: N8
 Parameter Set: 2.2
 Index: 200000000
 Seed: 18546
 Configured Users: 12000
 Pipe Name: DRIVER3-2120427953
 Connect Rate: 11
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 5
 CPU: 1

Name: cl8b
 Description:
 Directory: c:\blog\cl8b.log
 Machine: N8
 Parameter Set: 2.2
 Index: 300000000
 Seed: 18546
 Configured Users: 12000
 Pipe Name: DRIVER4-2120385859
 Connect Rate: 11
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 5
 CPU: 0

Name: cl1a
 Description:
 Directory: c:\blog\cl21a.log
 Machine: n7
 Parameter Set: 2.2
 Index: 400000000
 Seed: 18546
 Configured Users: 12000
 Pipe Name: DRIVER5-1160480687
 Connect Rate: 11
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 5
 CPU: 0

Name: cl1b
 Description:
 Directory: c:\blog\cl21b.log
 Machine: n7
 Parameter Set: 2.2
 Index: 500000000
 Seed: 18546
 Configured Users: 12000
 Pipe Name: DRIVER6-1160429187
 Connect Rate: 11
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 5
 CPU: 1

Number of User groups: 4

Driver Engine: cl8a
 IIS Server: cr8
 SQL Server: vy
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 1201 - 2400
 w_id Min Warehouse: 1
 w_id Max Warehouse: 4800
 Scale: Normal
 User Count: 12000
 District id: 1
 Scale Down: No

Driver Engine: cl8b
 IIS Server: cr8
 SQL Server: vy
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 1 - 1200
 w_id Min Warehouse: 1
 w_id Max Warehouse: 4800
 Scale: Normal
 User Count: 12000
 District id: 1
 Scale Down: No

Driver Engine: cl1a
 IIS Server: cr1
 SQL Server: vy
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 2401 - 3600
 w_id Min Warehouse: 1
 w_id Max Warehouse: 4800
 Scale: Normal
 User Count: 12000
 District id: 1
 Scale Down: No

Driver Engine: cl1b
 IIS Server: cr1
 SQL Server: vy
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 3601 - 4800
 w_id Min Warehouse: 1
 w_id Max Warehouse: 4800
 Scale: Normal
 User Count: 12000
 District id: 1
 Scale Down: No

Number of Parameter Sets: 63

~Default
 Default Parameter Set

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.05		18.01	New Order	5.00	10.00
			0.10	10.00	0.10
12.05		3.01	Payment	5.00	10.00
			0.10	1.00	0.10
5.05		2.01	Delivery	5.00	1.00
			0.10	20.00	0.10
5.05		2.01	Stock Level	1.00	1.00
			0.10	20.00	0.10
10.05		2.01	Order Status	5.00	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	5.00	44.75
			0.10	5.00	0.10
12.05		3.01	Payment	5.00	43.10
			0.10	5.00	0.10
5.05		2.01	Delivery	5.00	4.05
			0.10	5.00	0.10
5.05		2.01	Stock Level	20.00	4.05
			0.10	20.00	0.10
10.05		2.01	Order Status	5.00	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	5.00	10.00
			0.00	5.00	0.00
0.00		0.00	Payment	5.00	10.00
			0.00	5.00	0.00
0.00		0.00	Delivery	5.00	1.00
			0.00	5.00	0.00
0.00		0.00	Stock Level	20.00	1.00
			0.00	20.00	0.00
0.00		0.00	Order Status	5.00	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	5.00	44.75
			0.10	5.00	0.10
13.00		3.01	Payment	5.00	43.10
			0.10	5.00	0.10
6.00		2.01	Delivery	5.00	4.05
			0.10	5.00	0.10
6.00		2.01	Stock Level	20.00	4.05
			0.10	20.00	0.10

Key	RT	RT	Menu	Txn	Think
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	44.83	
			0.10	5.00	0.10
16.00	3.01		Payment	43.05	
			0.10	5.00	0.10
9.00	2.01		Delivery	4.04	
			0.10	5.00	0.10
9.00	2.01		Stock Level	4.04	
			0.10	20.00	0.10
14.00	2.01		Order Status	4.04	
			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

Key	RT	RT	Menu	Txn	Think
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
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	0.10	5.00	0.10	4.05		
		2.0					
		2.0 tt					
Key	RT	RT	Menu	Txn	Think		
				Weight	Time		
Time	Delay	Fence	Delay				
24.10	18.01	New Order	0.10	5.00	44.75	0.10	
24.10	3.01	Payment	0.10	5.00	43.10	0.10	
10.10	2.01	Delivery	0.10	5.00	4.05	0.10	
10.10	2.01	Stock Level	0.10	20.00	4.05	0.10	
20.10	2.01	Order Status	0.10	5.00	4.05	0.10	
		5.0					
		5.0 tt					
Key	RT	RT	Menu	Txn	Think		
				Weight	Time		
Time	Delay	Fence	Delay				
60.25	18.01	New Order	0.10	5.00	44.75	0.10	
60.25	3.01	Payment	0.10	5.00	43.10	0.10	
25.25	2.01	Delivery	0.10	5.00	4.05	0.10	
25.25	2.01	Stock Level	0.10	20.00	4.05	0.10	
50.25	2.01	Order Status	0.10	5.00	4.05	0.10	
		4.5					
		4.5 tt					
Key	RT	RT	Menu	Txn	Think		
				Weight	Time		
Time	Delay	Fence	Delay				
54.20	18.01	New Order	0.10	5.00	44.75	0.10	
54.20	3.01	Payment	0.10	5.00	43.10	0.10	
22.70	2.01	Delivery	0.10	5.00	4.05	0.10	
22.70	2.01	Stock Level	0.10	20.00	4.05	0.10	
45.20	2.01	Order Status	0.10	5.00	4.05	0.10	
		3.5					
		3.5 tt					
Key	RT	RT	Menu	Txn	Think		
				Weight	Time		
Time	Delay	Fence	Delay				
42.10	18.01	New Order	0.10	5.00	44.75	0.10	

42.10	3.01	Payment	0.10	5.00	43.10	0.10	
17.60	2.01	Delivery	0.10	5.00	4.05	0.10	
17.60	2.01	Stock Level	0.10	20.00	4.05	0.10	
35.10	2.01	Order Status	0.10	5.00	4.05	0.10	
		1.8					
		1.8 tt					
Key	RT	RT	Menu	Txn	Think		
				Weight	Time		
Time	Delay	Fence	Delay				
21.60	18.01	New Order	0.10	5.00	44.75	0.10	
21.60	3.01	Payment	0.10	5.00	43.10	0.10	
9.09	2.01	Delivery	0.10	5.00	4.05	0.10	
9.09	2.01	Stock Level	0.10	20.00	4.05	0.10	
18.09	2.01	Order Status	0.10	5.00	4.05	0.10	
		4.2					
		4.2 tt					
Key	RT	RT	Menu	Txn	Think		
				Weight	Time		
Time	Delay	Fence	Delay				
54.20	18.01	New Order	0.10	5.00	44.75	0.10	
54.20	3.01	Payment	0.10	5.00	43.10	0.10	
22.70	2.01	Delivery	0.10	5.00	4.05	0.10	
22.70	2.01	Stock Level	0.10	20.00	4.05	0.10	
45.20	2.01	Order Status	0.10	5.00	4.05	0.10	
		1.6					
		1.6 tt					
Key	RT	RT	Menu	Txn	Think		
				Weight	Time		
Time	Delay	Fence	Delay				
19.20	18.01	New Order	0.10	5.00	44.75	0.10	
19.20	3.01	Payment	0.10	5.00	43.10	0.10	
8.08	2.01	Delivery	0.10	5.00	4.05	0.10	
8.08	2.01	Stock Level	0.10	20.00	4.05	0.10	
16.08	2.01	Order Status	0.10	5.00	4.05	0.10	
		1.4					
		1.4 tt					

Key	RT	RT	Menu	Txn	Think		
				Weight	Time		
Time	Delay	Fence	Delay				
16.87	18.01	New Order	0.10	5.00	44.75	0.10	
16.87	3.01	Payment	0.10	5.00	43.10	0.10	
7.07	2.01	Delivery	0.10	5.00	4.05	0.10	
7.07	2.01	Stock Level	0.10	20.00	4.05	0.10	
14.07	2.01	Order Status	0.10	5.00	4.05	0.10	
		1.2					
		1.2 tt					
Key	RT	RT	Menu	Txn	Think		
				Weight	Time		
Time	Delay	Fence	Delay				
14.46	18.01	New Order	0.10	5.00	44.83	0.10	
14.46	3.01	Payment	0.10	5.00	43.05	0.10	
6.06	2.01	Delivery	0.10	5.00	4.04	0.10	
6.06	2.01	Stock Level	0.10	20.00	4.04	0.10	
12.06	2.01	Order Status	0.10	5.00	4.04	0.10	
		3.5					
		3.5 tt					
Key	RT	RT	Menu	Txn	Think		
				Weight	Time		
Time	Delay	Fence	Delay				
42.10	18.01	New Order	0.10	5.00	44.75	0.10	
42.10	3.01	Payment	0.10	5.00	43.10	0.10	
17.60	2.01	Delivery	0.10	5.00	4.05	0.10	
17.60	2.01	Stock Level	0.10	20.00	4.05	0.10	
35.10	2.01	Order Status	0.10	5.00	4.05	0.10	
		1.9					
		1.9 tt					
Key	RT	RT	Menu	Txn	Think		
				Weight	Time		
Time	Delay	Fence	Delay				
22.89	18.01	New Order	0.10	5.00	44.75	0.10	
22.89	3.01	Payment	0.10	5.00	43.10	0.10	
9.59	2.01	Delivery	0.10	5.00	4.05	0.10	
9.59	2.01	Stock Level	0.10	20.00	4.05	0.10	

Key	RT	RT	Menu	Txn	Think
19.09	2.01		Order Status	4.05	
			0.10	5.00	0.10
			1.1 better		
			1.1 tt better		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
13.25	18.01		New Order	44.92	
			0.10	5.00	0.10
13.25	3.01		Payment	43.02	
			0.10	5.00	0.10
5.55	2.01		Delivery	4.02	
			0.10	5.00	0.10
5.55	2.01		Stock Level	4.02	
			0.10	20.00	0.10
11.05	2.01		Order Status	4.02	
			0.10	5.00	0.10
			1.05 better		
			1.05 tt better		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.65	18.01		New Order	44.92	
			0.10	5.00	0.10
12.65	3.01		Payment	43.02	
			0.10	5.00	0.10
5.30	2.01		Delivery	4.02	
			0.10	5.00	0.10
5.30	2.01		Stock Level	4.02	
			0.10	20.00	0.10
10.55	2.01		Order Status	4.02	
			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	44.83	
			0.10	5.00	0.10
13.13	3.01		Payment	43.05	
			0.10	5.00	0.10
5.50	2.01		Delivery	4.04	
			0.10	5.00	0.10
5.50	2.01		Stock Level	4.04	
			0.10	20.00	0.10
10.95	2.01		Order Status	4.04	
			0.10	5.00	0.10
			1.08		
			1.08 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
13.01	18.01		New Order	44.83	
			0.10	5.00	0.10

Key	RT	RT	Menu	Txn	Think
13.01	3.01		Payment	43.05	
			0.10	5.00	0.10
5.45	2.01		Delivery	4.04	
			0.10	5.00	0.10
5.45	2.01		Stock Level	4.04	
			0.10	20.00	0.10
10.85	2.01		Order Status	4.04	
			0.10	5.00	0.10
			1.07		
			1.07 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.89	18.01		New Order	44.83	
			0.10	5.00	0.10
12.89	3.01		Payment	43.05	
			0.10	5.00	0.10
5.40	2.01		Delivery	4.04	
			0.10	5.00	0.10
5.40	2.01		Stock Level	4.04	
			0.10	20.00	0.10
10.75	2.01		Order Status	4.04	
			0.10	5.00	0.10
			1.06		
			1.06 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.77	18.01		New Order	44.83	
			0.10	5.00	0.10
12.77	3.01		Payment	43.05	
			0.10	5.00	0.10
5.35	2.01		Delivery	4.04	
			0.10	5.00	0.10
5.35	2.01		Stock Level	4.04	
			0.10	20.00	0.10
10.65	2.01		Order Status	4.04	
			0.10	5.00	0.10
			1.15		
			1.15 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
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
15.06	18.01		New Order	44.83	
			0.10	5.00	0.10
15.06	3.01		Payment	43.05	
			0.10	5.00	0.10
6.31	2.01		Delivery	4.04	
			0.10	5.00	0.10
6.31	2.01		Stock Level	4.04	
			0.10	20.00	0.10
12.56	2.01		Order Status	4.04	
			0.10	5.00	0.10
			1.3		
			1.3 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
15.66	18.01		New Order	44.83	
			0.10	5.00	0.10
15.66	3.01		Payment	43.05	
			0.10	5.00	0.10
6.56	2.01		Delivery	4.04	
			0.10	5.00	0.10
6.56	2.01		Stock Level	4.04	
			0.10	20.00	0.10
13.06	2.01		Order Status	4.04	
			0.10	5.00	0.10
			1.12		
			1.12 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
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
Time	Delay	Fence	Delay	Weight	Time
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

Key	RT	RT	Menu	Txn	Think
11.85	2.01	0.10	5.00	4.05	0.10
		1.22			
		1.22 tt			
Time	Delay	Fence	Delay	Weight	Time
14.70	18.01	0.10	5.00	44.75	0.10
14.70	3.01	0.10	5.00	43.10	0.10
6.16	2.01	0.10	5.00	4.05	0.10
6.16	2.01	0.10	20.00	4.05	0.10
12.26	2.01	0.10	5.00	4.05	0.10
		1.28			
		1.28 tt			
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
15.42	18.01	0.10	5.00	44.75	0.10
15.42	3.01	0.10	5.00	43.10	0.10
6.46	2.01	0.10	5.00	4.05	0.10
6.46	2.01	0.10	20.00	4.05	0.10
12.86	2.01	0.10	5.00	4.05	0.10
		1.04 better			
		1.04 tt better			
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.53	18.01	0.10	5.00	44.92	0.10
12.53	3.01	0.10	5.00	43.01	0.10
5.25	2.01	0.10	5.00	4.02	0.10
5.25	2.01	0.10	20.00	4.03	0.10
10.45	2.01	0.10	5.00	4.02	0.10
		1.03			
		1.03 tt			
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.41	18.01	0.10	5.00	44.83	0.10

Key	RT	RT	Menu	Txn	Think
12.41	3.01	0.10	5.00	43.05	0.10
5.20	2.01	0.10	5.00	4.04	0.10
5.20	2.01	0.10	20.00	4.04	0.10
10.35	2.01	0.10	5.00	4.04	0.10
		1.02			
		1.02 tt			
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.29	18.01	0.10	5.00	44.83	0.10
12.29	3.01	0.10	5.00	43.05	0.10
5.15	2.01	0.10	5.00	4.04	0.10
5.15	2.01	0.10	20.00	4.04	0.10
10.25	2.01	0.10	5.00	4.04	0.10
		1.01			
		1.01 tt			
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.17	18.01	0.10	5.00	44.83	0.10
12.17	3.01	0.10	5.00	43.05	0.10
5.10	2.01	0.10	5.00	4.04	0.10
5.10	2.01	0.10	20.00	4.04	0.10
10.15	2.01	0.10	5.00	4.04	0.10
		1.005_best			
		1.005 tt best			
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.11	18.01	0.10	5.00	44.96	0.10
12.11	3.01	0.10	5.00	43.00	0.10
5.07	2.01	0.10	5.00	4.00	0.10
5.07	2.01	0.10	20.00	4.03	0.10
10.10	2.01	0.10	5.00	4.01	0.10
		1.001_best			
		1.001 tt best			

Key	RT	RT	Menu	Txn	Think
12.06	18.01	0.10	5.00	44.96	0.10
12.06	3.01	0.10	5.00	43.00	0.10
5.06	2.01	0.10	5.00	4.00	0.10
5.06	2.01	0.10	20.00	4.03	0.10
10.06	2.01	0.10	5.00	4.01	0.10
		1.03 better			
		1.03 tt more aggressive			
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.41	18.01	0.10	5.00	44.92	0.10
12.41	3.01	0.10	5.00	43.01	0.10
5.20	2.01	0.10	5.00	4.02	0.10
5.20	2.01	0.10	20.00	4.03	0.10
10.35	2.01	0.10	5.00	4.02	0.10
		1.005 better			
		1.005 tt more aggressive			
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.11	18.01	0.10	5.00	44.92	0.10
12.11	3.01	0.10	5.00	43.01	0.10
5.07	2.01	0.10	5.00	4.02	0.10
5.07	2.01	0.10	20.00	4.03	0.10
10.10	2.01	0.10	5.00	4.02	0.10
		1.02 better			
		1.02 tt more aggressive			
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.29	18.01	0.10	5.00	44.92	0.10
12.29	3.01	0.10	5.00	43.02	0.10
5.15	2.01	0.10	5.00	4.02	0.10
5.15	2.01	0.10	20.00	4.02	0.10

Key	RT	RT	Menu	Txn	Think
10.25	2.01		Order Status	4.02	
			0.10	5.00	0.10
			1.01 better		
			1.01 tt better		
Time	Delay	Fence	Delay	Weight	Time
12.17	18.01		New Order	44.92	
			0.10	5.00	0.10
12.17	3.01		Payment	43.02	
			0.10	5.00	0.10
5.10	2.01		Delivery	4.02	
			0.10	5.00	0.10
5.10	2.01		Stock Level	4.02	
			0.10	20.00	0.10
10.15	2.01		Order Status	4.02	
			0.10	5.00	0.10
			1.02 best		
			1.02 tt best		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.29	18.01		New Order	44.96	
			0.10	5.00	0.10
12.29	3.01		Payment	43.00	
			0.10	5.00	0.10
5.15	2.01		Delivery	4.00	
			0.10	5.00	0.10
5.15	2.01		Stock Level	4.03	
			0.10	20.00	0.10
10.25	2.01		Order Status	4.01	
			0.10	5.00	0.10
			1.03 best		
			1.03 tt best		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.41	18.01		New Order	44.96	
			0.10	5.00	0.10
12.41	3.01		Payment	43.01	
			0.10	5.00	0.10
5.20	2.01		Delivery	4.01	
			0.10	5.00	0.10
5.20	2.01		Stock Level	4.01	
			0.10	20.00	0.10
10.35	2.01		Order Status	4.01	
			0.10	5.00	0.10
			5.5		
			5.5 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
66.28	18.01		New Order	44.83	
			0.10	5.00	0.10

Key	RT	RT	Menu	Txn	Think
66.28	3.01		Payment	43.05	
			0.10	5.00	0.10
27.77	2.01		Delivery	4.04	
			0.10	5.00	0.10
27.77	2.01		Stock Level	4.04	
			0.10	20.00	0.10
55.27	2.01		Order Status	4.04	
			0.10	5.00	0.10
			6.0		
			6.0 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
72.30	18.01		New Order	44.83	
			0.10	5.00	0.10
72.30	3.01		Payment	43.05	
			0.10	5.00	0.10
30.30	2.01		Delivery	4.04	
			0.10	5.00	0.10
30.30	2.01		Stock Level	4.04	
			0.10	20.00	0.10
60.30	2.01		Order Status	4.04	
			0.10	5.00	0.10
			6.5		
			6.5 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
79.53	18.01		New Order	44.83	
			0.10	5.00	0.10
79.53	3.01		Payment	43.05	
			0.10	5.00	0.10
33.33	2.01		Delivery	4.04	
			0.10	5.00	0.10
33.33	2.01		Stock Level	4.04	
			0.10	20.00	0.10
66.33	2.01		Order Status	4.04	
			0.10	5.00	0.10
			7.0		
			7.0 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
84.35	18.01		New Order	44.83	
			0.10	5.00	0.10
84.35	3.01		Payment	43.05	
			0.10	5.00	0.10
35.35	2.01		Delivery	4.04	
			0.10	5.00	0.10
35.35	2.01		Stock Level	4.04	
			0.10	20.00	0.10
70.35	2.01		Order Status	4.04	
			0.10	5.00	0.10
			7.5		
			7.5 tt		

Key	RT	RT	Menu	Txn	Think
90.38	18.01		New Order	44.83	
			0.10	5.00	0.10
90.38	3.01		Payment	43.05	
			0.10	5.00	0.10
37.88	2.01		Delivery	4.04	
			0.10	5.00	0.10
37.88	2.01		Stock Level	4.04	
			0.10	20.00	0.10
75.38	2.01		Order Status	4.04	
			0.10	5.00	0.10
			8.0		
			8.0 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
96.40	18.01		New Order	44.83	
			0.10	5.00	0.10
96.40	3.01		Payment	43.05	
			0.10	5.00	0.10
40.40	2.01		Delivery	4.04	
			0.10	5.00	0.10
40.40	2.01		Stock Level	4.04	
			0.10	20.00	0.10
80.40	2.01		Order Status	4.04	
			0.10	5.00	0.10
			8.5		
			8.5 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
102.43	18.01		New Order	44.83	
			0.10	5.00	0.10
192.43	3.01		Payment	43.05	
			0.10	5.00	0.10
42.92	2.01		Delivery	4.04	
			0.10	5.00	0.10
42.92	2.01		Stock Level	4.04	
			0.10	20.00	0.10
85.42	2.01		Order Status	4.04	
			0.10	5.00	0.10
			9.0		
			9.0 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
108.45	18.01		New Order	44.83	
			0.10	5.00	0.10
108.45	3.01		Payment	43.05	
			0.10	5.00	0.10
45.45	2.01		Delivery	4.04	
			0.10	5.00	0.10
45.45	2.01		Stock Level	4.04	
			0.10	20.00	0.10

```

Order Status      4.04
90.45      2.01      0.10      5.00      0.10
          9.5
          9.5 tt
Key      RT      RT      Menu      Txn      Think
Time      Delay      Fence      Delay      Weight      Time
114.47      18.01      New Order      0.10      5.00      44.83      0.10
114.47      3.01      Payment      0.10      5.00      43.05      0.10
47.98      2.01      Delivery      0.10      5.00      4.04      0.10
47.98      2.01      Stock Level      0.10      20.00      4.04      0.10
95.47      2.01      Order Status      0.10      5.00      4.04      0.10
          10
          10 tt
Key      RT      RT      Menu      Txn      Think
Time      Delay      Fence      Delay      Weight      Time
120.50      18.01      New Order      0.10      5.00      44.83      0.10
120.50      3.01      Payment      0.10      5.00      43.05      0.10
50.50      2.01      Delivery      0.10      5.00      4.04      0.10
50.50      2.01      Stock Level      0.10      20.00      4.04      0.10
100.50      2.01      Order Status      0.10      5.00      4.04      0.10
          1.001 better
          1.001 more aggressive
Key      RT      RT      Menu      Txn      Think
Time      Delay      Fence      Delay      Weight      Time
12.06      18.01      New Order      0.10      5.00      44.92      0.10
12.06      3.01      Payment      0.10      5.00      43.02      0.10
5.06      2.01      Delivery      0.10      5.00      4.02      0.10
5.06      2.01      Stock Level      0.10      20.00      4.02      0.10
10.06      2.01      Order Status      0.10      5.00      4.02      0.10
          FullSpeed
          1.000 tt
Key      RT      RT      Menu      Txn      Think
Time      Delay      Fence      Delay      Weight      Time
12.05      18.01      New Order      0.10      5.00      44.92      0.10

```

```

Payment      43.02
12.05      3.01      0.10      5.00      0.10
Delivery      4.02
5.05      2.01      0.10      5.00      0.10
Stock Level      4.02
5.05      2.01      0.10      20.00      0.10
Order Status      4.02
10.05      2.01      0.10      5.00      0.10

```

Internet Server Registry Parameters

REGEDIT4

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
InetInfo]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,44,41,50,53,56,43,00,00
"PoolThreadLimit"=dword:000003fe
"ThreadTimeout"=dword:00015180

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
InetInfo\Performance]
"Library"="infectrs.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:30,bb,ee,43,77,5b,c2,01,10,25,00,00,00,00,0
0,00
"WbemAdapFileTime"=hex:00,73,79,5b,bc,d4,c0,01
"WbemAdapFileSize"=dword:00002510
"WbemAdapStatus"=dword:00000000

```

World Wide Web Service Registry Parameters

REGEDIT4

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,7
3,74,65,6d,33,32,5c,69,6e,\

```

```

65,74,73,72,76,5c,69,6e,65,74,69,6e,66,6f,2e,65,78,65
,00
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):49,49,53,41,44,4d,49,4e,00,0
0
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Provides Web connectivity and
administration through the Internet Information
Services snap-in."
"FailureActions"=hex:ff,ff,ff,ff,80,3a,0e,00,90,3a,0e
,00,03,00,00,00,98,3a,0e,\

```

```

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00

```

```

[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\\iisrmap
.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,1"
"/IISHelp"="c:\\winnt\\help\\iishelp,1"
"/IISAdmin"="C:\\WINNT\\System32\\inetsrv\\iisadmin,
1"

```

```
"/IISSamples"="c:\\inetpub\\iissamples,,1"
"/MSADC"="c:\\program files\\common
files\\system\\msadc,,1"
"/_vti_bin"="C:\\Program Files\\Common
Files\\Microsoft Shared\\Web Server
Extensions\\40\\isapi,,1"
"/Printers"="C:\\WINNT\\web\\printers,,201"
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services
\\W3SVC\\Performance]
"Library"="w3ctrs.dll"
"Open"="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:de,61,7e,46,77,5b,c2,01,10,3d,00,00,00,00,0,0,0,0
"WbemAdapFileTime"=hex:00,73,79,5b,bc,d4,c0,01
"WbemAdapFileSize"=dword:00001d10
"WbemAdapStatus"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services
\\W3SVC\\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\\
```

```
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00,00,\\
```

```
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,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\\
```

```
00,05,20,00,00,00,23,02,00,00,72,00,73,00,01,01,00,00,00,00,05,12,00,00,\\
00,01,01,00,00,00,00,05,12,00,00,00
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services
\\W3SVC\\Enum]
"0"="Root\\LEGACY_W3SVC\\00000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

TPCC Application

Registry Parameters

REGEDIT4

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\TPCC]
"Path"="C:\\inetpub\\wwwroot\\"
"NumberOfDeliveryThreads"=dword:0000002f
"MaxConnections"=dword:00006cfc
"MaxPendingDeliveries"=dword:000003e8
"DB_Protocol"="ODBC"
"TxnMonitor"="COM"
"DbServer"="vy"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"=""
"COM_SinglePool"="YES"
```

Server Bus Performance Driver Registry Parameters

```
Key Name:
HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\
hpqcissb
Class Name: <NO CLASS>
Last Write Time: 2/26/2004 - 12:03 PM
Value 0
Name: Type
Type: REG_DWORD
Data: 0x1
```

```
Value 1
Name: Start
Type: REG_DWORD
Data: 0
```

```
Value 2
Name: ErrorControl
Type: REG_DWORD
Data: 0x1
```

```
Value 3
Name: Tag
Type: REG_DWORD
Data: 0x102
```

```
Value 4
Name: ImagePath
Type: REG_EXPAND_SZ
Data: system32\\DRIVERS\\hpqcissb.sys
```

```
Value 5
Name: DisplayName
Type: REG_SZ
Data: Smart Array Controllers Non-
Miniport Bus Driver
```

```
Value 6
Name: Group
Type: REG_SZ
Data: port
```

```
Key Name:
HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\
hpqcissb\\Parameters
Class Name: <NO CLASS>
Last Write Time: 2/26/2004 - 1:26 AM
Value 0
Name: CompletionMode
Type: REG_DWORD
Data: 0x2
```

```
Key Name:
HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\
hpqcissb\\Parameters\\Controller0
Class Name: <NO CLASS>
Last Write Time: 11/4/2002 - 11:59 AM
Value 0
Name: CompletionMode
Type: REG_DWORD
Data: 0x1
```

```
Key Name:
HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\
hpqcissb\\Security
Class Name: <NO CLASS>
Last Write Time: 2/26/2004 - 11:50 AM
Value 0
Name: Security
Type: REG_BINARY
Data:
00000000 01 00 14 80 90 00 00 00 - 9c 00 00 00 14
00 00 00 .....
00000010 30 00 00 00 02 00 1c 00 - 01 00 00 00 02
80 14 00 0.....
00000020 ff 01 0f 00 01 01 00 00 - 00 00 00 01 00
00 00 00 Ÿ.....
00000030 02 00 60 00 04 00 00 00 - 00 00 14 00 fd
01 02 00 .`.....Ÿ...
00000040 01 01 00 00 00 00 05 - 12 00 00 00 00
00 18 00 .....
00000050 ff 01 0f 00 01 02 00 00 - 00 00 00 05 20
00 00 00 Ÿ.....
00000060 20 02 00 00 00 14 00 - 8d 01 02 00 01
01 00 00 .....
00000070 00 00 00 05 0b 00 00 00 - 00 00 18 00 fd
01 02 00 .....Ÿ...
00000080 01 02 00 00 00 00 05 - 20 00 00 00 23
02 00 00 .....#...
00000090 01 01 00 00 00 00 05 - 12 00 00 00 01
01 00 00 .....
```

00 00 00 05 12 00 00 00 -
.....

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
hpqcissb\Enum
Class Name: <NO CLASS>
Last Write Time: 2/26/2004 - 12:03 PM
Value 0
Name: 0
Type: REG_SZ
Data:
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\3&107002
0&0&08
Value 1
Name: Count
Type: REG_DWORD
Data: 0x3
Value 2
Name: NextInstance
Type: REG_DWORD
Data: 0x3
Value 3
Name: 1
Type: REG_SZ
Data:
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\3&29e819
82&0&10
Value 4
Name: 2
Type: REG_SZ
Data:
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\3&172e68
dd&0&10

System Summary

System Information report written at: 02/11/04
15:24:45
System Name: VY
[System Summary]

Item	Value
OS Name	Microsoft(R) Windows(R) Server 2003, Enterprise Edition
Version	5.2.3790 Build 3790
OS Manufacturer	Microsoft Corporation
System Name	VY
System Manufacturer	Compaq
System Model	ProLiant ML370 G3
System Type	X86-based PC

Processor x86 Family 15 Model 2 Stepping 5
GenuineIntel ~3189 Mhz
Processor x86 Family 15 Model 2 Stepping 5
GenuineIntel ~3189 Mhz
Processor x86 Family 15 Model 2 Stepping 5
GenuineIntel ~3189 Mhz
Processor x86 Family 15 Model 2 Stepping 5
GenuineIntel ~3189 Mhz
BIOS Version/Date Compaq P28, 12/31/2003
SMBIOS Version 2.3
Windows Directory C:\WINDOWS.0
System Directory C:\WINDOWS.0\system32
Boot Device \Device\HarddiskVolume8
Locale United States
Hardware Abstraction Layer Version = "5.2.3790.0
(srv03_rtm.030324-2048)"
User Name VY\Administrator
Time Zone Central Standard Time
Total Physical Memory 12,288.00 MB
Available Physical Memory 11.40 GB
Total Virtual Memory 25.28 GB
Available Virtual Memory 24.80 GB
Page File Space 13.53 GB
Page File C:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

Resource	Device	Status
Memory Address 0xF7800000-0xF7BFFFFFF	PCI bus	
Memory Address 0xF7800000-0xF7BFFFFFF	Smart Array	
5300 Controller (Non-Miniport)		
I/O Port 0x00000000-0x00000CFF	PCI bus	
I/O Port 0x00000000-0x00000CFF	PCI bus	
I/O Port 0x00000000-0x00000CFF	Direct memory	
access controller		
Memory Address 0xF7C00000-0xF7FFFFFF	PCI bus	
Memory Address 0xF7C00000-0xF7FFFFFF	Smart Array	
5300 Controller (Non-Miniport)		
I/O Port 0x000003C0-0x000003DF	PCI bus	
I/O Port 0x000003C0-0x000003DF	Standard VGA	
Graphics Adapter		
I/O Port 0x00006000-0x000064FF	PCI bus	
I/O Port 0x00006000-0x000064FF	Smart Array	
5300 Controller (Non-Miniport)		
I/O Port 0x00003000-0x000034FF	PCI bus	
I/O Port 0x00003000-0x000034FF	Compaq 64-	
bit/66MHz Dual Channel Wide Ultra3 SCSI Adapter		
Memory Address 0xF7400000-0xF77FFFFF	PCI bus	
Memory Address 0xF7400000-0xF77FFFFF	Smart Array	
5300 Controller (Non-Miniport)		
I/O Port 0x00005000-0x000054FF	PCI bus	

I/O Port 0x00005000-0x000054FF	Smart Array
5300 Controller (Non-Miniport)	
Memory Address 0xA0000-0xBFFFF	PCI bus
Memory Address 0xA0000-0xBFFFF	Standard VGA
Graphics Adapter	
Memory Address 0xF5F00000-0xF6FFFFFF	PCI bus
Memory Address 0xF5F00000-0xF6FFFFFF	Base System
Device	
I/O Port 0x000003B0-0x000003BB	PCI bus
I/O Port 0x000003B0-0x000003BB	Standard VGA
Graphics Adapter	
I/O Port 0x00004000-0x000044FF	PCI bus
I/O Port 0x00004000-0x000044FF	Smart Array
5300 Controller (Non-Miniport)	

[DMA]

Resource	Device	Status
Channel 7 Direct memory access controller		OK
Channel 2 Standard floppy disk controller		OK

[Forced Hardware]

Device PNP Device ID

[I/O]

Resource	Device	Status
0x00000000-0x00000CFF	PCI bus	OK
0x00000000-0x00000CFF	PCI bus	OK
0x00000000-0x00000CFF	Direct memory	
access controller		OK
0x000003B0-0x000003BB	PCI bus	OK
0x000003B0-0x000003BB	Standard VGA	
Graphics Adapter		OK
0x000003C0-0x000003DF	PCI bus	OK
0x000003C0-0x000003DF	Standard VGA	
Graphics Adapter		OK
0x00002400-0x000024FF	Standard VGA	
Graphics Adapter		OK
0x00001800-0x000018FF	Base System	Device OK
0x00002800-0x000028FF	Base System	Device OK
0x00000A79-0x00000A79	ISAPNP	Read Data Port
OK		
0x00000279-0x00000279	ISAPNP	Read Data Port
OK		
0x00000274-0x00000277	ISAPNP	Read Data Port
OK		
0x00000F50-0x00000F58	Motherboard	resources
OK		
0x00000408-0x0000040F	Motherboard	resources
OK		
0x00000092-0x00000092	Motherboard	resources
OK		


```

0xF79C0000-0xF79FFFFF Smart Array 5300
Controller (Non-Miniport) OK
0xF7C00000-0xF7FFFFF PCI bus OK
0xF7C00000-0xF7FFFFF Smart Array 5300
Controller (Non-Miniport) OK
0xF7FC0000-0xF7FFFFF Smart Array 5300
Controller (Non-Miniport) OK
0xF7E00000-0xF7EFFFFF Smart Array 5300
Controller (Non-Miniport) OK
0xF7DC0000-0xF7DFFFFF Smart Array 5300
Controller (Non-Miniport) OK

```

[Components]

[Multimedia]

[Audio Codecs]

CODEC	Manufacturer	Description	Status	File	Version	Size
c:\windows.0\system32\l3codeca.acm	Fraunhofer Institut Integrierte Schaltungen IIS	Fraunhofer IIS MPEG Layer-3 Codec	OK	C:\WINDOWS.0\system32\L3CODECA.ACM	1, 9, 0, 0305	284.00 KB (290,816 bytes)
c:\windows.0\system32\sl_anet.acm	Sipro Lab Telecom Inc.	Sipro Lab Telecom Audio Codec	OK	C:\WINDOWS.0\system32\SL_ANET.ACM	3.02	84.00 KB (86,016 bytes)
c:\windows.0\system32\msaud32.acm	Microsoft Corporation	Windows Media Audio Codec	OK	C:\WINDOWS.0\system32\MSAUD32.ACM	8.00.00.4487	288.00 KB (294,912 bytes)
c:\windows.0\system32\imaadp32.acm	Microsoft Corporation	IMAADP32	OK	C:\WINDOWS.0\system32\IMAADP32.ACM	5.2.3790.0 (srv03_rtm.030324-2048)	15.50 KB (15,872 bytes)
c:\windows.0\system32\msadp32.acm	Microsoft Corporation	MSADP32	OK	C:\WINDOWS.0\system32\MSADP32.ACM	5.2.3790.0 (srv03_rtm.030324-2048)	14.50 KB (14,848 bytes)
c:\windows.0\system32\tssoft32.acm	DSP GROUP, INC.	TSSOFT32	OK	C:\WINDOWS.0\system32\TSSOFT32.ACM	1.01	9.50 KB (9,728 bytes)
c:\windows.0\system32\msg723.acm	Microsoft Corporation	MSG723	OK	C:\WINDOWS.0\system32\MSG723.ACM	4.4.4000	116.00 KB (118,784 bytes)

```

c:\windows.0\system32\msgsm32.acm Microsoft Corporation OK
C:\WINDOWS.0\system32\MSGSM32.ACM
5.2.3790.0 (srv03_rtm.030324-2048)
20.50 KB (20,992 bytes) 3/25/2003

```

```

12:00 AM
c:\windows.0\system32\msg711.acm Microsoft Corporation OK
C:\WINDOWS.0\system32\MSG711.ACM
5.2.3790.0 (srv03_rtm.030324-2048)
10.00 KB (10,240 bytes) 3/25/2003

```

[Video Codecs]

CODEC	Manufacturer	Description	Status	File	Version	Size
c:\windows.0\system32\msh263.drv	Microsoft Corporation	MSH263.DRV	OK	C:\WINDOWS.0\system32\MSH263.DRV	4.4.4000	284.00 KB (290,816 bytes)
c:\windows.0\system32\msyuv.dll	Microsoft Corporation	MSYUV.DLL	OK	C:\WINDOWS.0\system32\MSYUV.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	16.50 KB (16,896 bytes)
c:\windows.0\system32\mrle32.dll	Microsoft Corporation	MSRLE32.DLL	OK	C:\WINDOWS.0\system32\MSRLE32.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	10.50 KB (10,752 bytes)
c:\windows.0\system32\tsbyuv.dll	Microsoft Corporation	TSBYUV.DLL	OK	C:\WINDOWS.0\system32\TSBYUV.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	8.00 KB (8,192 bytes)
c:\windows.0\system32\msvidc32.dll	Microsoft Corporation	MSVIDC32.DLL	OK	C:\WINDOWS.0\system32\MSVIDC32.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	26.50 KB (27,136 bytes)
c:\windows.0\system32\msh261.drv	Microsoft Corporation	MSH261.DRV	OK	C:\WINDOWS.0\system32\MSH261.DRV	4.4.4000	180.00 KB (184,320 bytes)
c:\windows.0\system32\iyuv_32.dll	Microsoft Corporation	IYUV_32.DLL	OK	C:\WINDOWS.0\system32\IYUV_32.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	45.00 KB (46,080 bytes)

[CD-ROM]

Item	Value
Drive	D:

Description	Value
CD-ROM Drive	CD-ROM Drive
Media Loaded	No
Media Type	CD-ROM
Name	COMPAQ CD-ROM SC-148C
Manufacturer	(Standard CD-ROM drives)
Status	OK
Transfer Rate	Not Available
SCSI Target ID	0
PNP Device ID	IDE\CDROMCOMPAQ_CD-ROM_SC-148C
Driver	c:\windows.0\system32\drivers\cdrom.sys (5.2.3790.0 (srv03_rtm.030324-2048), 49.50 KB (50,688 bytes), 3/25/2003 12:00 AM)

[Sound Device]

Item	Value
Name	Standard VGA Graphics Adapter
PNP Device ID	PCI\VEN_1002&DEV_4752&SUBSYS_001E0E11&REV_27\3&267A616A&0&18
Adapter Type	ATI MACH64, (Standard display types) compatible
Adapter Description	Standard VGA Graphics Adapter
Adapter RAM	7.94 MB (8,323,072 bytes)
Installed Drivers	vga.dll,framebuf.dll,vga256.dll,vga64k.dll

[Display]

Item	Value
Name	Standard VGA Graphics Adapter
PNP Device ID	PCI\VEN_1002&DEV_4752&SUBSYS_001E0E11&REV_27\3&267A616A&0&18
Adapter Type	ATI MACH64, (Standard display types) compatible
Adapter Description	Standard VGA Graphics Adapter
Adapter RAM	7.94 MB (8,323,072 bytes)
Installed Drivers	vga.dll,framebuf.dll,vga256.dll,vga64k.dll
Driver Version	5.2.3790.0
INF File	display.inf (vga section)
Color Planes	1
Color Table Entries	256
Resolution	800 x 600 x 1 hertz
Bits/Pixel	8
Memory Address	0xF6000000-0xF6FFFFFF
I/O Port	0x00002400-0x000024FF
Memory Address	0xF5FF0000-0xF5FF0FFF
I/O Port	0x00003B00-0x00003BBB
I/O Port	0x00003C00-0x00003CFF
Memory Address	0xA0000-0xBFFFF
Driver	c:\windows.0\system32\drivers\vgapnp.sys (5.2.3790.0 (srv03_rtm.030324-2048), 23.00 KB (23,552 bytes), 12/1/2003 10:46 AM)

[Infrared]

Item	Value
Item	Value

[Input]

[Keyboard]

Item	Value
Description	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
Name	Enhanced (101- or 102-key)
Layout	00000409

PNP Device ID ACPI\PNP0303\4&35118DFF&0
Number of Function Keys 12
I/O Port 0x00000060-0x00000060
I/O Port 0x00000064-0x00000064
IRQ Channel IRQ 1
Driver c:\windows.0\system32\drivers\i8042prt.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 68.50 KB (70,144
bytes), 3/25/2003 12:00 AM)

[Pointing Device]

Item Value
Hardware Type PS/2 Compatible Mouse
Number of Buttons 5
Status OK
PNP Device ID ACPI\PNP0F13\4&35118DFF&0
Power Management Supported No
Double Click Threshold 6
Handedness Right Handed Operation
IRQ Channel IRQ 12
Driver c:\windows.0\system32\drivers\i8042prt.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 68.50 KB (70,144
bytes), 3/25/2003 12:00 AM)

[Modem]

Item Value

[Network]

[Adapter]

Item Value
Name [00000001] BCM5703 Gigabit Ethernet
Adapter Type Ethernet 802.3
Product Type BCM5703 Gigabit Ethernet
Installed Yes
PNP Device ID
PCI\VEN_14E4&DEV_16A7&SUBSYS_00CB0E11&REV_0
2\3&13C0B0C5&0&20
Last Reset 2/11/2004 3:16 PM
Index 1
Service Name b57w2k
IP Address 130.168.209.68
IP Subnet 255.255.0.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:02:A5:FF:2D:AE
Memory Address 0xF73D0000-0xF73DFFFF
IRQ Channel IRQ 29
Driver c:\windows.0\system32\drivers\b57xp32.sys
(2.91.0.0 built by: WinDDK, 137.00 KB (140,288
bytes), 11/24/2003 11:58 AM)
Name [00000002] RAS Async Adapter
Adapter Type Not Available
Product Type RAS Async Adapter
Installed Yes

PNP Device ID Not Available
Last Reset 2/11/2004 3:16 PM
Index 2
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000003] WAN Miniport (L2TP)
Adapter Type Not Available
Product Type WAN Miniport (L2TP)
Installed Yes
PNP Device ID ROOT\MS_L2TPMINIPOINT\0000
Last Reset 2/11/2004 3:16 PM
Index 3
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows.0\system32\drivers\rasl2tp.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 77.00 KB (78,848
bytes), 3/25/2003 12:00 AM)

Name [00000004] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPTP)
Installed Yes
PNP Device ID ROOT\MS_PPTPMINIPOINT\0000
Last Reset 2/11/2004 3:16 PM
Index 4
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30
Driver c:\windows.0\system32\drivers\raspttp.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 70.50 KB (72,192
bytes), 3/25/2003 12:00 AM)

Name [00000005] WAN Miniport (PPPOE)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPPOE)
Installed Yes
PNP Device ID ROOT\MS_PPPOEMINIPOINT\0000
Last Reset 2/11/2004 3:16 PM
Index 5
Service Name Rasppoe
IP Address Not Available
IP Subnet Not Available

Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30
Driver c:\windows.0\system32\drivers\rasppoe.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 38.00 KB (38,912
bytes), 3/25/2003 12:00 AM)

Name [00000006] Direct Parallel
Adapter Type Not Available
Product Type Direct Parallel
Installed Yes
PNP Device ID ROOT\MS_PTMINIPOINT\0000
Last Reset 2/11/2004 3:16 PM
Index 6
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows.0\system32\drivers\raspti.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 18.50 KB (18,944
bytes), 3/25/2003 12:00 AM)

Name [00000007] WAN Miniport (IP)
Adapter Type Not Available
Product Type WAN Miniport (IP)
Installed Yes
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 2/11/2004 3:16 PM
Index 7
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows.0\system32\drivers\ndiswan.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 96.50 KB (98,816
bytes), 3/25/2003 12:00 AM)

[Protocol]

Item Value
Name MSFDF Tcpip [TCP/IP]
Connectionless Service No
Guarantees Delivery Yes
Guarantees Sequencing Yes
Maximum Address Size 16 bytes
Maximum Message Size 0 bytes
Message Oriented No
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting No

Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data Yes
 Supports Graceful Closing Yes
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

Name MSAFD Tcpip [UDP/IP]
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 16 bytes
 Maximum Message Size 63.93 KB (65,467 bytes)

Message Oriented Yes
 Minimum Address Size 16 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting Yes
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting Yes

Name RSVP UDP Service Provider
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 16 bytes
 Maximum Message Size 63.93 KB (65,467 bytes)

Message Oriented Yes
 Minimum Address Size 16 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting Yes
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption Yes
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting Yes

Name RSVP TCP Service Provider
 Connectionless Service No
 Guarantees Delivery Yes
 Guarantees Sequencing Yes
 Maximum Address Size 16 bytes
 Maximum Message Size 0 bytes
 Message Oriented No
 Minimum Address Size 16 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting No
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption Yes
 Supports Expedited Data Yes
 Supports Graceful Closing Yes
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{C251CEC3-55A8-4845-974D-85EE861FF1D0}] SEQPACKET 0
 Connectionless Service No
 Guarantees Delivery Yes
 Guarantees Sequencing Yes
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting No
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{C251CEC3-55A8-4845-974D-85EE861FF1D0}] DATAGRAM 0
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting Yes
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{2DF02B57-AA9F-4A4D-B2E5-F44971983A39}] SEQPACKET 1
 Connectionless Service No
 Guarantees Delivery Yes
 Guarantees Sequencing Yes
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting No
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{2DF02B57-AA9F-4A4D-B2E5-F44971983A39}] DATAGRAM 1
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting Yes
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{B3C099D4-7BE7-4816-8C46-173736694F28}] SEQPACKET 2
 Connectionless Service No
 Guarantees Delivery Yes
 Guarantees Sequencing Yes
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting No
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{B3C099D4-7BE7-4816-8C46-173736694F28}] DATAGRAM 2
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting Yes
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

[WinSock]

```
Item Value
File c:\windows.0\system32\winsock.dll
Size 2.80 KB (2,864 bytes)
Version 3.10
```

```
File c:\windows.0\system32\wsock32.dll
Size 22.00 KB (22,528 bytes)
Version 5.2.3790.0 (srv03_rtm.030324-2048)
```

[Ports]

[Serial]

```
Item Value
Name Communications Port (COM1)
Status OK
PNP Device ID ACPI\PNP0501\0
Maximum Input Buffer Size 0
Maximum Output Buffer Size No
Settable Baud Rate Yes
Settable Data Bits Yes
Settable Flow Control Yes
Settable Parity Yes
Settable Parity Check Yes
Settable Stop Bits Yes
Settable RLSDD Yes
Supports RLSDD Yes
Supports 16 Bit Mode No
Supports Special Characters No
Baud Rate 9600
Bits/Byte 8
Stop Bits 1
Parity None
Busy No
Abort Read/Write on Error No
Binary Mode Enabled Yes
Continue XMit on XOff No
CTS Outflow Control No
Discard NULL Bytes No
DSR Outflow Control 0
DSR Sensitivity 0
DTR Flow Control Type Enable
EOF Character 0
Error Replace Character 0
Error Replacement Enabled No
Event Character 0
Parity Check Enabled No
RTS Flow Control Type Enable
XOff Character 19
XOffXMit Threshold 512
XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control 0
IRQ Channel IRQ 4
I/O Port 0x000003F8-0x000003FF
```

```
Driver c:\windows.0\system32\drivers\serial.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 76.00 KB (77,824
bytes), 3/25/2003 12:00 AM)
```

```
Name Communications Port (COM2)
Status OK
PNP Device ID ACPI\PNP0501\1
Maximum Input Buffer Size 0
Maximum Output Buffer Size No
Settable Baud Rate Yes
Settable Data Bits Yes
Settable Flow Control Yes
Settable Parity Yes
Settable Parity Check Yes
Settable Stop Bits Yes
Settable RLSDD Yes
Supports RLSDD Yes
Supports 16 Bit Mode No
Supports Special Characters No
Baud Rate 9600
Bits/Byte 8
Stop Bits 1
Parity None
Busy No
Abort Read/Write on Error No
Binary Mode Enabled Yes
Continue XMit on XOff No
CTS Outflow Control No
Discard NULL Bytes No
DSR Outflow Control 0
DSR Sensitivity 0
DTR Flow Control Type Enable
EOF Character 0
Error Replace Character 0
Error Replacement Enabled No
Event Character 0
Parity Check Enabled No
RTS Flow Control Type Enable
XOff Character 19
XOffXMit Threshold 512
XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control 0
IRQ Channel IRQ 3
I/O Port 0x000002F8-0x000002FF
Driver c:\windows.0\system32\drivers\serial.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 76.00 KB (77,824
bytes), 3/25/2003 12:00 AM)
```

[Parallel]

```
Item Value
Name LPT1
PNP Device ID ACPI\PNP0400\5&13237358&0
I/O Port 0x00000378-0x0000037F
Driver c:\windows.0\system32\drivers\parport.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 76.50 KB (78,336
bytes), 3/24/2003 5:04 PM)
```

[Storage]

[Drives]

```
Item Value
Drive A:
Description 3 1/2 Inch Floppy Drive

Drive C:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 16.94 GB (18,186,092,544 bytes)
Free Space 12.15 GB (13,045,559,296 bytes)
```

```
Volume Name
Volume Serial Number 000CAE0C
```

```
Drive D:
Description CD-ROM Disc
```

```
Drive E:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
```

```
Drive F:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
```

```
Drive G:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
```

```
Drive H:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available
```

```
Drive I:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
```

Volume Serial Number Not Available

Drive J:
 Description Local Fixed Disk
 Compressed Not Available
 File System Not Available
 Size Not Available
 Free Space Not Available
 Volume Name Not Available
 Volume Serial Number Not Available

Drive X:
 Description Network Connection
 Provider Name \\inforb\mount

Drive Y:
 Description Local Fixed Disk
 Compressed No
 File System NTFS
 Size 401.57 GB (431,185,592,320 bytes)
 Free Space 401.50 GB (431,104,860,160 bytes)

Volume Name backup2
 Volume Serial Number 54FFF1EB

Drive Z:
 Description Local Fixed Disk
 Compressed No
 File System NTFS
 Size 435.75 GB (467,886,792,704 bytes)
 Free Space 435.68 GB (467,804,938,240 bytes)

Volume Name backup1
 Volume Serial Number BCF7E3BA

[Disks]

Item	Value
Description	\\.\PHYSICALDRIVE1
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	Yes
Media Type	Fixed hard disk
Partitions	1
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	146.48 GB (157,283,804,160 bytes)
Total Cylinders	19,122
Total Sectors	307,194,930
Total Tracks	4,876,110
Tracks/Cylinder	255
Partition Disk #1, Partition #0	
Partition Size	146.48 GB (157,283,254,272 bytes)
Partition Starting Offset	65,536 bytes
Description	\\.\PHYSICALDRIVE2
Manufacturer	Not Available
Model	Not Available

Bytes/Sector	512
Media Loaded	Yes
Media Type	Fixed hard disk
Partitions	1
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	401.58 GB (431,193,853,440 bytes)
Total Cylinders	52,423
Total Sectors	842,175,495
Total Tracks	13,367,865
Tracks/Cylinder	255
Partition Disk #2, Partition #0	
Partition Size	401.57 GB (431,185,595,904 bytes)
Partition Starting Offset	32,256 bytes
Description	\\.\PHYSICALDRIVE4
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	Yes
Media Type	Fixed hard disk
Partitions	1
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	78.12 GB (83,881,405,440 bytes)
Total Cylinders	10,198
Total Sectors	163,830,870
Total Tracks	2,600,490
Tracks/Cylinder	255
Partition Disk #4, Partition #0	
Partition Size	78.12 GB (83,880,837,120 bytes)
Partition Starting Offset	65,536 bytes
Description	\\.\PHYSICALDRIVE5
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	Yes
Media Type	Fixed hard disk
Partitions	1
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	78.12 GB (83,881,405,440 bytes)
Total Cylinders	10,198
Total Sectors	163,830,870
Total Tracks	2,600,490
Tracks/Cylinder	255
Partition Disk #5, Partition #0	
Partition Size	78.12 GB (83,880,837,120 bytes)
Partition Starting Offset	65,536 bytes

Description	\\.\PHYSICALDRIVE6
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	Yes
Media Type	Fixed hard disk
Partitions	1
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	78.12 GB (83,881,405,440 bytes)
Total Cylinders	10,198
Total Sectors	163,830,870
Total Tracks	2,600,490
Tracks/Cylinder	255
Partition Disk #6, Partition #0	
Partition Size	78.12 GB (83,880,837,120 bytes)
Partition Starting Offset	65,536 bytes
Description	\\.\PHYSICALDRIVE7
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	Yes
Media Type	Fixed hard disk
Partitions	1
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	435.76 GB (467,895,052,800 bytes)
Total Cylinders	56,885
Total Sectors	913,857,525
Total Tracks	14,505,675
Tracks/Cylinder	255
Partition Disk #7, Partition #0	
Partition Size	435.75 GB (467,886,795,264 bytes)
Partition Starting Offset	32,256 bytes
Description	\\.\PHYSICALDRIVE0
Manufacturer	Not Available
Model	Not Available
Bytes/Sector	512
Media Loaded	Yes
Media Type	Fixed hard disk
Partitions	1
SCSI Bus	Not Available
SCSI Logical Unit	Not Available
SCSI Port	Not Available
SCSI Target ID	Not Available
Sectors/Track	63
Size	169.59 GB (182,091,248,640 bytes)
Total Cylinders	22,138
Total Sectors	355,646,970
Total Tracks	5,645,190
Tracks/Cylinder	255
Partition Disk #0, Partition #0	

Partition Size 126.95 GB (136,308,588,544 bytes)

Partition Starting Offset 65,536 bytes

Description \\.\PHYSICALDRIVE3
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1

SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 78.12 GB (83,881,405,440 bytes)
Total Cylinders 10,198
Total Sectors 163,830,870
Total Tracks 2,600,490
Tracks/Cylinder 255
Partition Disk #3, Partition #0
Partition Size 78.12 GB (83,880,837,120 bytes)

Partition Starting Offset 65,536 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ BF01885A34 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2

SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 2
SCSI Target ID 0
Sectors/Track 63
Size 16.95 GB (18,202,544,640 bytes)
Total Cylinders 2,213
Total Sectors 35,551,845
Total Tracks 564,315
Tracks/Cylinder 255
Partition Disk #8, Partition #0
Partition Size 7.81 MB (8,193,024 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #8, Partition #1
Partition Size 16.94 GB (18,186,094,080 bytes)

Partition Starting Offset 8,225,280 bytes

[SCSI]

Item Value
Name Compaq 64-bit/66MHz Dual Channel Wide
Ultra3 SCSI Adapter
Manufacturer Adaptec
Status OK
PNP Device ID
PCI\VEN_9005&DEV_00C0&SUBSYS_F6200E11&REV_0
1\3&13C0B0C5&0&18
I/O Port 0x00003000-0x000034FF
Memory Address 0xF73F0000-0xF73F0FFF

IRQ Channel IRQ 30
Driver c:\windows.0\system32\drivers\adpuls60m.sys
(RTC_XP07 (lab01_n(storbuild).010917-1031), 99.63 KB
(102,016 bytes), 3/25/2003 12:00 AM)

Name Compaq 64-bit/66MHz Dual Channel Wide
Ultra3 SCSI Adapter
Manufacturer Adaptec
Status OK
PNP Device ID
PCI\VEN_9005&DEV_00C0&SUBSYS_F6200E11&REV_0
1\3&13C0B0C5&0&19
I/O Port 0x00003400-0x000034FF
Memory Address 0xF73E0000-0xF73E0FFF
IRQ Channel IRQ 31
Driver c:\windows.0\system32\drivers\adpuls60m.sys
(RTC_XP07 (lab01_n(storbuild).010917-1031), 99.63 KB
(102,016 bytes), 3/25/2003 12:00 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_0
2\3&1070020&0&08
Memory Address 0xF77C0000-0xF777FFFF
Memory Address 0xF7600000-0xF76FFFFF
I/O Port 0x00004000-0x000044FF
IRQ Channel IRQ 16
Driver c:\windows.0\system32\drivers\hpgcissb.sys
(5.7.62.32 built by: WinDDK, 38.00 KB (38,912 bytes),
12/1/2003 11:05 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_0
2\3&1070020&0&010
Memory Address 0xF75C0000-0xF75FFFFF
Memory Address 0xF7400000-0xF777FFFF
I/O Port 0x00004400-0x000044FF
IRQ Channel IRQ 18
Driver c:\windows.0\system32\drivers\hpgcissb.sys
(5.7.62.32 built by: WinDDK, 38.00 KB (38,912 bytes),
12/1/2003 11:05 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_0
2\3&29E81982&0&08
Memory Address 0xF7BC0000-0xF7BFFFFF
Memory Address 0xF7A00000-0xF7AFFFFF
I/O Port 0x00005000-0x000054FF
IRQ Channel IRQ 24
Driver c:\windows.0\system32\drivers\hpgcissb.sys
(5.7.62.32 built by: WinDDK, 38.00 KB (38,912 bytes),
12/1/2003 11:05 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_0
2\3&29E81982&0&10
Memory Address 0xF79C0000-0xF79FFFFF
Memory Address 0xF7800000-0xF7BFFFFF
I/O Port 0x00005400-0x000054FF
IRQ Channel IRQ 26
Driver c:\windows.0\system32\drivers\hpgcissb.sys
(5.7.62.32 built by: WinDDK, 38.00 KB (38,912 bytes),
12/1/2003 11:05 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_0
2\3&172E68DD&0&08
Memory Address 0xF7FC0000-0xF7FFFFFF
Memory Address 0xF7E00000-0xF7EFFFFF
I/O Port 0x00006000-0x000064FF
IRQ Channel IRQ 20
Driver c:\windows.0\system32\drivers\hpgcissb.sys
(5.7.62.32 built by: WinDDK, 38.00 KB (38,912 bytes),
12/1/2003 11:05 AM)

Name Smart Array 5300 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_0
2\3&172E68DD&0&10
Memory Address 0xF7DC0000-0xF7DFFFFFF
Memory Address 0xF7C00000-0xF7FFFFFF
I/O Port 0x00006400-0x000064FF
IRQ Channel IRQ 22
Driver c:\windows.0\system32\drivers\hpgcissb.sys
(5.7.62.32 built by: WinDDK, 38.00 KB (38,912 bytes),
12/1/2003 11:05 AM)

[IDE]

Item Value
Name CSB5 IDE Controller
Manufacturer ServerWorks
Status OK
PNP Device ID
PCI\VEN_1166&DEV_0212&SUBSYS_02121166&REV_9
3\3&267A616A&0&79
I/O Port 0x00002000-0x0000200F
Driver c:\windows.0\system32\drivers\pciide.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 5.50 KB (5,632
bytes), 3/25/2003 12:00 AM)
Name Primary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI
controllers)

```

Status OK
PNP Device ID PCI\VEN_1024D5C6&0&0

I/O Port 0x000001F0-0x000001F7
I/O Port 0x000003F6-0x000003F6
IRQ Channel IRQ 14
Driver c:\windows.0\system32\drivers\atapi.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 89.00 KB (91,136
bytes), 3/25/2003 12:00 AM)

Name Secondary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI
controllers)
Status OK
PNP Device ID PCI\VEN_1024D5C6&0&1

I/O Port 0x00000170-0x00000177
I/O Port 0x00000376-0x00000376
Driver c:\windows.0\system32\drivers\atapi.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 89.00 KB (91,136
bytes), 3/25/2003 12:00 AM)

[Printing]

Name Driver Port Name Server Name

[Problem Devices]

Device PNP Device ID Error Code
Base System Device
PCI\VEN_0E11&DEV_B203&SUBSYS_B2060E11&REV_0
1\3&267A616A&0&20 The drivers for this device are
not installed.
Base System Device
PCI\VEN_0E11&DEV_B204&SUBSYS_B2060E11&REV_0
1\3&267A616A&0&22 The drivers for this device are
not installed.

[USB]

Device PNP Device ID

[Software Environment]

[System Drivers]

Name Description File Type
Started Start Mode State
Status Error Control Accept Pause
Accept Stop
abiosdsk Abiosdsk Not Available Kernel Driver
No Disabled Stopped OK
Ignore No No
acpi Microsoft ACPI Driver
c:\windows.0\system32\drivers\acpi.sys
Kernel Driver Yes Boot
Running OK Normal No Yes
acpiec ACPIEC
c:\windows.0\system32\drivers\acpiec.sys
Kernel Driver No Disabled

```

```

Stopped OK Normal No No
adpul60m adpul60m
c:\windows.0\system32\drivers\adpul60m.sys
Kernel Driver Yes Boot
Running OK Normal No Yes
adpu320 adpu320 Not Available Kernel Driver
No Disabled Stopped OK
Normal No No
afcnt afcnt Not Available Kernel Driver
No Disabled Stopped OK
Normal No No
afd AFD Networking Support Environment
c:\windows.0\system32\drivers\afd.sys
Kernel Driver Yes Auto
Running OK Normal No Yes
ahal54x Ahal54x Not Available Kernel Driver
No Disabled Stopped OK
Normal No No
aic78u2 aic78u2 Not Available Kernel Driver
No Disabled Stopped OK
Normal No No
aic78xx aic78xx Not Available Kernel Driver
No Disabled Stopped OK
Normal No No
aliide AliIde Not Available Kernel Driver
No Disabled Stopped OK
Normal No No
asynccmac RAS Asynchronous Media Driver
c:\windows.0\system32\drivers\asynccmac.sys
Kernel Driver No Manual
Stopped OK Normal No No
atapi Standard IDE/ESDI Hard Disk Controller
c:\windows.0\system32\drivers\atapi.sys
Kernel Driver Yes Boot
Running OK Normal No Yes
atdisk Atdisk Not Available Kernel Driver
No Disabled Stopped OK
Ignore No No
ati2mpad ati2mpad
c:\windows.0\system32\drivers\ati2mpad.sys
Kernel Driver No Manual
Stopped OK Ignore No No
atmarpc ATM ARP Client Protocol
c:\windows.0\system32\drivers\atmarpc.sys
Kernel Driver No Manual
Stopped OK Normal No No
audstub Audio Stub Driver
c:\windows.0\system32\drivers\audstub.sys
Kernel Driver Yes Manual
Running OK Normal No Yes
b57w2k BCM5703 Gigabit Ethernet
c:\windows.0\system32\drivers\b57xp32.sys
Kernel Driver Yes Manual
Running OK Normal No Yes

```

```

beep Beep
c:\windows.0\system32\drivers\beep.sys
Kernel Driver Yes System
Running OK Normal No Yes
cbidf2k cbidf2k
c:\windows.0\system32\drivers\cbidf2k.sys
Kernel Driver No Disabled
Stopped OK Normal No No
cd20xrnt cd20xrnt Not Available Kernel Driver
No Disabled Stopped OK
Normal No No
cdfc Cdfs
c:\windows.0\system32\drivers\cdfc.sys
File System Driver Yes Disabled
Running OK Normal No Yes
cdrom CD-ROM Driver
c:\windows.0\system32\drivers\cdrom.sys
Kernel Driver Yes System
Running OK Normal No Yes
changer Changer Not Available Kernel Driver
No System Stopped OK
Ignore No No
clusdisk Cluster Disk Driver
c:\windows.0\system32\drivers\clusdisk.sys
Kernel Driver No Disabled
Stopped OK Normal No No
cmdide CmdIde Not Available Kernel Driver
No Disabled Stopped OK
Normal No No
cpqarray Cpqarray Not Available Kernel Driver
No Disabled Stopped OK
Normal No No
cpqarray2 Cpqarray2 Not Available Kernel Driver
No Disabled Stopped OK
Normal No No
cpqcissm cpqcissm
c:\windows.0\system32\drivers\cpqcissm.sys
Kernel Driver Yes Boot
Running OK Normal No Yes
cpqfcalm cpqfcalm Not Available Kernel Driver
No Disabled Stopped OK
Normal No No
crcdisk CRC Disk Filter Driver
c:\windows.0\system32\drivers\crcdisk.sys
Kernel Driver Yes Boot
Running OK Normal No Yes
dac960nt dac960nt Not Available Kernel Driver
No Disabled Stopped OK
Normal No No
dellcerc dellcerc Not Available Kernel Driver
No Disabled Stopped OK
Normal No No
dfsdriver DfsDriver
c:\windows.0\system32\drivers\dfs.sys
File System Driver Yes Boot

```

	Running	OK	Normal	No	Yes
disk	Disk Driver c:\windows.0\system32\drivers\disk.sys Kernel Driver Yes Boot Running OK Normal No Yes				
dmboot	dmboot c:\windows.0\system32\drivers\dmboot.sys Kernel Driver No Disabled Stopped OK Normal No No				
dmio	Logical Disk Manager Driver c:\windows.0\system32\drivers\dmio.sys Kernel Driver Yes Boot Running OK Normal No Yes				
dmload	dmload c:\windows.0\system32\drivers\dmload.sys Kernel Driver Yes Boot Running OK Normal No Yes				
dpti2o	dpti2o	Not Available		Kernel Driver	
		Disabled	Stopped	OK	
fastfat	Fastfat c:\windows.0\system32\drivers\fastfat.sys File System Driver No Disabled Stopped OK Normal No No				
fdc	Floppy Disk Controller Driver c:\windows.0\system32\drivers\fdc.sys Kernel Driver Yes Manual Running OK Normal No Yes				
fips	Fips c:\windows.0\system32\drivers\fips.sys Kernel Driver Yes System Running OK Normal No Yes				
flpydisk	Floppy Disk Driver c:\windows.0\system32\drivers\flpydisk.sys Kernel Driver Yes Manual Running OK Normal No Yes				
ftdisk	Volume Manager Driver c:\windows.0\system32\drivers\ftdisk.sys Kernel Driver Yes Boot Running OK Normal No Yes				
gpc	Generic Packet Classifier c:\windows.0\system32\drivers\msgpc.sys Kernel Driver Yes Manual Running OK Normal No Yes				
hpn	hpn	Not Available		Kernel Driver	
		Disabled	Stopped	OK	
hpqcissb	Smart Array Controllers Non-Miniport Bus Driver c:\windows.0\system32\drivers\hpqcissb.sys Kernel Driver Yes Boot Running OK Normal No Yes				

hpqcissd	Smart Array Controllers Non-Miniport Disk Driver c:\windows.0\system32\drivers\hpqcissd.sys Kernel Driver Yes Boot Running OK Normal No Yes				
hpt3xx	hpt3xx	Not Available		Kernel Driver	
		Disabled	Stopped	OK	
http	HTTP c:\windows.0\system32\drivers\http.sys Kernel Driver No Manual Stopped OK Normal No No				
i2omgmt	i2omgmt	Not Available		Kernel Driver	
		System	Stopped	OK	
		Normal	No	No	
i2omp	i2omp	Not Available		Kernel Driver	
		Disabled	Stopped	OK	
		Normal	No	No	
i8042prt	18042 Keyboard and PS/2 Mouse Port Driver c:\windows.0\system32\drivers\i8042prt.sys Kernel Driver Yes System Running OK Normal No Yes				
iirsp	iirsp	Not Available		Kernel Driver	
		Disabled	Stopped	OK	
		Normal	No	No	
imapi	CD-Burning Filter Driver c:\windows.0\system32\drivers\imapi.sys Kernel Driver No System Stopped OK Normal No No				
intelide	IntelIde	Not Available		Kernel Driver	
		Disabled	Stopped	OK	
		Normal	No	No	
ipfilterdriver	IP Traffic Filter Driver c:\windows.0\system32\drivers\ipfltdrv.sys Kernel Driver No Manual Stopped OK Normal No No				
ipinip	IP in IP Tunnel Driver c:\windows.0\system32\drivers\ipinip.sys Kernel Driver No Manual Stopped OK Normal No No				
ipnat	IP Network Address Translator c:\windows.0\system32\drivers\ipnat.sys Kernel Driver No Manual Stopped OK Normal No No				
ipsec	IPSEC driver c:\windows.0\system32\drivers\ipsec.sys Kernel Driver Yes System Running OK Normal No Yes				
ipsraidn	ipsraidn	Not Available		Kernel Driver	
		Disabled	Stopped	OK	
		Normal	No	No	
isapnp	PnP ISA/EISA Bus Driver c:\windows.0\system32\drivers\isapnp.sys Kernel Driver Yes Boot Running OK Critical No Yes				

kbdclass	Keyboard Class Driver c:\windows.0\system32\drivers\kbdclass.sys Kernel Driver Yes System Running OK Normal No Yes				
ksecdd	KSecDD c:\windows.0\system32\drivers\ksecdd.sys Kernel Driver Yes Boot Running OK Normal No Yes				
lp6nds35	lp6nds35	Not Available		Kernel Driver	
		Disabled	Stopped	OK	
		Normal	No	No	
mmdd	mmdd c:\windows.0\system32\drivers\mmdd.sys Kernel Driver Yes System Running OK Ignore No Yes				
modem	Modem c:\windows.0\system32\drivers\modem.sys Kernel Driver No Manual Stopped OK Ignore No No				
mouclass	Mouse Class Driver c:\windows.0\system32\drivers\mouclass.sys Kernel Driver Yes System Running OK Normal No Yes				
mountmgr	Mount Point Manager c:\windows.0\system32\drivers\mountmgr.sys Kernel Driver Yes Boot Running OK Normal No Yes				
mrraid35x	mrraid35x	Not Available		Kernel Driver	
		Disabled	Stopped	OK	
		Normal	No	No	
mrxdav	WebDav Client Redirector c:\windows.0\system32\drivers\mrxdav.sys File System Driver No Manual Stopped OK Normal No No				
mrxsmb	MRXSMB c:\windows.0\system32\drivers\mrxsmb.sys File System Driver Yes System Running OK Normal No Yes				
msfs	Msfs c:\windows.0\system32\drivers\msfs.sys File System Driver Yes System Running OK Normal No Yes				
mup	Mup c:\windows.0\system32\drivers\mup.sys File System Driver Yes Boot Running OK Normal No Yes				
ndis	NDIS System Driver c:\windows.0\system32\drivers\ndis.sys Kernel Driver Yes Boot Running OK Normal No Yes				
ndistapi	Remote Access NDIS TAPI Driver c:\windows.0\system32\drivers\ndistapi.sys				

	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
ndisuiio	NDIS Usermode I/O Protocol				
	c:\windows.0\system32\drivers\ndisuiio.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
ndiswan	Remote Access NDIS WAN Driver				
	c:\windows.0\system32\drivers\ndiswan.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
ndproxy	NDIS Proxy				
	c:\windows.0\system32\drivers\ndproxy.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
netbios	NetBIOS Interface				
	c:\windows.0\system32\drivers\netbios.sys				
	File System Driver	Yes	System		
	Running	OK	Normal	No	Yes
netbt	NetBios over Tcpip				
	c:\windows.0\system32\drivers\netbt.sys				
	Kernel Driver	Yes	System		
	Running	OK	Normal	No	Yes
nfrd960	nfrd960	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
npfs	Npfs				
	c:\windows.0\system32\drivers\npfs.sys				
	File System Driver	Yes	System		
	Running	OK	Normal	No	Yes
ntfs	Ntfs				
	c:\windows.0\system32\drivers\ntfs.sys				
	File System Driver	Yes	Disabled		
	Running	OK	Normal	No	Yes
null	Null				
	c:\windows.0\system32\drivers\null.sys				
	Kernel Driver	Yes	System		
	Running	OK	Normal	No	Yes
parport	Parallel port driver				
	c:\windows.0\system32\drivers\parport.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
partmgr	Partition Manager				
	c:\windows.0\system32\drivers\partmgr.sys				
	Kernel Driver	Yes	Boot		
	Running	OK	Normal	No	Yes
parvdm	Parvdm				
	c:\windows.0\system32\drivers\parvdm.sys				
	Kernel Driver	Yes	Auto		
	Running	OK	Ignore	No	Yes
pci	PCI Bus Driver				
	c:\windows.0\system32\drivers\pci.sys				

	Kernel Driver	Yes	Boot		
	Running	OK	Critical	No	Yes
pciide	PCIIde				
	c:\windows.0\system32\drivers\pciide.sys				
	Kernel Driver	Yes	Boot		
	Running	OK	Normal	No	Yes
pcmcia	Pcmcia				
	c:\windows.0\system32\drivers\pcmcia.sys				
	Kernel Driver	No	Disabled		
	Stopped	OK	Normal	No	No
pdcomp	PDCOMP	Not Available		Kernel Driver	
	No	Manual	Stopped	OK	
	Ignore	No	No		
pdframe	PDFRAME	Not Available		Kernel Driver	
	No	Manual	Stopped	OK	
	Ignore	No	No		
pdreli	PDRELI	Not Available		Kernel Driver	
	No	Manual	Stopped	OK	
	Ignore	No	No		
pdrframe	PDRFRAME	Not Available		Kernel Driver	
	No	Manual	Stopped	OK	
	Ignore	No	No		
perc2	perc2	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
perc2hib	perc2hib	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
pptpminiport	WAN Miniport (PPTP)				
	c:\windows.0\system32\drivers\rasptp.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
processor	Processor Driver				
	c:\windows.0\system32\drivers\processr.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
ptilink	Direct Parallel Link Driver				
	c:\windows.0\system32\drivers\ptilink.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
ql1080	ql1080	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
ql10wnt	Ql10wnt	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
ql12160	ql12160	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
ql1240	ql1240	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
ql1280	ql1280	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		

ql2100	ql2100	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
ql2200	ql2200	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
ql2300	ql2300	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	
	Normal	No	No		
rasacd	Remote Access Auto Connection Driver				
	c:\windows.0\system32\drivers\rasacd.sys				
	Kernel Driver	Yes	System		
	Running	OK	Normal	No	Yes
rasl2tp	WAN Miniport (L2TP)				
	c:\windows.0\system32\drivers\rasl2tp.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
raspppoe	Remote Access PPPOE Driver				
	c:\windows.0\system32\drivers\raspppoe.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
raspti	Direct Parallel				
	c:\windows.0\system32\drivers\raspti.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
rdbss	Rdbss				
	c:\windows.0\system32\drivers\rdbss.sys				
	File System Driver	Yes	System		
	Running	OK	Normal	No	Yes
rdpccd	RDPCCD				
	c:\windows.0\system32\drivers\rdpccd.sys				
	Kernel Driver	Yes	System		
	Running	OK	Ignore	No	Yes
rdpdr	Terminal Server Device Redirector Driver				
	c:\windows.0\system32\drivers\rdpdr.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
rdpwd	RDPWD				
	c:\windows.0\system32\drivers\rdpwd.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Ignore	No	Yes
redbook	Digital CD Audio Playback Filter Driver				
	c:\windows.0\system32\drivers\redbook.sys				
	Kernel Driver	Yes	System		
	Running	OK	Normal	No	Yes
secdrv	Secdrv				
	c:\windows.0\system32\drivers\secdrv.sys				
	Kernel Driver	No	Manual		
	Stopped	OK	Normal	No	No
serenum	Serenum Filter Driver				
	c:\windows.0\system32\drivers\serenum.sys				
	Kernel Driver	Yes	Manual		

	Running	OK	Normal	No	Yes
serial	Serial port driver c:\windows.0\system32\drivers\serial.sys Kernel Driver Yes System Running OK Ignore No Yes				
sfloppy	Sfloppy c:\windows.0\system32\drivers\sfloppy.sys Kernel Driver No System Stopped OK Ignore No No				
simbad	Simbad Not Available Kernel Driver No Disabled Stopped OK Normal No No				
sparrow	Sparrow Not Available Kernel Driver No Disabled Stopped OK Normal No No				
srv	Srv c:\windows.0\system32\drivers\srv.sys File System Driver Yes Manual Running OK Normal No Yes				
swenum	Software Bus Driver c:\windows.0\system32\drivers\swenum.sys Kernel Driver Yes Manual Running OK Normal No Yes				
symc810	symc810 Not Available Kernel Driver No Disabled Stopped OK Normal No No				
symc8xx	symc8xx Not Available Kernel Driver No Disabled Stopped OK Normal No No				
symmpi	symmpi Not Available Kernel Driver No Disabled Stopped OK Normal No No				
sym_hi	sym_hi Not Available Kernel Driver No Disabled Stopped OK Normal No No				
sym_u3	sym_u3 Not Available Kernel Driver No Disabled Stopped OK Normal No No				
tcpip	TCP/IP Protocol Driver c:\windows.0\system32\drivers\tcpip.sys Kernel Driver Yes System Running OK Normal No Yes				
tdpipe	TDPIPE c:\windows.0\system32\drivers\tdpipe.sys Kernel Driver No Manual Stopped OK Ignore No No				
tdtcp	TDTCP c:\windows.0\system32\drivers\tdtcp.sys Kernel Driver Yes Manual Running OK Ignore No Yes				
termdd	Terminal Device Driver c:\windows.0\system32\drivers\termdd.sys Kernel Driver Yes System Running OK Normal No Yes				

toside	TosIde	Not Available	Kernel Driver
	No	Disabled Stopped	OK
udfs	Udfs	Normal No No	
	c:\windows.0\system32\drivers\udfs.sys	File System Driver No Disabled	Stopped OK Normal No No
ultra	ultra	Not Available	Kernel Driver
	No	Disabled Stopped	OK
	Normal No No		
update	Microcode Update Driver c:\windows.0\system32\drivers\update.sys Kernel Driver Yes Manual Running OK Normal No Yes		
usbhub	USB2 Enabled Hub c:\windows.0\system32\drivers\usbhub.sys Kernel Driver No Manual Stopped OK Normal No No		
usbohci	Microsoft USB Open Host Controller Miniport Driver c:\windows.0\system32\drivers\usbohci.sys Kernel Driver No Manual Stopped OK Normal No No		
vga	vga c:\windows.0\system32\drivers\vgapnp.sys Kernel Driver Yes Manual Running OK Ignore No Yes		
vgasave	VGA Display Controller. c:\windows.0\system32\drivers\vga.sys Kernel Driver No System Stopped OK ignore No No		
viaide	ViaIde	Not Available	Kernel Driver
	No	Disabled Stopped	OK
	Normal No No		
volsnap	Storage volumes c:\windows.0\system32\drivers\volsnap.sys Kernel Driver Yes Boot Running OK Normal No Yes		
wanarp	Remote Access IP ARP Driver c:\windows.0\system32\drivers\wanarp.sys Kernel Driver Yes Manual Running OK Normal No Yes		
wdica	WDICA	Not Available	Kernel Driver
	No	Manual Stopped	OK
	Ignore No No		
wlbs	Network Load Balancing c:\windows.0\system32\drivers\wlbs.sys Kernel Driver No Manual Stopped OK Normal No No		
[Signed Drivers]			
Device Name	Signed	Device Class	
	Driver Version	Driver Date	

Manufacturer	INF Name	Driver Name
Device ID		
Not Available	Not Available	Not Available
Not Available	Not Available	Not Available
Available	Not Available	Not Available
HTREE\ROOT\0		
ACPI Multiprocessor PC	Yes	COMPUTER
5.2.3790.0	10/1/2002 (Standard	
computers)	hal.inf	Not Available
ROOT\ACPI_HAL\0000		
Microsoft ACPI-Compliant System	Yes	
SYSTEM	5.2.3790.0	10/1/2002
Microsoft acpi.inf	Not Available	
ACPI_HAL\PNP0C08\0		
Processor	Yes	PROCESSOR 5.2.3790.0
10/1/2002 (Standard processor types)		
cpu.inf	Not Available	
ACPI\GENUINEINTEL_-		
_X86_FAMILY_15_MODEL_2\0		
Processor	Yes	PROCESSOR 5.2.3790.0
10/1/2002 (Standard processor types)		
cpu.inf	Not Available	
ACPI\GENUINEINTEL_-		
_X86_FAMILY_15_MODEL_2\1		
Processor	Yes	PROCESSOR 5.2.3790.0
10/1/2002 (Standard processor types)		
cpu.inf	Not Available	
ACPI\GENUINEINTEL_-		
_X86_FAMILY_15_MODEL_2\7		
PCI bus	Yes	SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)		
machine.inf	Not Available	
ACPI\PNP0A03\0		
ServerWorks (RCC) CMIC_LE Processor to PCI Bridge(*)		
Yes	SYSTEM	5.2.3790.0
10/1/2002 ServerWorks (RCC)	machine.inf	
Not Available		
PCI\VEN_1166&DEV_0014&SUBSYS_00000000&REV_3		
1\3&267A616A&0&00		
ServerWorks (RCC) CMIC_LE Processor to PCI Bridge(*)		
Yes	SYSTEM	5.2.3790.0
10/1/2002 ServerWorks (RCC)	machine.inf	
Not Available		
PCI\VEN_1166&DEV_0014&SUBSYS_00000000&REV_0		
0\3&267A616A&0&01		
ServerWorks (RCC) CMIC_LE Processor to PCI Bridge(*)		
Yes	SYSTEM	5.2.3790.0
10/1/2002 ServerWorks (RCC)	machine.inf	
Not Available		
PCI\VEN_1166&DEV_0014&SUBSYS_00000000&REV_0		
0\3&267A616A&0&02		
Standard VGA Graphics Adapter	Yes	DISPLAY
5.2.3790.0	10/1/2002 (Standard	
display types)	display.inf	Not Available
PCI\VEN_1002&DEV_4752&SUBSYS_001E0E11&REV_2		
7\3&267A616A&0&18		
Plug and Play Monitor	Yes	MONITOR
5.1.2001.0	6/6/2001	(Standard


```

monitor types) monitor.inf Not Available
DISPLAY\AV00402\4&89B5141&0&12345678&0&0&03

Base System Device Not Available UNKNOWN Not
Available Not Available Not Available Not
Available Not Available
PCI\VEN_0E11&DEV_B203&SUBSYS_B2060E11&REV_0
1\3&267A616A&0&20
Base System Device Not Available UNKNOWN Not
Available Not Available Not Available Not
Available Not Available
PCI\VEN_0E11&DEV_B204&SUBSYS_B2060E11&REV_0
1\3&267A616A&0&22
PCI standard ISA bridge Yes SYSTEM
5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available
PCI\VEN_1166&DEV_0201&SUBSYS_00000000&REV_9
3\3&267A616A&0&78
ISAPNP Read Data Port Yes SYSTEM
5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ISAPNP\READDATAPORT\0
Motherboard resources Yes SYSTEM
5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNPOC02\0
Programmable interrupt controller Yes
SYSTEM 5.2.3790.0 10/1/2002
(Standard system devices) machine.inf
Not Available
ACPI\PNP0000\4&35118DFF&0
System timer Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0100\4&35118DFF&0
Direct memory access controller Yes
SYSTEM 5.2.3790.0 10/1/2002
(Standard system devices) machine.inf
Not Available
ACPI\PNP0200\4&35118DFF&0
System speaker Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0800\4&35118DFF&0
Standard 101/102-Key or Microsoft Natural PS/2
Keyboard Yes KEYBOARD 5.2.3790.0
10/1/2002 (Standard keyboards)
keyboard.inf Not Available
ACPI\PNP0303\4&35118DFF&0
PS/2 Compatible Mouse Yes MOUSE
5.2.3790.0 10/1/2002 Microsoft
msmouse.inf Not Available
ACPI\PNP0F13\4&35118DFF&0
Extended IO Bus Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A06\4&35118DFF&0
Printer Port Yes PORTS 5.2.3790.0
10/1/2002 (Standard port types)
msports.inf Not Available
ACPI\PNP0400\5&13237358&0
Printer Port Logical Interface Yes
SYSTEM 5.2.3790.0 10/1/2002

```

```

(Standard system devices) machine.inf
Not Available
LPTENUM\MICROSOFTRAWPORT\6&BCCF519&0&LPT1

Communications Port Yes PORTS 5.2.3790.0
10/1/2002 (Standard port types)
msports.inf Not Available
ACPI\PNP0501\0
Communications Port Yes PORTS 5.2.3790.0
10/1/2002 (Standard port types)
msports.inf Not Available
ACPI\PNP0501\1
Standard floppy disk controller Yes FDC
5.2.3790.0 10/1/2002 (Standard
floppy disk controllers) fdc.inf Not Available
ACPI\PNP0700\5&13237358&0
Floppy disk drive Yes FLOPPYDISK
5.2.3790.0 10/1/2002 (Standard
floppy disk drives) flpydisk.inf Not Available
FDC\GENERIC_FLOPPY_DRIVE\6&1C650E5D&0&0
CSB5 IDE Controller Yes HDC 5.2.3790.0
10/1/2002 ServerWorks mshdc.inf Not
Available
PCI\VEN_1166&DEV_0212&SUBSYS_02121166&REV_9
3\3&267A616A&0&79
Primary IDE Channel Yes HDC 5.2.3790.0
10/1/2002 (Standard IDE ATA/ATAPI
controllers) mshdc.inf Not Available
PCIIDE\IDECHANNEL\4&1024D5C6&0&0
CD-ROM Drive Yes CDROM 5.2.3790.0
10/1/2002 (Standard CD-ROM drives)
cdrom.inf Not Available
IDE\CDROMCOMPAGQ_CD-ROM_SC-
148C_____B101_____5\FB0C83D&0&0.0.0
Secondary IDE Channel Yes HDC
5.2.3790.0 10/1/2002 (Standard IDE
ATA/ATAPI controllers) mshdc.inf Not Available
PCIIDE\IDECHANNEL\4&1024D5C6&0&1
Serverworks Champion CSB5 - SouthBridge 5 LPC Yes
SYSTEM 5.2.3790.0 10/1/2002
ServerWorks (RCC) machine.inf Not
Available
PCI\VEN_1166&DEV_0225&SUBSYS_00000000&REV_0
0\3&267A616A&0&7B
ServerWorks Grand Champion CIOB_X2 - I/O Bridge 133
Mhz Yes SYSTEM 5.2.3790.0
10/1/2002 ServerWorks (RCC) machine.inf
Not Available
PCI\VEN_1166&DEV_0101&SUBSYS_00000000&REV_0
5\3&267A616A&0&80
ServerWorks Grand Champion CIOB_X2 - I/O Bridge 133
Mhz Yes SYSTEM 5.2.3790.0
10/1/2002 ServerWorks (RCC) machine.inf
Not Available
PCI\VEN_1166&DEV_0101&SUBSYS_00000000&REV_0
5\3&267A616A&0&82
ServerWorks Grand Champion CIOB_X2 - I/O Bridge 133
Mhz Yes SYSTEM 5.2.3790.0
10/1/2002 ServerWorks (RCC) machine.inf
Not Available
PCI\VEN_1166&DEV_0101&SUBSYS_00000000&REV_0
5\3&267A616A&0&88

```

```

ServerWorks Grand Champion CIOB_X2 - I/O Bridge 133
Mhz Yes SYSTEM 5.2.3790.0
10/1/2002 ServerWorks (RCC) machine.inf
Not Available
PCI\VEN_1166&DEV_0101&SUBSYS_00000000&REV_0
5\3&267A616A&0&8A
PCI bus Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A03\1
Compaq 64-bit/66MHz Dual Channel Wide Ultra3 SCSI
Adapter Yes SCSIADAPTER 5.2.3790.0
10/1/2002 Adaptec pnp SCSI.inf Not
Available
PCI\VEN_9005&DEV_00C0&SUBSYS_F6200E11&REV_0
1\3&13C0B0C5&0&18
Disk drive Yes DISKDRIVE 5.2.3790.0
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_BF01885A34&REV_HP
B4\4&F4A740&0&000
Compaq StorageWorks/ProLiant Storage Subsystem Yes
SYSTEM 5.2.3790.0 10/1/2002
Compaq scsidev.inf Not Available
SCSI\PROCESSOR&VEN_COMPAQ&PROD_PROLIANT_4L2
I&REV_1.70\4&F4A740&0&0F0
Compaq 64-bit/66MHz Dual Channel Wide Ultra3 SCSI
Adapter Yes SCSIADAPTER 5.2.3790.0
10/1/2002 Adaptec pnp SCSI.inf Not
Available
PCI\VEN_9005&DEV_00C0&SUBSYS_F6200E11&REV_0
1\3&13C0B0C5&0&19
BCM5703 Gigabit Ethernet Yes NET
2.91.0.0 10/1/2002 Narrowcom netb57xp.inf
Not Available
PCI\VEN_14E4&DEV_16A7&SUBSYS_00CB0E11&REV_0
2\3&13C0B0C5&0&20
PCI bus Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A03\2
Smart Array 5300 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.32 4/8/2003
Hewlett-Packard oem0.inf Not Available
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_0
2\3&1070020&0&08
Smart Array Logical Volume No DISKDRIVE
5.6.56.32 4/8/2003 Hewlett-Packard
oem1.inf Not Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME
\4&33332AB6&0&0000004000000000
Smart Array 5300 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.32 4/8/2003
Hewlett-Packard oem0.inf Not Available
PCI\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_0
2\3&1070020&0&10
Smart Array Logical Volume No DISKDRIVE
5.6.56.32 4/8/2003 Hewlett-Packard
oem1.inf Not Available
HPQCISS\DISK&VEN_COMPAQ&PROD_LOGICAL_VOLUME
\4&16A16360&0&0000004000000000
Smart Array Logical Volume No DISKDRIVE
5.6.56.32 4/8/2003 Hewlett-Packard

```



```

NetBios over Tcpip Not Available LEGACYDRIVER
Not Available Not Available Not
Available Not Available Not Available
ROOT\LEGACY_NETBT\0000
Null Not Available LEGACYDRIVER Not
Available Not Available Not Available Not
Available Not Available ROOT\LEGACY_NULL\0000

Partition Manager Not Available LEGACYDRIVER
Not Available Not Available Not
Available Not Available Not Available
ROOT\LEGACY_PARTMGR\0000
Parvdm Not Available LEGACYDRIVER Not
Available Not Available Not Available Not
Available Not Available ROOT\LEGACY_PARVDM\0000

Remote Access Auto Connection Driver Not Available
LEGACYDRIVER Not Available Not
Available Not Available Not Available Not
Available ROOT\LEGACY_RASACD\0000
RDPCCD Not Available LEGACYDRIVER Not
Available Not Available Not Available Not
Available Not Available ROOT\LEGACY_RDPCCD\0000

RDPWD Not Available LEGACYDRIVER Not
Available Not Available Not Available Not
Available Not Available ROOT\LEGACY_RDPWD\0000

TCP/IP Protocol Driver Not Available
LEGACYDRIVER Not Available Not
Available Not Available Not Available Not
Available ROOT\LEGACY_TCPIP\0000
TDTCP Not Available LEGACYDRIVER Not
Available Not Available Not Available Not
Available Not Available ROOT\LEGACY_TDTCP\0000

volsnap Not Available LEGACYDRIVER Not
Available Not Available Not Available Not
Available Not Available
ROOT\LEGACY_VOLSNAP\0000
Remote Access IP ARP Driver Not Available
LEGACYDRIVER Not Available Not
Available Not Available Not Available Not
Available ROOT\LEGACY_WANARP\0000
Audio Codecs Yes MEDIA 5.2.3790.0
10/1/2002 (Standard system devices)
wave.inf Not Available
ROOT\MEDIA\MS_MMCM
Legacy Audio Drivers Yes MEDIA
5.2.3790.0 10/1/2002 (Standard
system devices) wave.inf Not Available
ROOT\MEDIA\MS_MMDRV
Media Control Devices Yes MEDIA
5.2.3790.0 10/1/2002 (Standard
system devices) wave.inf Not Available
ROOT\MEDIA\MS_MMCI
Legacy Video Capture Devices Yes MEDIA
5.2.3790.0 10/1/2002 (Standard
system devices) wave.inf Not Available
ROOT\MEDIA\MS_MMVCD
Video Codecs Yes MEDIA 5.2.3790.0
10/1/2002 (Standard system devices)

```

```

wave.inf Not Available
ROOT\MEDIA\MS_MMVID
WAN Miniport (L2TP) Yes NET 5.2.3790.0
10/1/2002 Microsoft netrasa.inf Not
Available ROOT\MS_L2TPMINIPOINT\0000
WAN Miniport (IP) Yes NET 5.2.3790.0
10/1/2002 Microsoft netrasa.inf Not
Available ROOT\MS_NDISWANIP\0000
WAN Miniport (PPPOE) Yes NET
5.2.3790.0 10/1/2002 Microsoft
netrasa.inf Not Available
ROOT\MS_PPPOEMINIPOINT\0000
WAN Miniport (PPTP) Yes NET 5.2.3790.0
10/1/2002 Microsoft netrasa.inf Not
Available ROOT\MS_PPTPMINIPOINT\0000
Direct Parallel Yes NET 5.2.3790.0
10/1/2002 Microsoft netrasa.inf Not
Available ROOT\MS_PTMINIPOINT\0000
Terminal Server Device Redirector Yes
SYSTEM 5.2.3790.0 10/1/2002
(Standard system devices) machine.inf
Not Available ROOT\RDPDR\0000
Terminal Server Keyboard Driver Yes
SYSTEM 5.2.3790.0 10/1/2002
(Standard system devices) machine.inf
Not Available ROOT\RDP_KBD\0000
Terminal Server Mouse Driver Yes
SYSTEM
5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ROOT\RDP_MOU\0000
Plug and Play Software Device Enumerator Yes
SYSTEM 5.2.3790.0 10/1/2002
(Standard system devices) machine.inf
Not Available ROOT\SYSTEM\0000
Microcode Update Device Yes
SYSTEM
5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ROOT\SYSTEM\0001

[Environment Variables]
Variable Value User Name
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
Path
%SystemRoot%\system32;%SystemR
oot%\System32\Wbem;C:\Program Files\Microsoft SQL
Server\80\Tools\BINN <SYSTEM>
windir %SystemRoot% <SYSTEM>
OS Windows_NT <SYSTEM>
PROCESSOR_ARCHITECTURE x86 <SYSTEM>
PROCESSOR_LEVEL 15 <SYSTEM>
PROCESSOR_IDENTIFIER x86 Family 15 Model 2
Stepping 5, GenuineIntel <SYSTEM>
PROCESSOR_REVISION 0205 <SYSTEM>
NUMBER_OF_PROCESSORS 4 <SYSTEM>
ClusterLog C:\WINDOWS.0\Cluster\cluster.log
<SYSTEM>
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF
;.WSH <SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>

```

```

TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\SYSTEM
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\SYSTEM
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\LOCAL SERVICE
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\LOCAL SERVICE
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\NETWORK SERVICE
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\NETWORK SERVICE
TEMP %USERPROFILE%\Local Settings\Temp
VY\Administrator
TMP %USERPROFILE%\Local Settings\Temp
VY\Administrator

[Print Jobs]
Document Size Owner Notify Status
Time Submitted Start Time
Until Time Elapsed Time
Pages Printed Job ID Priority
Parameters Driver Print
Processor Host Print Queue Data Type Name

[Network Connections]
Local Name Remote Name Type
Status User Name

[Running Tasks]
Name Path Process ID Priority Min
Working Set Max Working Set Start Time
Version Size File Date
system idle process Not Available 0 0
Not Available Not Available Not
Available Not Available Not Available Not
Available
system Not Available 4 8 0
1413120 Not Available Not Available
Not Available Not Available
smss.exe Not Available 388 11
204800 1413120 2/11/2004 3:16 PM Not
Available Not Available Not Available
csrss.exe Not Available 512 13 Not
Available Not Available 2/11/2004 3:16 PM Not
Available Not Available Not Available
winlogon.exe
c:\windows.0\system32\winlogon.exe 536
13 204800 1413120 2/11/2004
3:16 PM 5.2.3790.0 (srv03_rtm.030324-2048)
536.50 KB (549,376 bytes) 3/25/2003
12:00 AM
services.exe
c:\windows.0\system32\services.exe 580
9 204800 1413120 2/11/2004
3:16 PM 5.2.3790.0 (srv03_rtm.030324-2048)
102.00 KB (104,448 bytes) 3/25/2003
12:00 AM
lsass.exe c:\windows.0\system32\lsass.exe 592
9 204800 1413120 2/11/2004

```

```

3:16 PM 5.2.3790.0 (srv03_rtm.030324-2048)
13.00 KB (13,312 bytes) 3/25/2003
12:00 AM
svchost.exe c:\windows.0\system32\svchost.exe
752 8 204800 1413120
2/11/2004 3:16 PM 5.2.3790.0
(srv03_rtm.030324-2048) 13.00 KB (13,312 bytes)
3/25/2003 12:00 AM
svchost.exe c:\windows.0\system32\svchost.exe
804 8 204800 1413120
2/11/2004 3:16 PM 5.2.3790.0
(srv03_rtm.030324-2048) 13.00 KB (13,312 bytes)
3/25/2003 12:00 AM
svchost.exe Not Available 952 8
Not Available Not Available
2/11/2004 3:16 PM Not Available Not
Available Not Available
svchost.exe Not Available 1000 8
Not Available Not Available
2/11/2004 3:16 PM Not Available Not
Available Not Available
svchost.exe c:\windows.0\system32\svchost.exe
1012 8 204800 1413120
2/11/2004 3:16 PM 5.2.3790.0
(srv03_rtm.030324-2048) 13.00 KB (13,312 bytes)
3/25/2003 12:00 AM
spoolsv.exe c:\windows.0\system32\spoolsv.exe
1176 8 204800 1413120
2/11/2004 3:16 PM 5.2.3790.0
(srv03_rtm.030324-2048) 55.00 KB (56,320 bytes)
3/25/2003 12:00 AM
msdtc.exe Not Available 1216 8 Not
Available Not Available 2/11/2004 3:16 PM Not
Available Not Available Not Available
svchost.exe c:\windows.0\system32\svchost.exe
1396 8 204800 1413120
2/11/2004 3:16 PM 5.2.3790.0
(srv03_rtm.030324-2048) 13.00 KB (13,312 bytes)
3/25/2003 12:00 AM
svchost.exe Not Available 1432 8
Not Available Not Available
2/11/2004 3:16 PM Not Available Not
Available Not Available
mssearch.exe c:\program files\common
files\system\mssearch\bin\mssearch.exe 1468 8
204800 1413120 2/11/2004 3:16 PM
9.107.8320.0 68.00 KB (69,632 bytes)
1/21/2003 9:30 AM
dfssvc.exe c:\windows.0\system32\dfssvc.exe
1684 8 204800 1413120
2/11/2004 3:16 PM 5.2.3790.0
(srv03_rtm.030324-2048) 130.50 KB (133,632
bytes) 3/25/2003 12:00 AM
wmiprvse.exe Not Available 1812 8
Not Available Not Available
2/11/2004 3:18 PM Not Available Not
Available Not Available
explorer.exe c:\windows.0\explorer.exe
2020 8 204800 1413120
2/11/2004 3:18 PM 6.00.3790.0
(srv03_rtm.030324-2048) 1,008.50 KB (1,032,704
bytes) 3/25/2003 12:00 AM

```

```

sqlmgr.exe c:\program files\microsoft sql
server\80\tools\bin\sqlmgr.exe 252 8
204800 1413120 2/11/2004 3:18 PM
2000.080.0760.00 72.57 KB (74,308 bytes)
12/1/2003 10:34 AM
wmiprvse.exe Not Available 308 8
Not Available Not Available
2/11/2004 3:18 PM Not Available Not
Available Not Available
helpsvc.exe
c:\windows.0\pchealth\helpctr\binaries\help
svc.exe 516 8 204800 1413120
2/11/2004 3:18 PM 5.2.3790.0
(srv03_rtm.030324-2048) 720.00 KB (737,280
bytes) 11/25/2003 12:48 PM
mmc.exe c:\windows.0\system32\mmc.exe 1532 8
204800 1413120 2/11/2004 3:19 PM
5.2.3790.0 (srv03_rtm.030324-2048)
762.50 KB (780,800 bytes) 3/25/2003
12:00 AM
vds.exe c:\windows.0\system32\vds.exe 832 8
204800 1413120 2/11/2004 3:19 PM
5.2.3790.0 (srv03_rtm.030324-2048)
157.00 KB (160,768 bytes) 3/25/2003
12:00 AM
dmadmin.exe c:\windows.0\system32\dmadmin.exe
1808 8 204800 1413120
2/11/2004 3:19 PM 5.2.3790.0
(srv03_rtm.030324-2048) 220.00 KB (225,280
bytes) 3/25/2003 12:00 AM
helpctr.exe
c:\windows.0\pchealth\helpctr\binaries\help
ctr.exe 2504 8 204800 1413120
2/11/2004 3:23 PM 5.2.3790.0
(srv03_rtm.030324-2048) 764.00 KB (782,336
bytes) 11/25/2003 12:48 PM

[Loaded Modules]

Name Version Size File Date Manufacturer
Path
winlogon 5.2.3790.0 (srv03_rtm.030324-2048)
536.50 KB (549,376 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\winlogon.exe
ntdll 5.2.3790.0 (srv03_rtm.030324-2048)
722.50 KB (739,840 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\ntdll.dll
kernel32 5.2.3790.0 (srv03_rtm.030324-2048)
965.00 KB (988,160 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\kernel32.dll
msvcrt 7.0.3790.0 (srv03_rtm.030324-2048)
319.50 KB (327,168 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\msvcrt.dll
advapi32 5.2.3790.0 (srv03_rtm.030324-2048)
559.50 KB (572,928 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\advapi32.dll
rpcrt4 5.2.3790.0 (srv03_rtm.030324-2048)
643.50 KB (658,944 bytes) 3/25/2003

```

```

12:00 AM Microsoft Corporation
c:\windows.0\system32\rpcrt4.dll
user32 5.2.3790.0 (srv03_rtm.030324-2048)
562.00 KB (575,488 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\user32.dll
gdi32 5.2.3790.0 (srv03_rtm.030324-2048)
263.00 KB (269,312 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\gdi32.dll
userenv 5.2.3790.0 (srv03_rtm.030324-2048)
732.50 KB (750,080 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\userenv.dll
nddeapi 5.2.3790.0 (srv03_rtm.030324-2048)
16.00 KB (16,384 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\nddeapi.dll
crypt32 5.131.3790.0 (srv03_rtm.030324-2048)
598.00 KB (612,352 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\crypt32.dll
msasn1 5.2.3790.0 (srv03_rtm.030324-2048)
58.00 KB (59,392 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\msasn1.dll
secur32 5.2.3790.0 (srv03_rtm.030324-2048)
63.00 KB (64,512 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\secur32.dll
winsta 5.2.3790.0 (srv03_rtm.030324-2048)
51.00 KB (52,224 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\winsta.dll
netapi32 5.2.3790.0 (srv03_rtm.030324-2048)
317.00 KB (324,608 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\netapi32.dll
profmap 5.2.3790.0 (srv03_rtm.030324-2048)
22.00 KB (22,528 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\profmap.dll
regapi 5.2.3790.0 (srv03_rtm.030324-2048)
48.50 KB (49,664 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\regapi.dll
ws2_32 5.2.3790.0 (srv03_rtm.030324-2048)
87.50 KB (89,600 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\ws2_32.dll
ws2help 5.2.3790.0 (srv03_rtm.030324-2048)
19.50 KB (19,968 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\ws2help.dll
psapi 5.2.3790.0 (srv03_rtm.030324-2048)
21.50 KB (22,016 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\psapi.dll
version 5.2.3790.0 (srv03_rtm.030324-2048)
17.00 KB (17,408 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\version.dll

```

setupapi 5.2.3790.0 (srv03_rtm.030324-2048)
1,014.50 KB (1,038,848 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\setupapi.dll
msgina 5.2.3790.0 (srv03_rtm.030324-2048)
1.14 MB (1,191,936 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\msgina.dll
shsvcs 6.00.3790.0 (srv03_rtm.030324-2048)
121.50 KB (124,416 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\shsvcs.dll
shlwapi 6.00.3790.0 (srv03_rtm.030324-2048)
281.00 KB (287,744 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\shlwapi.dll
sfc 5.2.3790.0 (srv03_rtm.030324-2048)
4.50 KB (4,608 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\sfc.dll
sfc_os 5.2.3790.0 (srv03_rtm.030324-2048)
133.00 KB (136,192 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\sfc_os.dll
wintrust 5.131.3790.0 (srv03_rtm.030324-2048)
161.50 KB (165,376 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\wintrust.dll
ole32 5.2.3790.0 (srv03_rtm.030324-2048)
1.13 MB (1,187,328 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\ole32.dll
imagehlp 5.2.3790.0 (srv03_rtm.030324-2048)
142.50 KB (145,920 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\imagehlp.dll
comctl32 6.0 (srv03_rtm.030324-2048) 907.00 KB
(928,768 bytes) 11/24/2003 11:47 AM Microsoft
Corporation
c:\windows.0\winsxs\x86_microsoft.windows.c
ommon-controls_6595b64144ccf1df_6.0.100.0_x-
ww_8417450b\comctl32.dll
winscard 5.2.3790.0 (srv03_rtm.030324-2048)
98.50 KB (100,864 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\winscard.dll
wtsapi32 5.2.3790.0 (srv03_rtm.030324-2048)
17.50 KB (17,920 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\wtsapi32.dll
winmm 5.2.3790.0 (srv03_rtm.030324-2048)
166.00 KB (169,984 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\winmm.dll
sxs 5.2.3790.0 (srv03_rtm.030324-2048)
733.00 KB (750,592 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\sxs.dll
shell32 6.00.3790.0 (srv03_rtm.030324-2048)
7.79 MB (8,166,400 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\shell32.dll

rsaenh 5.2.3790.0 (srv03_rtm.030324-2048)
176.83 KB (181,072 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\rsaenh.dll
wldap32 5.2.3790.0 (srv03_rtm.030324-2048)
158.00 KB (161,792 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\wldap32.dll
csddl 5.2.3790.0 (srv03_rtm.030324-2048)
99.00 KB (101,376 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\csddl.dll
wlnotify 5.2.3790.0 (srv03_rtm.030324-2048)
87.50 KB (89,600 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\wlnotify.dll
winspool 5.2.3790.0 (srv03_rtm.030324-2048)
140.00 KB (143,360 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\winspool.drv
mpr 5.2.3790.0 (srv03_rtm.030324-2048)
56.00 KB (57,344 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\mpr.dll
comctl32 5.82 (srv03_rtm.030324-2048) 561.00 KB
(574,464 bytes) 11/24/2003 11:47 AM Microsoft
Corporation
c:\windows.0\winsxs\x86_microsoft.windows.c
ommon-controls_6595b64144ccf1df_5.82.0.0_x-
ww_8a69ba05\comctl32.dll
uxtheme 6.00.3790.0 (srv03_rtm.030324-2048)
196.00 KB (200,704 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\uxtheme.dll
samlib 5.2.3790.0 (srv03_rtm.030324-2048)
49.00 KB (50,176 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\samlib.dll
cscui 5.2.3790.0 (srv03_rtm.030324-2048)
305.00 KB (312,320 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\cscui.dll
oleaut32 5.2.3790.0 486.00 KB (497,664
bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows.0\system32\oleaut32.dll
clbcatq 2001.12.4720.0 (srv03_rtm.030324-2048)
481.00 KB (492,544 bytes) 11/25/2003
12:45 PM Microsoft Corporation
c:\windows.0\system32\clbcatq.dll
comres 2001.12.4720.0 (srv03_rtm.030324-2048)
778.00 KB (796,672 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\comres.dll
ntmarta 5.2.3790.0 (srv03_rtm.030324-2048)
114.00 KB (116,736 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\ntmarta.dll
services 5.2.3790.0 (srv03_rtm.030324-2048)
102.00 KB (104,448 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\services.exe
scserv 5.2.3790.0 (srv03_rtm.030324-2048)
316.50 KB (324,096 bytes) 3/25/2003

12:00 AM Microsoft Corporation
c:\windows.0\system32\scserv.dll
authz 5.2.3790.0 (srv03_rtm.030324-2048)
67.00 KB (68,608 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\authz.dll
umpnpgmr 5.2.3790.0 (srv03_rtm.030324-2048)
121.50 KB (124,416 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\umpnpgmr.dll
ncobjapi 5.2.3790.0 (srv03_rtm.030324-2048)
34.50 KB (35,328 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\ncobjapi.dll
msvcp60 6.05.2144.0 388.00 KB (397,312
bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows.0\system32\msvcp60.dll
eventlog 5.2.3790.0 (srv03_rtm.030324-2048)
60.50 KB (61,952 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\eventlog.dll
cryptnet 5.131.3790.0 (srv03_rtm.030324-2048)
59.50 KB (60,928 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\cryptnet.dll
sensapi 5.2.3790.0 (srv03_rtm.030324-2048)
6.00 KB (6,144 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\sensapi.dll
cabinet 5.2.3790.0 (srv03_rtm.030324-2048)
61.00 KB (62,464 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\cabinet.dll
imm32 5.2.3790.0 (srv03_rtm.030324-2048)
105.50 KB (108,032 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\imm32.dll
apphelp 5.2.3790.0 (srv03_rtm.030324-2048)
122.00 KB (124,928 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\apphelp.dll
lsass 5.2.3790.0 (srv03_rtm.030324-2048)
13.00 KB (13,312 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\lsass.exe
lsasrv 5.2.3790.0 (srv03_rtm.030324-2048)
780.50 KB (799,232 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\lsasrv.dll
samsvr 5.2.3790.0 (srv03_rtm.030324-2048)
452.00 KB (462,848 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\samsrv.dll
cryptdll 5.2.3790.0 (srv03_rtm.030324-2048)
34.00 KB (34,816 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\cryptdll.dll
dnsapi 5.2.3790.0 (srv03_rtm.030324-2048)
147.50 KB (151,040 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\dnsapi.dll
ntdsapi 5.2.3790.0 (srv03_rtm.030324-2048)
76.00 KB (77,824 bytes) 3/25/2003

12:00 AM Microsoft Corporation
c:\windows.0\system32\ntdsapi.dll
msprivs 5.2.3790.0 (srv03_rtm.030324-2048)
46.50 KB (47,616 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\msprivs.dll
kerberos 5.2.3790.0 (srv03_rtm.030324-2048)
332.50 KB (340,480 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\kerberos.dll
msv1_0 5.2.3790.0 (srv03_rtm.030324-2048)
127.00 KB (130,048 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\msv1_0.dll
netlogon 5.2.3790.0 (srv03_rtm.030324-2048)
409.00 KB (418,816 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\netlogon.dll
w32time 5.2.3790.0 (srv03_rtm.030324-2048)
216.00 KB (221,184 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\w32time.dll
iphlpapi 5.2.3790.0 (srv03_rtm.030324-2048)
82.50 KB (84,480 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\iphlpapi.dll
schannel 5.2.3790.0 (srv03_rtm.030324-2048)
149.50 KB (153,088 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\schannel.dll
wdigest 5.2.3790.0 (srv03_rtm.030324-2048)
61.00 KB (62,464 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\wdigest.dll
rassfm 5.2.3790.0 (srv03_rtm.030324-2048)
20.50 KB (20,992 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\rassfm.dll
kdcsvc 5.2.3790.0 (srv03_rtm.030324-2048)
221.00 KB (226,304 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\kdcsvc.dll
ntdsa 5.2.3790.0 (srv03_rtm.030324-2048)
1.45 MB (1,520,640 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\ntdsa.dll
ntdsatq 5.2.3790.0 (srv03_rtm.030324-2048)
32.00 KB (32,768 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\ntdsatq.dll
mawsock 5.2.3790.0 (srv03_rtm.030324-2048)
254.00 KB (260,096 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\mawsock.dll
esent 5.2.3790.0 (srv03_rtm.030324-2048)
1.01 MB (1,056,256 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\esent.dll
scecli 5.2.3790.0 (srv03_rtm.030324-2048)
179.50 KB (183,808 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\scecli.dll

wshtcpip 5.2.3790.0 (srv03_rtm.030324-2048)
18.00 KB (18,432 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\wshtcpip.dll
ipsecsvc 5.2.3790.0 (srv03_rtm.030324-2048)
162.50 KB (166,400 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\ipsecsvc.dll
oakley 5.2.3790.0 (srv03_rtm.030324-2048)
325.50 KB (333,312 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\oakley.dll
winipsec 5.2.3790.0 (srv03_rtm.030324-2048)
34.50 KB (35,328 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\winipsec.dll
pstorsvc 5.2.3790.0 (srv03_rtm.030324-2048)
24.00 KB (24,576 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\pstorsvc.dll
psbase 5.2.3790.0 (srv03_rtm.030324-2048)
81.00 KB (82,944 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\psbase.dll
dssenh 5.2.3790.0 (srv03_rtm.030324-2048)
131.33 KB (134,480 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\dssenh.dll
wlbsctrl 5.2.3790.0 (srv03_rtm.030324-2048)
78.00 KB (79,872 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\wlbsctrl.dll
svchost 5.2.3790.0 (srv03_rtm.030324-2048)
13.00 KB (13,312 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\svchost.exe
rpcss 5.2.3790.0 (srv03_rtm.030324-2048)
276.50 KB (283,136 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\rpcss.dll
termsrv 5.2.3790.0 (srv03_rtm.030324-2048)
216.50 KB (221,696 bytes) 11/25/2003
12:46 PM Microsoft Corporation
c:\windows.0\system32\termsrv.dll
icaapi 5.2.3790.0 (srv03_rtm.030324-2048)
10.50 KB (10,752 bytes) 11/25/2003
12:46 PM Microsoft Corporation
c:\windows.0\system32\icaapi.dll
mstlsapi 5.2.3790.0 (srv03_rtm.030324-2048)
104.50 KB (107,008 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\mstlsapi.dll
activeds 5.2.3790.0 (srv03_rtm.030324-2048)
189.00 KB (193,536 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\activeds.dll
adslsdp 5.2.3790.0 (srv03_rtm.030324-2048)
142.50 KB (145,920 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\adslsdp.dll
credui 5.2.3790.0 (srv03_rtm.030324-2048)
159.00 KB (162,816 bytes) 3/25/2003

12:00 AM Microsoft Corporation
c:\windows.0\system32\credui.dll
atl 3.05.2283 83.00 KB (84,992 bytes)
3/25/2003 12:00 AM Microsoft Corporation
c:\windows.0\system32\atl.dll
rdpwsx 5.2.3790.0 (srv03_rtm.030324-2048)
80.13 KB (82,056 bytes) 11/25/2003
12:46 PM Microsoft Corporation
c:\windows.0\system32\rdpwsx.dll
wzcsvc 5.2.3790.0 (srv03_rtm.030324-2048)
272.50 KB (279,040 bytes) 3/25/2003
6:15 AM Microsoft Corporation
c:\windows.0\system32\wzcsvc.dll
rtutils 5.2.3790.0 (srv03_rtm.030324-2048)
32.00 KB (32,768 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\rtutils.dll
wmi 5.2.3790.0 (srv03_rtm.030324-2048)
6.50 KB (6,656 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\wmi.dll
dhcpcsvc 5.2.3790.0 (srv03_rtm.030324-2048)
101.50 KB (103,936 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\dhcpcsvc.dll
rastls 5.2.3790.0 (srv03_rtm.030324-2048)
155.00 KB (158,720 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\rastls.dll
cryptui 5.131.3790.0 (srv03_rtm.030324-2048)
473.50 KB (484,864 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\cryptui.dll
mprapi 5.2.3790.0 (srv03_rtm.030324-2048)
81.00 KB (82,944 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\mprapi.dll
rasapi32 5.2.3790.0 (srv03_rtm.030324-2048)
227.50 KB (232,960 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\rasapi32.dll
rasman 5.2.3790.0 (srv03_rtm.030324-2048)
56.50 KB (57,856 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\rasman.dll
tapi32 5.2.3790.0 (srv03_rtm.030324-2048)
175.00 KB (179,200 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\tapi32.dll
raschap 5.2.3790.0 (srv03_rtm.030324-2048)
106.00 KB (108,544 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\raschap.dll
schedsvc 5.2.3790.0 (srv03_rtm.030324-2048)
176.00 KB (180,224 bytes) 11/25/2003
12:48 PM Microsoft Corporation
c:\windows.0\system32\schedsvc.dll
msidle 6.00.3790.0 (srv03_rtm.030324-2048)
5.50 KB (5,632 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\msidle.dll
wkssvc 5.2.3790.0 (srv03_rtm.030324-2048)
125.00 KB (128,000 bytes) 3/25/2003

12:00 AM Microsoft Corporation
c:\windows.0\system32\wkssvc.dll
wiarpc 5.2.3790.0 (srv03_rtm.030324-2048)
30.00 KB (30,720 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\wiarpc.dll
cryptsvc 5.2.3790.0 (srv03_rtm.030324-2048)
51.00 KB (52,224 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\cryptsvc.dll
certcli 5.2.3790.0 (srv03_rtm.030324-2048)
228.00 KB (233,472 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\certcli.dll
vssapi 5.2.3790.0 (srv03_rtm.030324-2048)
528.00 KB (540,672 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\vssapi.dll
dmserver 5.2.3790.0 (srv03_rtm.030324-2048)
24.00 KB (24,576 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\dmserver.dll
es 2001.12.4720.0 (srv03_rtm.030324-2048)
221.50 KB (226,816 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\es.dll
pchsvc 5.2.3790.0 (srv03_rtm.030324-2048)
31.50 KB (32,256 bytes) 11/25/2003
12:48 PM Microsoft Corporation
c:\windows.0\pchealth\helpctr\binaries\pchs
vc.dll
srvsvc 5.2.3790.0 (srv03_rtm.030324-2048)
89.00 KB (91,136 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\srvsvc.dll
seclogon 5.2.3790.0 (srv03_rtm.030324-2048)
16.50 KB (16,896 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\seclogon.dll
trkwks 5.2.3790.0 (srv03_rtm.030324-2048)
85.00 KB (87,040 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\trkwks.dll
wmisvc 5.2.3790.0 (srv03_rtm.030324-2048)
131.00 KB (134,144 bytes) 11/25/2003
12:45 PM Microsoft Corporation
c:\windows.0\system32\wbem\wmisvc.dll
wuauerv 5.4.3790.0 (srv03_rtm.030324-2048)
10.50 KB (10,752 bytes) 11/25/2003
12:46 PM Microsoft Corporation
c:\windows.0\system32\wuauerv.dll
wuaueng 5.4.3790.0 (srv03_rtm.030324-2048)
188.50 KB (193,024 bytes) 11/25/2003
12:46 PM Microsoft Corporation
c:\windows.0\system32\wuaueng.dll
advpack 6.00.3790.0 (srv03_rtm.030324-2048)
93.50 KB (95,744 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\advpack.dll
wininet 6.00.3790.0 (srv03_rtm.030324-2048)
609.00 KB (623,616 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\wininet.dll

sens 5.2.3790.0 (srv03_rtm.030324-2048)
35.50 KB (36,352 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\sens.dll
comsvcs 2001.12.4720.0 (srv03_rtm.030324-2048)
1.14 MB (1,199,616 bytes) 11/25/2003
12:45 PM Microsoft Corporation
c:\windows.0\system32\comsvcs.dll
browser 5.2.3790.0 (srv03_rtm.030324-2048)
70.50 KB (72,192 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\browser.dll
netrap 5.2.3790.0 (srv03_rtm.030324-2048)
11.50 KB (11,776 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\netrap.dll
winhttp 5.2.3790.0 (srv03_rtm.030324-2048)
327.50 KB (335,360 bytes) 11/24/2003
11:47 AM Microsoft Corporation
c:\windows.0\winsxs\x86_microsoft.windows.w
inhttp_6595b64144ccfd5f_5.1.0.0_x-
ww_e0651936\winhttp.dll
wbemcore 5.2.3790.0 (srv03_rtm.030324-2048)
457.00 KB (467,968 bytes) 11/25/2003
12:45 PM Microsoft Corporation
c:\windows.0\system32\wbem\wbemcore.dll
esscli 5.2.3790.0 (srv03_rtm.030324-2048)
235.50 KB (241,152 bytes) 11/25/2003
12:45 PM Microsoft Corporation
c:\windows.0\system32\wbem\esscli.dll
wbemcomn 5.2.3790.0 (srv03_rtm.030324-2048)
211.50 KB (216,576 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\wbem\wbemcomn.dll
fastprox 5.2.3790.0 (srv03_rtm.030324-2048)
443.00 KB (453,632 bytes) 11/25/2003
12:45 PM Microsoft Corporation
c:\windows.0\system32\wbem\fastprox.dll
wmiutils 5.2.3790.0 (srv03_rtm.030324-2048)
90.50 KB (92,672 bytes) 11/25/2003
12:45 PM Microsoft Corporation
c:\windows.0\system32\wbem\wmiutils.dll
repdrvfs 5.2.3790.0 (srv03_rtm.030324-2048)
165.00 KB (168,960 bytes) 11/25/2003
12:45 PM Microsoft Corporation
c:\windows.0\system32\wbem\repdrvfs.dll
wmiprvsd 5.2.3790.0 (srv03_rtm.030324-2048)
405.50 KB (415,232 bytes) 11/25/2003
12:45 PM Microsoft Corporation
c:\windows.0\system32\wbem\wmiprvsd.dll
wbemess 5.2.3790.0 (srv03_rtm.030324-2048)
256.50 KB (262,656 bytes) 11/25/2003
12:45 PM Microsoft Corporation
c:\windows.0\system32\wbem\wbemess.dll
wbemsvc 5.2.3790.0 (srv03_rtm.030324-2048)
42.50 KB (43,520 bytes) 11/25/2003
12:45 PM Microsoft Corporation
c:\windows.0\system32\wbem\wbemsvc.dll
ncprov 5.2.3790.0 (srv03_rtm.030324-2048)
43.00 KB (44,032 bytes) 11/25/2003
12:45 PM Microsoft Corporation
c:\windows.0\system32\wbem\ncprov.dll

wbemcons 5.2.3790.0 (srv03_rtm.030324-2048)
69.00 KB (70,656 bytes) 11/25/2003
12:45 PM Microsoft Corporation
c:\windows.0\system32\wbem\wbemcons.dll
actxprxy 6.00.3790.0 (srv03_rtm.030324-2048)
95.00 KB (97,280 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\actxprxy.dll
netman 5.2.3790.0 (srv03_rtm.030324-2048)
209.00 KB (214,016 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\netman.dll
wzcsapi 5.2.3790.0 (srv03_rtm.030324-2048)
24.50 KB (25,088 bytes) 3/25/2003
6:15 AM Microsoft Corporation
c:\windows.0\system32\wzcsapi.dll
netshell 5.2.3790.0 (srv03_rtm.030324-2048)
1.67 MB (1,747,456 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\netshell.dll
clusapi 5.2.3790.0 (srv03_rtm.030324-2048)
56.00 KB (57,344 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\clusapi.dll
netcfgx 5.2.3790.0 (srv03_rtm.030324-2048)
726.00 KB (743,424 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\netcfgx.dll
hnetcfg 5.2.3790.0 (srv03_rtm.030324-2048)
243.50 KB (249,344 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\hnetcfg.dll
wbemprox 5.2.3790.0 (srv03_rtm.030324-2048)
17.50 KB (17,920 bytes) 11/25/2003
12:45 PM Microsoft Corporation
c:\windows.0\system32\wbem\wbemprox.dll
rasdlg 5.2.3790.0 (srv03_rtm.030324-2048)
642.00 KB (657,408 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\rasdlg.dll
rasadhlp 5.2.3790.0 (srv03_rtm.030324-2048)
6.50 KB (6,656 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\rasadhlp.dll
spoolsv 5.2.3790.0 (srv03_rtm.030324-2048)
55.00 KB (56,320 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\spoolsv.exe
spoolss 5.2.3790.0 (srv03_rtm.030324-2048)
79.00 KB (80,896 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\spoolss.dll
localspl 5.2.3790.0 (srv03_rtm.030324-2048)
304.50 KB (311,808 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\localspl.dll
cnbjmon 5.2.3680.0 (Lab03_dev\skatari).020509-1043)
45.50 KB (46,592 bytes) 3/24/2003
7:48 PM Microsoft Corporation
c:\windows.0\system32\cnbjmon.dll
pjlmon 5.2.3790.0 (srv03_rtm.030324-2048)
15.00 KB (15,360 bytes) 3/24/2003

```

7:49 PM Microsoft Corporation
c:\windows.0\system32\pjlmon.dll
tcpmon 5.2.3790.0 (srv03_rtm.030324-2048)
44.00 KB (45,056 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\tcpmon.dll
mgmtapi 5.2.3790.0 (srv03_rtm.030324-2048)
14.00 KB (14,336 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\mgmtapi.dll
snmpapi 5.2.3790.0 (srv03_rtm.030324-2048)
17.50 KB (17,920 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\snmpapi.dll
wsnmp32 5.2.3790.0 (srv03_rtm.030324-2048)
39.50 KB (40,448 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\wsnmp32.dll
usbmon 5.2.3790.0 (srv03_rtm.030324-2048)
17.00 KB (17,408 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\usbmon.dll
winnr 5.2.3790.0 (srv03_rtm.030324-2048)
15.00 KB (15,360 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\winnr.dll
wshqos 5.2.3790.0 (srv03_rtm.030324-2048)
23.00 KB (23,552 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\wshqos.dll
win32spl 5.2.3790.0 (srv03_rtm.030324-2048)
94.50 KB (96,768 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\win32spl.dll
inetpp 5.2.3790.0 (srv03_rtm.030324-2048)
71.50 KB (73,216 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\inetpp.dll
icmp 5.2.3790.0 (srv03_rtm.030324-2048)
4.50 KB (4,608 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\icmp.dll
ersvc 5.2.3790.0 (srv03_rtm.030324-2048)
22.00 KB (22,528 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\ersvc.dll
mssearch 9.107.8320.0 68.00 KB (69,632 bytes)
1/21/2003 9:30 AM Microsoft Corporation
c:\program files\common
files\system\mssearch\bin\mssearch.exe
mssws 9.107.8320.0 32.00 KB (32,768 bytes)
1/21/2003 9:30 AM Microsoft Corporation
c:\program files\common
files\system\mssearch\bin\mssws.dll
mssrch 9.107.8320.0 1.24 MB (1,302,528
bytes) 1/21/2003 9:30 AM Microsoft Corporation
c:\progra-1\common-1\system\mssearch\bin\ms
srch.dll
security 5.2.3790.0 (srv03_rtm.030324-2048)
5.50 KB (5,632 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\security.dll

```

```

tquery 9.107.8320.0 1.46 MB (1,536,000
bytes) 1/21/2003 9:30 AM Microsoft Corporation
c:\program files\common
files\system\mssearch\bin\tquery.dll
propdefs 9.107.8320.0 136.00 KB (139,264
bytes) 1/21/2003 9:30 AM Microsoft Corporation
c:\progra-1\common-1\system\mssearch\bin\pr
opdefs.dll
srchidx 9.107.8320.0 384.00 KB (393,216
bytes) 1/21/2003 9:30 AM Microsoft Corporation
c:\progra-1\common-1\system\mssearch\bin\sr
chidx.dll
iprop 5.2.3790.0 (srv03_rtm.030324-2048)
3.50 KB (3,584 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\iprop.dll
dfssvc 5.2.3790.0 (srv03_rtm.030324-2048)
130.50 KB (133,632 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\dfssvc.exe
resutils 5.2.3790.0 (srv03_rtm.030324-2048)
59.00 KB (60,416 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\resutils.dll
mfc42u 6.05.3014.0 960.00 KB (983,040
bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows.0\system32\mfc42u.dll
wsock32 5.2.3790.0 (srv03_rtm.030324-2048)
22.00 KB (22,528 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\wsock32.dll
explorer 6.00.3790.0 (srv03_rtm.030324-2048)
1,008.50 KB (1,032,704 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\explorer.exe
browseui 6.00.3790.0 (srv03_rtm.030324-2048)
1.01 MB (1,057,280 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\browseui.dll
shdocvw 6.00.3790.0 (srv03_rtm.030324-2048)
1.33 MB (1,393,664 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\shdocvw.dll
themeui 6.00.3790.0 (srv03_rtm.030324-2048)
360.50 KB (369,152 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\themeui.dll
msimg32 5.2.3790.0 (srv03_rtm.030324-2048)
4.50 KB (4,608 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\msimg32.dll
linkinfo 5.2.3790.0 (srv03_rtm.030324-2048)
16.50 KB (16,896 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\linkinfo.dll
ntshrui 6.00.3790.0 (srv03_rtm.030324-2048)
136.00 KB (139,264 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\ntshrui.dll
urlmon 6.00.3790.0 (srv03_rtm.030324-2048)
501.50 KB (513,536 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\urlmon.dll

```

```

webcheck 6.00.3790.0 (srv03_rtm.030324-2048)
261.50 KB (267,776 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\webcheck.dll
stobject 5.2.3790.0 (srv03_rtm.030324-2048)
117.50 KB (120,320 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\stobject.dll
batmeter 6.00.3790.0 (srv03_rtm.030324-2048)
28.50 KB (29,184 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\batmeter.dll
powrprof 6.00.3790.0 (srv03_rtm.030324-2048)
14.50 KB (14,848 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\powrprof.dll
printui 5.2.3790.0 (srv03_rtm.030324-2048)
536.50 KB (549,376 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\printui.dll
cfgmgr32 5.2.3790.0 (srv03_rtm.030324-2048)
17.50 KB (17,920 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\cfgmgr32.dll
drprov 5.2.3790.0 (srv03_rtm.030324-2048)
12.50 KB (12,800 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\drprov.dll
ntlanman 5.2.3790.0 (srv03_rtm.030324-2048)
41.00 KB (41,984 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\ntlanman.dll
netui0 5.2.3790.0 (srv03_rtm.030324-2048)
75.50 KB (77,312 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\netui0.dll
netui1 5.2.3790.0 (srv03_rtm.030324-2048)
184.00 KB (188,416 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\netui1.dll
davclnt 5.2.3790.0 (srv03_rtm.030324-2048)
23.50 KB (24,064 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\davclnt.dll
browseic 6.00.3790.0 (srv03_rtm.030324-2048)
62.00 KB (63,488 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\browseic.dll
shdoclc 6.00.3790.0 (srv03_rtm.030324-2048)
588.50 KB (602,624 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\shdoclc.dll
mprui 5.2.3790.0 (srv03_rtm.030324-2048)
49.00 KB (50,176 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\mprui.dll
netui2 5.2.3790.0 (srv03_rtm.030324-2048)
309.50 KB (316,928 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\netui2.dll
comdlg32 6.00.3790.0 (srv03_rtm.030324-2048)
261.00 KB (267,264 bytes) 3/25/2003

```


12:00 AM Microsoft Corporation
 c:\windows.0\system32\comdlg32.dll
 netmsg 5.2.3790.0 (srv03_rtm.030324-2048)
 178.00 KB (182,272 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\netmsg.dll
 netplwiz 5.2.3790.0 (srv03_rtm.030324-2048)
 843.00 KB (863,232 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\netplwiz.dll
 sqlmangr 2000.080.0760.00 72.57 KB (74,308 bytes)
 12/1/2003 10:34 AM Microsoft Corporation
 c:\program files\microsoft sql
 server\80\tools\bin\sqlmangr.exe
 sqlunirl 2000.080.0728.00 176.56 KB (180,800
 bytes) 3/25/2003 12:00 AM Microsoft Corporation
 c:\windows.0\system32\sqlunirl.dll
 w95scm 2000.080.0760.00 48.56 KB (49,728 bytes)
 12/1/2003 10:34 AM Microsoft Corporation
 c:\program files\microsoft sql
 server\80\tools\bin\w95scm.dll
 odbc32 3.525.1022.0 (srv03_rtm.030324-2048)
 232.00 KB (237,568 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\odbc32.dll
 sqlsvc 2000.080.0760.00 92.56 KB (94,784 bytes)
 12/1/2003 10:34 AM Microsoft Corporation
 c:\program files\microsoft sql
 server\80\tools\bin\sqlsvc.dll
 odbcbcp 2000.085.1022.0 (srv03_rtm.030324-2048)
 24.00 KB (24,576 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\odbcbcp.dll
 sqlresld 2000.080.0382.00 28.56 KB (29,248 bytes)
 12/1/2003 10:34 AM Microsoft Corporation
 c:\program files\microsoft sql
 server\80\tools\bin\sqlresld.dll
 odbccint 3.525.1022.0 (srv03_rtm.030324-2048)
 92.00 KB (94,208 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\odbccint.dll
 sqlsvc 2000.080.0194.00 24.00 KB (24,576 bytes)
 12/1/2003 10:34 AM Microsoft Corporation
 c:\program files\microsoft sql
 server\80\tools\bin\resources\1033\sqlsvc.rll
 sqlmangr 2000.080.0194.00 96.00 KB (98,304 bytes)
 12/1/2003 10:34 AM Microsoft Corporation
 c:\program files\microsoft sql
 server\80\tools\bin\resources\1033\sqlmangr.rll
 helpsvc 5.2.3790.0 (srv03_rtm.030324-2048)
 720.00 KB (737,280 bytes) 11/25/2003
 12:48 PM Microsoft Corporation
 c:\windows.0\pchealth\helpctr\binaries\help
 svc.exe
 hcappres 5.2.3790.0 (srv03_rtm.030324-2048)
 6.50 KB (6,656 bytes) 11/25/2003
 12:48 PM Microsoft Corporation
 c:\windows.0\pchealth\helpctr\binaries\hcapp
 res.dll
 itss 5.2.3790.0 (srv03_rtm.030324-2048)
 119.50 KB (122,368 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\itss.dll

mmc 5.2.3790.0 (srv03_rtm.030324-2048)
 762.50 KB (780,800 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\mmc.exe
 oleacc 4.2.5406.0 (srv03_rtm.030324-2048)
 171.00 KB (175,104 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\oleacc.dll
 mmcbase 5.2.3790.0 (srv03_rtm.030324-2048)
 70.50 KB (72,192 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\mmcbase.dll
 mmcndmgr 5.2.3790.0 (srv03_rtm.030324-2048)
 1.13 MB (1,182,720 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\mmcndmgr.dll
 msxml3 8.40.9419.0 1.28 MB (1,337,344
 bytes) 3/25/2003 12:00 AM Microsoft Corporation
 c:\windows.0\system32\msxml3.dll
 mycomput 5.2.3790.0 (srv03_rtm.030324-2048)
 96.50 KB (98,816 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\mycomput.dll
 ntmsgmgr 5.2.3790.0 (srv03_rtm.030324-2048)
 495.50 KB (507,392 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\ntmsgmgr.dll
 ntmsapi 5.2.3790.0 (srv03_rtm.030324-2048)
 42.50 KB (43,520 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\ntmsapi.dll
 dfrgsnap 5.2.3790.0 (srv03_rtm.030324-2048)
 36.00 KB (36,864 bytes) 3/25/2003
 12:00 AM Microsoft Corp. and Executive Software
 International, Inc.
 c:\windows.0\system32\dfrgsnap.dll
 dfrgres 5.2.3790.0 (srv03_rtm.030324-2048)
 50.50 KB (51,712 bytes) 3/25/2003
 12:00 AM Microsoft Corp. and Executive Software
 International, Inc. c:\windows.0\system32\dfrgres.dll
 dmdskmgr 5.2.3790.0 (srv03_rtm.030324-2048)
 164.00 KB (167,936 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\dmdskmgr.dll
 dmutil 5.2.3790.0 (srv03_rtm.030324-2048)
 51.50 KB (52,736 bytes) 3/24/2003
 7:48 PM Microsoft Corporation
 c:\windows.0\system32\dmutil.dll
 dmdskres 5.2.3790.0 (srv03_rtm.030324-2048)
 115.50 KB (118,272 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\dmdskres.dll
 els 5.2.3790.0 (srv03_rtm.030324-2048)
 178.00 KB (182,272 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\els.dll
 riched32 5.2.3790.0 (srv03_rtm.030324-2048)
 3.50 KB (3,584 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\riched32.dll

riched20 5.31.23.1218 406.00 KB (415,744
 bytes) 3/25/2003 12:00 AM Microsoft Corporation
 c:\windows.0\system32\riched20.dll
 filemgmt 5.2.3790.0 (srv03_rtm.030324-2048)
 327.50 KB (335,360 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\filemgmt.dll
 localsec 5.2.3790.0 (srv03_rtm.030324-2048)
 223.50 KB (228,864 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\localsec.dll
 adsnt 5.2.3790.0 (srv03_rtm.030324-2048)
 260.00 KB (266,240 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\adsnt.dll
 smlogcfg 5.2.3790.0 (srv03_rtm.030324-2048)
 360.50 KB (369,152 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\smlogcfg.dll
 pdh 5.2.3790.0 (srv03_rtm.030324-2048)
 274.50 KB (281,088 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\pdh.dll
 devmgr 5.2.3790.0 (srv03_rtm.030324-2048)
 276.50 KB (283,136 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\devmgr.dll
 mlang 6.00.3790.0 (srv03_rtm.030324-2048)
 570.00 KB (583,680 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\mlang.dll
 dmdlgs 5.2.3790.0 (srv03_rtm.030324-2048)
 272.50 KB (279,040 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\dmdlgs.dll
 dmview 5.2.3790.0 (srv03_rtm.030324-2048)
 61.50 KB (62,976 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\dmview.ocx
 vds_ps 5.2.3790.0 (srv03_rtm.030324-2048)
 21.50 KB (22,016 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\vds_ps.dll
 dmvdstif 5.2.3790.0 (srv03_rtm.030324-2048)
 92.50 KB (94,720 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\dmvdstif.dll
 vds 5.2.3790.0 (srv03_rtm.030324-2048)
 157.00 KB (160,768 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\vds.exe
 osuninst 5.2.3790.0 (srv03_rtm.030324-2048)
 115.50 KB (118,272 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\osuninst.dll
 vdsutil 5.2.3790.0 (srv03_rtm.030324-2048)
 38.50 KB (39,424 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\vdsutil.dll
 vdsbas 5.2.3790.0 (srv03_rtm.030324-2048)
 103.50 KB (105,984 bytes) 3/25/2003
 12:00 AM Microsoft Corporation
 c:\windows.0\system32\vdsbas.dll

```

fmifs 5.2.3790.0 (srv03_rtm.030324-2048)
17.00 KB (17,408 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\fmifs.dll
ulib 5.2.3790.0 (srv03_rtm.030324-2048)
265.50 KB (271,872 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\ulib.dll
ifsutil 5.2.3790.0 (srv03_rtm.030324-2048)
72.00 KB (73,728 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\ifsutil.dll
vdsdyndr 5.2.3790.0 (srv03_rtm.030324-2048)
307.50 KB (314,880 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\vdsdyndr.dll
dmintf 5.2.3790.0 (srv03_rtm.030324-2048)
20.00 KB (20,480 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\dmintf.dll
dmadmin 5.2.3790.0 (srv03_rtm.030324-2048)
220.00 KB (225,280 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\dmadmin.exe
dmconfig 5.2.3790.0 (srv03_rtm.030324-2048)
332.50 KB (340,480 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\dmconfig.dll
helpctr 5.2.3790.0 (srv03_rtm.030324-2048)
764.00 KB (782,336 bytes) 11/25/2003
12:48 PM Microsoft Corporation
c:\windows.0\pchealth\helpctr\binaries\help
ctr.exe
pchshell 5.2.3790.0 (srv03_rtm.030324-2048)
100.50 KB (102,912 bytes) 11/25/2003
12:48 PM Microsoft Corporation
c:\windows.0\pchealth\helpctr\binaries\pchs
hell.dll
mshtml 6.00.3790.0 (srv03_rtm.030324-2048)
2.78 MB (2,916,352 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\mshtml.dll
msimtf 5.2.3790.0 (srv03_rtm.030324-2048)
149.00 KB (152,576 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\msimtf.dll
msctf 5.2.3790.0 (srv03_rtm.030324-2048)
287.00 KB (293,888 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\msctf.dll
jscript 5.6.0.8515 436.00 KB (446,464
bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows.0\system32\jscript.dll
msls31 3.10.349.0 147.00 KB (150,528
bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows.0\system32\msls31.dll
mshtml 6.00.3790.0 (srv03_rtm.030324-2048)
443.50 KB (454,144 bytes) 3/25/2003
12:00 AM Microsoft Corporation
c:\windows.0\system32\mshtml.dll
vbscript 5.6.0.8515 404.00 KB (413,696
bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows.0\system32\vbscript.dll

```

```

mfc42 6.05.3014.0 960.00 KB (983,040
bytes) 3/25/2003 12:00 AM Microsoft Corporation
c:\windows.0\system32\mfc42.dll
msinfo 5.2.3790.0 (srv03_rtm.030324-2048)
358.50 KB (367,104 bytes) 11/25/2003
12:48 PM Microsoft Corporation
c:\windows.0\pchealth\helpctr\binaries\msin
fo.dll

[Services]

Display Name Name State Start Mode
Service Type Path Error Control
Start Name Tag ID
Alerter Alerter Stopped Disabled Share Process
c:\windows.0\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
Application Layer Gateway Service ALG
Stopped Manual Own Process
c:\windows.0\system32\alg.exe Normal NT
AUTHORITY\LocalService 0
Application Management AppMgmt Stopped
Manual Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Windows Audio AudioSrv Stopped Disabled
Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Background Intelligent Transfer Service BITS
Stopped Manual Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Computer Browser Browser Running Auto
Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Indexing Service CiSvc Stopped Disabled
Share Process
c:\windows.0\system32\cisvc.exe
Normal LocalSystem 0
ClipBook ClipSrv Stopped Disabled Own Process
c:\windows.0\system32\clipsrv.exe
Normal LocalSystem 0
COM+ System Application COMSysApp Stopped
Manual Own Process
c:\windows.0\system32\dlhhost.exe
/processid:{02d4b3f1-fd88-11d1-960d-00805fc79235}
Normal LocalSystem 0
Cryptographic Services CryptSvc Running
Auto Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Distributed File System Dfs Running
Auto Own Process
c:\windows.0\system32\dfssvc.exe
Normal LocalSystem 0
DHCP Client Dhcp Running Auto
Share Process
c:\windows.0\system32\svchost.exe -k
networkservice Normal NT
AUTHORITY\NetworkService 0

```

```

Logical Disk Manager Administrative Service
dmdadmin Running Manual Share Process
c:\windows.0\system32\dmdadmin.exe /com
Normal LocalSystem 0
Logical Disk Manager dmserver Running
Auto Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
DNS Client Dnscache Running Auto
Share Process
c:\windows.0\system32\svchost.exe -k
networkservice Normal NT
AUTHORITY\NetworkService 0
Error Reporting Service ERSvc Running
Auto Share Process
c:\windows.0\system32\svchost.exe -k winerr
Ignore LocalSystem 0
Event Log Eventlog Running Auto Share Process
c:\windows.0\system32\services.exe
Normal LocalSystem 0
COM+ Event System EventSystem Running
Manual Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Help and Support helpsvc Running Auto
Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Human Interface Device Access HidServ Stopped
Disabled Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
HTTP SSL HTTPFilter Stopped Manual
Share Process
c:\windows.0\system32\lsass.exe
Normal LocalSystem 0
IMAPI CD-Burning COM Service ImapService
Stopped Disabled Own Process
c:\windows.0\system32\imapi.exe
Normal LocalSystem 0
Intersite Messaging IsmServ Stopped Disabled Own
Process c:\windows.0\system32\ismserv.exe
Normal LocalSystem 0
Kerberos Key Distribution Center kdc
Stopped Disabled Share Process
c:\windows.0\system32\lsass.exe
Normal LocalSystem 0
Server lanmanserver Running Auto
Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Workstation lanmanworkstation Running
Auto Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
License Logging LicenseService Stopped
Disabled Own Process
c:\windows.0\system32\llssrv.exe
Normal NT AUTHORITY\NetworkService 0
TCP/IP NetBIOS Helper LmHosts Running
Auto Share Process
c:\windows.0\system32\svchost.exe -k

```

```

localservice Normal NT
AUTHORITY\LocalService 0
Messenger Messenger Stopped Disabled Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
NetMeeting Remote Desktop Sharing mmsrvc
Stopped Disabled Own Process
c:\windows.0\system32\mmsrvc.exe
Normal LocalSystem 0
Distributed Transaction Coordinator MSDTC
Running Auto Own Process
c:\windows.0\system32\msdtc.exe
Normal NT AUTHORITY\NetworkService 0

Windows Installer MSIServer Stopped Manual
Share Process
c:\windows.0\system32\msiexec.exe /v
Normal LocalSystem 0
Microsoft Search MSSEARCH Running Auto
Share Process "c:\program
files\common files\system\mssearch\bin\mssearch.exe"
Normal LocalSystem 0
MSSQLSERVER MSSQLSERVER Stopped
Manual Own Process
c:\sql2k\mssql\bin\sqlservr.exe
Normal LocalSystem 0
MSSQLServerADHelper MSSQLServerADHelper Stopped
Manual Own Process c:\program
files\microsoft sql server\80\tools\bin\sqladhlp.exe
Normal LocalSystem 0
Network DDE NetDDE Stopped Disabled
Share Process
c:\windows.0\system32\netdde.exe
Normal LocalSystem 0
Network DDE DSDM NetDDEdadm Stopped
Disabled Share Process
c:\windows.0\system32\netdde.exe
Normal LocalSystem 0
Net Logon Netlogon Stopped Manual Share Process
c:\windows.0\system32\lsass.exe
Normal LocalSystem 0
Network Connections Netman Running Manual
Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Network Location Awareness (NLA) Nla
Running Manual Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
File Replication NtFrs Stopped Manual Own
Process c:\windows.0\system32\ntfrs.exe
Ignore LocalSystem 0
NT LM Security Support Provider NtLmSsp
Running Manual Share Process
c:\windows.0\system32\lsass.exe
Normal LocalSystem 0
Removable Storage NtmsSvc Stopped Manual
Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Plug and Play PlugPlay Running Auto
Share Process

```

```

c:\windows.0\system32\services.exe
Normal LocalSystem 0
IPSEC Services PolicyAgent Running
Auto Share Process
c:\windows.0\system32\lsass.exe
Normal LocalSystem 0
Protected Storage ProtectedStorage Running
Auto Share Process
c:\windows.0\system32\lsass.exe
Normal LocalSystem 0
Remote Access Auto Connection Manager RasAuto
Stopped Manual Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Remote Access Connection Manager RasMan
Stopped Manual Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Remote Desktop Help Session Manager RDSessMgr
Stopped Manual Own Process
c:\windows.0\system32\sessmgr.exe
Normal LocalSystem 0
Routing and Remote Access RemoteAccess
Stopped Disabled Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Remote Registry RemoteRegistry Running
Auto Share Process
c:\windows.0\system32\svchost.exe -k regsvr
Normal NT AUTHORITY\LocalService 0

Remote Procedure Call (RPC) Locator RpcLocator
Stopped Manual Own Process
c:\windows.0\system32\locator.exe
Normal NT AUTHORITY\NetworkService 0

Remote Procedure Call (RPC) RpcSs Running
Auto Share Process
c:\windows.0\system32\svchost -k rpcss
Normal LocalSystem 0
Resultant Set of Policy Provider RSOPProv
Stopped Manual Share Process
c:\windows.0\system32\rsopprov.exe
Normal LocalSystem 0
Special Administration Console Helper sacsvr
Stopped Manual Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Security Accounts Manager SamSs Running
Auto Share Process
c:\windows.0\system32\lsass.exe
Normal LocalSystem 0
Smart Card SCardSvr Stopped Manual
Share Process
c:\windows.0\system32\scardsvr.exe
Ignore NT AUTHORITY\LocalService 0

Task Scheduler Schedule Running Auto
Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Secondary Logon seclogon Running Auto
Share Process

```

```

c:\windows.0\system32\svchost.exe -k
netshvc Ignore LocalSystem 0
System Event Notification SENS Running
Auto Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Internet Connection Firewall (ICF) / Internet
Connection Sharing (ICS) SharedAccess
Stopped Disabled Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Shell Hardware Detection ShellHWDetection
Running Auto Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Ignore LocalSystem 0
Print Spooler Spooler Running Auto Own
Process c:\windows.0\system32\spoolsv.exe
Normal LocalSystem 0
SQLSERVERAGENT SQLSERVERAGENT Stopped
Manual Own Process
c:\sql2k\mssql\bin\sqlagent.exe
Normal LocalSystem 0
Windows Image Acquisition (WIA) stisvc
Stopped Disabled Share Process
c:\windows.0\system32\svchost.exe -k imgsvc
Normal NT AUTHORITY\LocalService 0

Microsoft Software Shadow Copy Provider swprv
Stopped Manual Own Process
c:\windows.0\system32\svchost.exe -k swprv
Normal LocalSystem 0
Performance Logs and Alerts SysmonLog Stopped
Manual Own Process
c:\windows.0\system32\smlogsvc.exe
Normal NT Authority\NetworkService 0

Telephony Tapisrv Stopped Manual Share Process
c:\windows.0\system32\svchost.exe -k
tapisrv Normal LocalSystem 0
Terminal Services TermService Running
Manual Share Process
c:\windows.0\system32\svchost.exe -k
termshvc Normal LocalSystem 0
Themes Themes Stopped Disabled Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Telnet TlntSvr Stopped Disabled Own Process
c:\windows.0\system32\tlntsvr.exe
Normal NT AUTHORITY\LocalService 0

Distributed Link Tracking Server TrkWks
Stopped Disabled Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Distributed Link Tracking Client TrkWks
Running Auto Share Process
c:\windows.0\system32\svchost.exe -k
netshvc Normal LocalSystem 0
Terminal Services Session Directory Tssdis
Stopped Disabled Own Process
c:\windows.0\system32\tssdis.exe
Normal LocalSystem 0

```

```

Upload Manager      uploadmgr Stopped Manual
Share Process
c:\windows.0\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Uninterruptible Power Supply UPS Stopped
Manual Own Process
c:\windows.0\system32\ups.exe Normal NT
AUTHORITY\LocalService 0
Virtual Disk Service vds Running
Manual Own Process
c:\windows.0\system32\vds.exe Normal
LocalSystem 0
Volume Shadow Copy VSS Stopped Manual Own
Process c:\windows.0\system32\vssvc.exe
Normal LocalSystem 0
Windows Time W32Time Stopped Manual
Share Process
c:\windows.0\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
WebClient WebClient Stopped Disabled Share Process
c:\windows.0\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
WinHTTP Web Proxy Auto-Discovery Service
WinHttpAutoProxySvc Stopped Manual
Share Process
c:\windows.0\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
Windows Management Instrumentation winmgmt
Running Auto Share Process
c:\windows.0\system32\svchost.exe -k
netsvcs Ignore LocalSystem 0
Portable Media Serial Number Service WmdmPmSN
Stopped Manual Share Process
c:\windows.0\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Windows Management Instrumentation Driver Extensions
Wmi Stopped Manual Share Process
c:\windows.0\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
WMI Performance Adapter WmiApSrv Stopped
Manual Own Process
c:\windows.0\system32\wbem\wmiapsrv.exe
Normal LocalSystem 0
Automatic Updates wuauclt Running Auto
Share Process
c:\windows.0\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Wireless Configuration WZCSVC Running
Auto Share Process
c:\windows.0\system32\svchost.exe -k
netsvcs Normal LocalSystem 0

[Program Groups]

Group Name Name User Name
Accessories Default User:Accessories
Default User
Accessories\Accessibility Default
User:Accessories\Accessibility Default User

```

```

Accessories\Entertainment Default
User:Accessories\Entertainment Default User

Startup Default User:Startup Default User

Accessories All Users:Accessories All
Users
Accessories\Accessibility All
Users:Accessories\Accessibility All Users
Accessories\Communications All
Users:Accessories\Communications All Users
Accessories\Entertainment All
Users:Accessories\Entertainment All Users
Accessories\System Tools All
Users:Accessories\System Tools All Users
Administrative Tools All
Users:Administrative Tools All Users
Microsoft SQL Server All Users:Microsoft SQL
Server All Users
Startup All Users:Startup All Users
Accessories NT AUTHORITY\SYSTEM:Accessories
NT AUTHORITY\SYSTEM
Accessories\Accessibility NT
AUTHORITY\SYSTEM:Accessories\Accessibility NT
AUTHORITY\SYSTEM
Accessories\Entertainment NT
AUTHORITY\SYSTEM:Accessories\Entertainment NT
AUTHORITY\SYSTEM
Startup NT AUTHORITY\SYSTEM:Startup NT
AUTHORITY\SYSTEM
Accessories VY\Administrator:Accessories
VY\Administrator
Accessories\Accessibility
VY\Administrator:Accessories\Accessibility
VY\Administrator
Accessories\Entertainment
VY\Administrator:Accessories\Entertainment
VY\Administrator
Administrative Tools
VY\Administrator:Administrative Tools
VY\Administrator
Startup VY\Administrator:Startup
VY\Administrator

[Startup Programs]

Program Command User Name Location
desktop desktop.ini NT AUTHORITY\SYSTEM
Startup
desktop desktop.ini VY\Administrator
Startup
desktop desktop.ini .DEFAULT Startup
desktop desktop.ini All Users Common
Startup
Service Manager
c:\progra-1\micros-1\80\tools\bin\sqlmgr
.exe /n All Users Common Startup

[OLE Registration]

Object Local Server
Sound (OLE2) sndrec32.exe
Media Clip mplay32.exe

```

```

Video Clip mplay32.exe /avi
MIDI Sequence mplay32.exe /mid
Sound Not Available
Media Clip Not Available
WordPad Document "%programfiles%\windows
nt\accessories\wordpad.exe"
Windows Media Services DRM Storage object Not
Available
Bitmap Image mspaint.exe

[Windows Error Reporting]

Time Type Details

[Internet Settings]

[Internet Explorer]

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

Item Value
Version 6.0.3790.0
Build 63790
Application Path C:\Program Files\Internet
Explorer
Language English (United States)
Active Printer Not Available

Cipher Strength 128-bit
Content Advisor Disabled
IEAK Install No

[File Versions]

File Version Size Date Path
Company
actxprxy.dll 6.0.3790.0 95 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft
advpack.dll 6.0.3790.0 94 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft
asctrls.ocx 6.0.3790.0 90 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft
browselc.dll 6.0.3790.0 62 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft
browseui.dll 6.0.3790.0 1,033 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft
cdfview.dll 6.0.3790.0 144 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft
comctl32.dll 5.82.3790.0 561 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft
Corporation

```

```

dxtrans.dll      6.3.3790.0      198 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
dxtmsft.dll      6.3.3790.0      344 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
iecont.dll       <File Missing>  Not Available
Not Available   Not Available   Not
Available
iecontlc.dll     <File Missing>  Not Available
Not Available   Not Available   Not
Available
iedkcs32.dll     16.0.3790.0     300 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
iepeers.dll      6.0.3790.0      230 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
iesetup.dll      6.0.3790.0      59 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
ieuinit.inf      Not Available   20 KB
3/25/2003 C:\WINDOWS.0\system32      Not
Available
iexplore.exe     6.0.3790.0      90 KB
3/25/2003 C:\Program Files\Internet
Explorer Microsoft Corporation
imgutil.dll      5.2.3790.0      35 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
inetcpl.cpl      6.0.3790.0      303 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
inetcplc.dll     6.0.3790.0      109 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
inseng.dll       6.0.3790.0      72 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
mlang.dll        6.0.3790.0      570 KB 3/25/2003
C:\WINDOWS.0\system32      Microsoft
Corporation
msencode.dll     2002.10.4.0     112 KB
3/25/2003 C:\WINDOWS.0\system32      Not
Available
mshta.exe        6.0.3790.0      26 KB 3/25/2003
C:\WINDOWS.0\system32      Microsoft
Corporation
mshtml.dll      6.0.3790.0      2,848 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
mshtml.tlb      6.0.3790.0      1,319 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
mshtmlmled.dll  6.0.3790.0      444 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
mshtmlmer.dll   6.0.3790.0      55 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
msident.dll     6.0.3790.0      47 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation

```

```

msidentld.dll   6.0.3790.0      15 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
msieftpl.dll    6.0.3790.0      230 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
msrating.dll    6.0.3790.0      132 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
mstime.dll      6.0.3790.0      491 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
occache.dll     6.0.3790.0      89 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
proctexe.ocx    6.3.3790.0      78 KB
3/25/2003 C:\WINDOWS.0\system32
Intel Corporation
sendmail.dll    6.0.3790.0      52 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
shdoclc.dll     6.0.3790.0      589 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
shdocvw.dll    6.0.3790.0      1,361 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
shfolder.dll    6.0.3790.0      23 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
shlwapi.dll     6.0.3790.0      281 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
tdc.ocx         1.3.0.3130      58 KB 3/25/2003
C:\WINDOWS.0\system32      Microsoft
Corporation
url.dll         6.0.3790.0      36 KB 3/25/2003
C:\WINDOWS.0\system32      Microsoft
Corporation
urlmon.dll      6.0.3790.0      502 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
webcheck.dll    6.0.3790.0      262 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation
wininet.dll     6.0.3790.0      609 KB
3/25/2003 C:\WINDOWS.0\system32
Microsoft Corporation

[Connectivity]

Item Value
Connection Preference      Never dial

LAN Settings

AutoConfigProxy      Not Available
AutoProxyDetectMode  Disabled
AutoConfigURL
Proxy Disabled
ProxyServer
ProxyOverride

```

```

[Cache]

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

Item Value
Page Refresh Type      Automatic
Temporary Internet Files Folder      C:\Documents
and Settings\NetworkService.NT AUTHORITY\Local
Settings\Temporary Internet Files
Total Disk Space      Not Available
Available Disk Space      Not Available
Maximum Cache Size      Not Available
Available Cache Size      Not Available

[List of Objects]

Program File      Status      CodeBase
No cached object information available

[Content]

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

Item Value
Content Advisor      Disabled

[Personal Certificates]

Issued To      Issued By      Validity      Signature Algorithm
No personal certificate information available

[Other People Certificates]

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

[Publishers]

Name
No publisher information available

[Security]

Zone      Security Level
My Computer      Custom
Local intranet      Medium-low
Trusted sites      Medium
Internet High
Restricted sites      High

```

Client Summary

System Information report written at: 02/11/2004

04:10:16 PM

[System Information]

[Following are sub-categories of this main category]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Server
Version	5.0.2195 Service Pack 2 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	CL8
System Manufacturer	HP
System Model	ProLiant DL360 G3
System Type	X86-based PC
Processor x86 Family 15 Model 2 Stepping 5	
GenuineIntel ~38092 Mhz	
Processor x86 Family 15 Model 2 Stepping 5	
GenuineIntel ~38092 Mhz	
Processor x86 Family 15 Model 2 Stepping 5	
GenuineIntel ~38092 Mhz	
BIOS Version	07/04/03
Windows Directory	C:\WINNT
System Directory	C:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition1
Locale	United States
User Name	CL8\Administrator
Time Zone	Central Standard Time
Total Physical Memory	1,048,084 KB
Available Physical Memory	887,172 KB
Total Virtual Memory	2,783,616 KB
Available Virtual Memory	2,557,632 KB
Page File Space	1,735,532 KB
Page File	C:\pagefile.sys

[Hardware Resources]

[Following are sub-categories of this main category]

[Conflicts/Sharing]

Resource	Device
IRQ 7	Base System Device
IRQ 7	Standard OpenHCD USB Host Controller

[DMA]

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

[Forced Hardware]

Device	PNP Device ID
No Forced Hardware	

[I/O]

Address Range	Device	Status
0x0000-0x0CFF	PCI bus	OK
0x0000-0x0CFF	PCI bus	OK
0x0000-0x0CFF	Direct memory access controller	OK
0x03B0-0x03BB	PCI bus	OK
0x03B0-0x03BB	ATI Technologies Inc. RAGE XL PCI	OK
0x03C0-0x03DF	PCI bus	OK
0x03C0-0x03DF	ATI Technologies Inc. RAGE XL PCI	OK
0x2400-0x24FF	ATI Technologies Inc. RAGE XL PCI	OK
0x2800-0x28FF	Compaq Smart Array 5i	OK
0x1800-0x18FF	Base System Device	OK
0x2C00-0x2CFF	Base System Device	OK
0x0A79-0x0A79	ISAPNP Read Data Port	OK
0x0279-0x0279	ISAPNP Read Data Port	OK
0x02F4-0x02F7	ISAPNP Read Data Port	OK
0x0F50-0x0F58	Motherboard resources	OK
0x0408-0x040F	Motherboard resources	OK
0x0092-0x0092	Motherboard resources	OK
0x0900-0x0903	Motherboard resources	OK
0x0910-0x0911	Motherboard resources	OK
0x0920-0x0923	Motherboard resources	OK
0x0930-0x0937	Motherboard resources	OK
0x0940-0x0947	Motherboard resources	OK
0x0950-0x0957	Motherboard resources	OK
0x0C06-0x0C08	Motherboard resources	OK
0x0C14-0x0C14	Motherboard resources	OK
0x0C49-0x0C4A	Motherboard resources	OK
0x0C50-0x0C52	Motherboard resources	OK
0x0C6C-0x0C6F	Motherboard resources	OK
0x0010-0x001F	Motherboard resources	OK
0x0230-0x0233	Motherboard resources	OK
0x0260-0x0267	Motherboard resources	OK
0x04D0-0x04D1	Motherboard resources	OK
0x0700-0x070F	Motherboard resources	OK
0x0800-0x081F	Motherboard resources	OK
0x0C80-0x0C83	Motherboard resources	OK
0x0CD4-0x0CD7	Motherboard resources	OK
0x0CF9-0x0CF9	Motherboard resources	OK
0x0020-0x0021	Programmable interrupt controller	OK
0x00A0-0x00A1	Programmable interrupt controller	OK
0x0C00-0x0C01	Programmable interrupt controller	OK
0x0040-0x0043	System timer	OK
0x0080-0x008F	Direct memory access controller	OK
0x00C0-0x00DF	Direct memory access controller	OK
0x040B-0x040B	Direct memory access controller	OK
0x04D6-0x04D6	Direct memory access controller	OK
0x0061-0x0061	System speaker	OK
0x0060-0x0060	Standard 101/102-Key or Microsoft	OK
Natural PS/2 Keyboard		OK
0x0064-0x0064	Standard 101/102-Key or Microsoft	OK
Natural PS/2 Keyboard		OK
0x002E-0x002F	Extended IO Bus	OK

0x0220-0x0223	Extended IO Bus	OK
0x0240-0x025F	Extended IO Bus	OK
0x0070-0x0073	Extended IO Bus	OK
0x03F8-0x03FF	Communications Port (COM1)	OK
0x03F2-0x03F5	Standard floppy disk controller	OK
0x03F7-0x03F7	Standard floppy disk controller	OK
0x2000-0x200F	Standard Dual Channel PCI IDE Controller	OK
0x01F0-0x01F7	Primary IDE Channel	OK
0x03F6-0x03F6	Primary IDE Channel	OK
0x0170-0x0177	Secondary IDE Channel	OK
0x0376-0x0376	Secondary IDE Channel	OK

[IRQs]

IRQ Number	Device
9	Microsoft ACPI-Compliant System
31	Compaq Smart Array 5i
5	Base System Device
7	Base System Device
7	Standard OpenHCD USB Host Controller
1	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
12	PS/2 Compatible Mouse
4	Communications Port (COM1)
6	Standard floppy disk controller
14	Primary IDE Channel
30	Compaq NC7781 Gigabit Server Adapter #2
29	Compaq NC7781 Gigabit Server Adapter

[Memory]

Range	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	ATI Technologies Inc. RAGE XL PCI	OK
0xF5D00000-0xF6FFFFFF	PCI bus	OK
0xF6000000-0xF6FFFFFF	ATI Technologies Inc. RAGE XL PCI	OK
0xF5FF0000-0xF5FF0FFF	ATI Technologies Inc. RAGE XL PCI	OK
0xF5F80000-0xF5FBFFFF	Compaq Smart Array 5i	OK
0xF5DF0000-0xF5DF3FFF	Compaq Smart Array 5i	OK
0xF5F70000-0xF5F701FF	Base System Device	OK
0xF5F60000-0xF5F607FF	Base System Device	OK
0xF5F50000-0xF5F51FFF	Base System Device	OK
0xF5E80000-0xF5E8FFFF	Base System Device	OK
0xF5E70000-0xF5E70FFF	Standard OpenHCD USB Host Controller	OK
0xF7E00000-0xF7EFFFFFF	PCI bus	OK
0xF7EF0000-0xF7EFFFFFF	Compaq NC7781 Gigabit Server Adapter #2	OK
0xF7F00000-0xF7FFFFFF	PCI bus	OK
0xF7FF0000-0xF7FFFFFF	Compaq NC7781 Gigabit Server Adapter	OK

[Components]

[Following are sub-categories of this main category]

[Multimedia]

[Following are sub-categories of this main category]

[Audio Codecs]

Codec	Manufacturer	Description	Status	File	Version	Size	Creation Date
c:\winnt\system32\iac25_32.ax	Intel Corporation	Indeo® audio software	OK	C:\WINNT\System32\IAC25_32.AX	2.05.53	195.00 KB (199,680 bytes)	12/7/1999
7:00:00 AM							
c:\winnt\system32\lhacm.acm	Microsoft Corporation		OK	C:\WINNT\System32\LHACM.ACM	4.4.3385	32.27 KB (34,064 bytes)	9/13/2002
5:46:04 PM							
c:\winnt\system32\msgsm32.acm	Microsoft Corporation		OK	C:\WINNT\System32\MSGSM32.ACM	5.00.2134.1	22.27 KB (22,800 bytes)	12/7/1999
7:00:00 AM							
c:\winnt\system32\msg711.acm	Microsoft Corporation		OK	C:\WINNT\System32\MSG711.ACM	5.00.2134.1	10.27 KB (10,512 bytes)	12/7/1999
7:00:00 AM							
c:\winnt\system32\msg723.acm	Microsoft Corporation		OK	C:\WINNT\System32\MSG723.ACM	4.4.3385	106.77 KB (109,328 bytes)	9/13/2002
5:46:03 PM							
c:\winnt\system32\imaadp32.acm	Microsoft Corporation		OK	C:\WINNT\System32\IMAADP32.ACM	5.00.2134.1	16.27 KB (16,656 bytes)	12/7/1999
7:00:00 AM							
c:\winnt\system32\tsssoft32.acm	DSP GROUP, INC.		OK	C:\WINNT\System32\TSSOFT32.ACM	1.01	9.27 KB (9,488 bytes)	12/7/1999
7:00:00 AM							
c:\winnt\system32\msadp32.acm	Microsoft Corporation		OK	C:\WINNT\System32\MSADP32.ACM	5.00.2134.1	14.77 KB (15,120 bytes)	12/7/1999
7:00:00 AM							

[Video Codecs]

Codec	Manufacturer	Description	Status	File	Version	Size	Creation Date
c:\winnt\system32\ir50_32.dll	Intel Corporation	Indeo® video 5.10	OK	C:\WINNT\System32\IR50_32.DLL			

R.5.10.15.2.55	737.50 KB (755,200 bytes)	12/7/1999 7:00:00 AM
c:\winnt\system32\msh261.drv	Microsoft Corporation	OK
C:\WINNT\System32\MSH261.DRV	4.4.3385	163.77 KB (167,696 bytes) 9/13/2002
5:46:04 PM		
c:\winnt\system32\msh263.drv	Microsoft Corporation	OK
C:\WINNT\System32\MSH263.DRV	4.4.3385	252.27 KB (258,320 bytes) 9/13/2002
5:45:39 PM		
c:\winnt\system32\ir32_32.dll	Intel(R) Corporation	OK
C:\WINNT\System32\IR32_32.DLL	Not Available	194.50 KB (199,168 bytes) 12/7/1999
7:00:00 AM		
c:\winnt\system32\iccvld.dll	Radius Inc.	OK
C:\WINNT\System32\ICCVLD.DLL	1.10.0.6	108.00 KB (110,592 bytes) 12/7/1999 7:00:00 AM
c:\winnt\system32\msvidc32.dll	Microsoft Corporation	OK
C:\WINNT\System32\MSVIDC32.DLL	5.00.2134.1	27.27 KB (27,920 bytes) 12/7/1999 7:00:00 AM
c:\winnt\system32\mrle32.dll	Microsoft Corporation	OK
C:\WINNT\System32\MSRLE32.DLL	5.00.2134.1	10.77 KB (11,024 bytes) 12/7/1999
7:00:00 AM		

[CD-ROM]

Item	Value
Drive D:	
Description	CD-ROM Drive
Media Loaded	False
Media Type	CD-ROM
Name	COMPAQ CRN-8245B
Manufacturer	(Standard CD-ROM drives)
Status	OK
Transfer Rate	Not Available
SCSI Target ID	0
PNP Device ID	IDE\CDROMCOMPAQ_CRN-8245B\2.19\5\FB0C83D&0&0.0
.0	

[Sound Device]

Item	Value
No sound devices	

[Display]

Item	Value
Name	ATI Technologies Inc. RAGE XL PCI
PNP Device ID	PCI\VEN_1002&DEV_4752&SUBSYS_001E0E11&REV_27\3&267A616A&0&18
Adapter Type	ATI RAGE XL PCI, ATI Technologies Inc. compatible

Adapter Description	ATI Technologies Inc. RAGE XL PCI
Adapter RAM	8.00 MB (8,388,608 bytes)
Installed Drivers	atidrab.dll
Driver Version	5.00.2179.1
INF File	display.inf (atirage3 section)
Color Planes	1
Color Table Entries	65536
Resolution	640 x 480 x 60 hertz
Bits/Pixel	16

[Infrared]

Item	Value
No infrared devices	

[Input]

[Following are sub-categories of this main category]

[Keyboard]

Item	Value
Description	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
Name	Enhanced (101- or 102-key)
Layout	00000409
PNP Device ID	ACPI\PNP0303\4&35118DFF&0
NumberOfFunctionKeys	12

[Pointing Device]

Item	Value
Hardware Type	PS/2 Compatible Mouse
Number of Buttons	2
Status	OK
PNP Device ID	ACPI\PNP0F13\4&35118DFF&0
Power Management Supported	False
Double Click Threshold	6
Handedness	Right Handed Operation

[Modem]

Item	Value
No modems	

[Network]

[Following are sub-categories of this main category]

[Adapter]

Item	Value
Name	[00000000] RAS Async Adapter
Adapter Type	Not Available
Product Name	RAS Async Adapter
Installed	True
PNP Device ID	Not Available
Last Reset	2/11/2004 9:52:34 AM

```

Index      0
Service Name      AsyncMac
IP Address      Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled     False
DHCP Server      Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address      Not Available
Service Name      Not Available

```

```

Name      [00000001] WAN Miniport (L2TP)
Adapter Type      Not Available
Product Name      WAN Miniport (L2TP)
Installed True
PNP Device ID      ROOT\MS_L2TPMINIPOINT\0000
Last Reset      2/11/2004 9:52:34 AM
Index      1
Service Name      Rasl2tp
IP Address      Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled     False
DHCP Server      Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address      Not Available
Service Name      Rasl2tp
Driver      c:\winnt\system32\drivers\rasl2tp.sys
(50800, 5.00.2179.1)

```

```

Name      [00000002] WAN Miniport (PPTP)
Adapter Type      Wide Area Network (WAN)
Product Name      WAN Miniport (PPTP)
Installed True
PNP Device ID      ROOT\MS_PPTPMINIPOINT\0000
Last Reset      2/11/2004 9:52:34 AM
Index      2
Service Name      PptpMiniport
IP Address      Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled     False
DHCP Server      Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address      50:50:54:50:30:30
Service Name      PptpMiniport
Driver      c:\winnt\system32\drivers\raspptp.sys
(47856, 5.00.2160.1)

```

```

Name      [00000003] Direct Parallel
Adapter Type      Not Available
Product Name      Direct Parallel
Installed True
PNP Device ID      ROOT\MS_PTIMINIPOINT\0000
Last Reset      2/11/2004 9:52:34 AM
Index      3
Service Name      Raspti
IP Address      Not Available
IP Subnet Not Available
Default IP Gateway Not Available

```

```

DHCP Enabled     False
DHCP Server      Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address      Not Available
Service Name      Raspti
Driver      c:\winnt\system32\drivers\raspti.sys
(16880, 5.00.2146.1)

```

```

Name      [00000004] WAN Miniport (IP)
Adapter Type      Not Available
Product Name      WAN Miniport (IP)
Installed True
PNP Device ID      ROOT\MS_NDISWANIP\0000
Last Reset      2/11/2004 9:52:34 AM
Index      4
Service Name      Ndiswan
IP Address      Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled     False
DHCP Server      Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address      Not Available
Service Name      Ndiswan
Driver      c:\winnt\system32\drivers\ndiswan.sys
(90096, 5.00.2195.2779)

```

```

Name      [00000008] Compaq NC7781 Gigabit Server
Adapter Type      Ethernet 802.3
Product Name      Compaq NC7781 Gigabit Server
Adapter Type      Ethernet 802.3
Installed True
PNP Device ID      PCI\VEN_14E4&DEV_16A7&SUBSYS_00CB0E11&REV_0
2\3&1070020&0&10
Last Reset      2/11/2004 9:52:34 AM
Index      8
Service Name      q57w2k
IP Address      130.172.11.8
IP Subnet 255.255.0.0
Default IP Gateway Not Available
DHCP Enabled     False
DHCP Server      Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address      00:0B:CD:AF:7F:F2
Service Name      q57w2k
IRQ Number      29
Driver      c:\winnt\system32\drivers\q57w2k.sys
(77776, 2.75.0.0)

```

```

Name      [00000009] Compaq NC7781 Gigabit Server
Adapter Type      Ethernet 802.3
Product Name      Compaq NC7781 Gigabit Server
Adapter Type      Ethernet 802.3
Installed True
PNP Device ID      PCI\VEN_14E4&DEV_16A7&SUBSYS_00CB0E11&REV_0
2\3&13C0B0C5&0&10

```

```

Last Reset      2/11/2004 9:52:34 AM
Index      9
Service Name      q57w2k
IP Address      130.168.40.8
IP Subnet 255.255.0.0
Default IP Gateway Not Available
DHCP Enabled     False
DHCP Server      Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address      00:0B:CD:AF:7F:A6
Service Name      q57w2k
IRQ Number      30
Driver      c:\winnt\system32\drivers\q57w2k.sys
(77776, 2.75.0.0)

```

[Protocol]

Item	Value
Name	MSAFD Tcpip [TCP/IP]
ConnectionlessService	False
GuaranteesDelivery	True
GuaranteesSequencing	True
MaximumAddressSize	16 bytes
MaximumMessageSize	0 bytes
MessageOriented	False
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	False
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	True
SupportsGracefulClosing	True
SupportsGuaranteedBandwidth	False
SupportsMulticasting	False

Name	MSAFD Tcpip [UDP/IP]
ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	16 bytes
MaximumMessageSize	65467 bytes
MessageOriented	True
MinimumAddressSize	16 bytes
PseudoStreamOriented	False
SupportsBroadcasting	True
SupportsConnectData	False
SupportsDisconnectData	False
SupportsEncryption	False
SupportsExpeditedData	False
SupportsGracefulClosing	False
SupportsGuaranteedBandwidth	False
SupportsMulticasting	True

Name	RSVP UDP Service Provider
ConnectionlessService	True
GuaranteesDelivery	False
GuaranteesSequencing	False
MaximumAddressSize	16 bytes
MaximumMessageSize	65467 bytes
MessageOriented	True

MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption True
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting True

Name RSVP TCP Service Provider
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 16 bytes
MaximumMessageSize 0 bytes
MessageOriented False
MinimumAddressSize 16 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption True
SupportsExpeditedData True
SupportsGracefulClosing True
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS
[\\Device\\NetBT_Tcpip_{C91CC83D-E635-400B-86DD-8C81DC0D07FB}] SEQPACKE 5
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS
[\\Device\\NetBT_Tcpip_{C91CC83D-E635-400B-86DD-8C81DC0D07FB}] DATAGRAM 5
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False

SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS
[\\Device\\NetBT_Tcpip_{F3D5BAF3-12F3-4911-93C9-FFE1B41398D1}] SEQPACKE 6
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS
[\\Device\\NetBT_Tcpip_{F3D5BAF3-12F3-4911-93C9-FFE1B41398D1}] DATAGRAM 6
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS
[\\Device\\NetBT_Tcpip_{D90E04F2-3AD9-4F98-9464-751E106D7E6A}] SEQPACKE 1
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False

SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS
[\\Device\\NetBT_Tcpip_{D90E04F2-3AD9-4F98-9464-751E106D7E6A}] DATAGRAM 1
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS
[\\Device\\NetBT_Tcpip_{3F1BA297-E685-416B-82D7-70E771CC8745}] SEQPACKE 2
ConnectionlessService False
GuaranteesDelivery True
GuaranteesSequencing True
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting False
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

Name MSAFD NetBIOS
[\\Device\\NetBT_Tcpip_{3F1BA297-E685-416B-82D7-70E771CC8745}] DATAGRAM 2
ConnectionlessService True
GuaranteesDelivery False
GuaranteesSequencing False
MaximumAddressSize 20 bytes
MaximumMessageSize 64000 bytes
MessageOriented True
MinimumAddressSize 20 bytes
PseudoStreamOriented False
SupportsBroadcasting True
SupportsConnectData False
SupportsDisconnectData False
SupportsEncryption False
SupportsExpeditedData False
SupportsGracefulClosing False
SupportsGuaranteedBandwidth False
SupportsMulticasting False

[WinSock]

```

Item      Value
File      c:\winnt\system32\winsock.dll
Version   3.10
Size      2.80 KB (2,864 bytes)

```

```

File      c:\winnt\system32\wsock32.dll
Version   5.00.2195.2871
Size      21.27 KB (21,776 bytes)

```

[Ports]

[Following are sub-categories of this main category]

[Serial]

```

Item      Value
Name      COM1
Status    OK
PNP Device ID  ACPI\PNP0501\0
Maximum Input Buffer Size  0
Maximum Output Buffer Size  False
Settable Baud Rate  True
Settable Data Bits  True
Settable Flow Control  True
Settable Parity  True
Settable Parity Check  True
Settable Stop Bits  True
Settable RLSD  True
Supports RLSD  True
Supports 16 Bit Mode  False
Supports Special Characters  False
Baud Rate  9600
Bits/Byte  8
Stop Bits  1
Parity  None
Busy  0
Abort Read/Write on Error  0
Binary Mode Enabled  -1
Continue XMit on XOff  0
CTS Outflow Control  0
Discard NULL Bytes  0
DSR Outflow Control  0
DSR Sensitivity  0
DTR Flow Control Type  Enable
EOF Character  0
Error Replace Character  0
Error Replacement Enabled  0
Event Character  0
Parity Check Enabled  0
RTS Flow Control Type  Enable
XOff Character  19
XOffXMit Threshold  512
XOn Character  17
XOnXmit Threshold  2048
XOnXOff InFlow Control  0
XOnXOff OutFlow Control  0
IRQ Number  4
I/O Port  0x03F8-0x03FF

```

```

Driver      c:\winnt\system32\drivers\serial.sys
(62416, 5.00.2195.2780)

```

[Parallel]

```

Item      Value
No parallel port information

```

[Storage]

[Following are sub-categories of this main category]

[Drives]

```

Item      Value
Drive      A:
Description 3 1/2 Inch Floppy Drive

Drive      C:
Description Local Fixed Disk
Compressed False
File System NTFS
Size       16.95 GB (18,198,999,040 bytes)
Free Space 14.08 GB (15,120,396,288 bytes)
Volume Name
Volume Serial Number C8B488FA
Partition Disk #0, Partition #0
Partition Size 16.95 GB (18,199,003,136 bytes)
Starting Offset 16384 bytes
Drive Description Disk drive
Drive Manufacturer (Standard disk drives)
Drive Model COMPAQ LOGICAL VOLUME SCSI Disk
Device
Drive BytesPerSector 512
Drive MediaLoaded True
Drive MediaType Fixed hard disk media
Drive Partitions 1
Drive SCSI Bus 0
Drive SCSI LogicalUnit 0
Drive SCSI Port 2
Drive SCSI TargetId 4
Drive SectorsPerTrack 32
Drive Size 18203197440 bytes
Drive TotalCylinders 4357
Drive TotalSectors 35553120
Drive TotalTracks 1111035
Drive TracksPerCylinder 255

```

[SCSI]

```

Item      Value
Name      Compaq Smart Array 5i
Caption   Compaq Smart Array 5i
Driver    cpqcissm
Status    OK
PNP Device ID
PCI\VEN_0E11&DEV_B178&SUBSYS_40800E11&REV_0
1\3&267A616A&0&20

```

Device ID

```

PCI\VEN_0E11&DEV_B178&SUBSYS_40800E11&REV_0
1\3&267A616A&0&20
Device Map Not Available
Index Not Available
Max Number Controlled Not Available
IRQ Number 31
I/O Port 0x2800-0x28FF
Driver c:\winnt\system32\drivers\cpqcissm.sys
(14992, 5.40.2.0)

```

[Printing]

```

Name      Port Name Server Name
No printing information

```

[Problem Devices]

```

Device      PNP Device ID      Error Code
Base System Device
PCI\VEN_0E11&DEV_B203&SUBSYS_B2060E11&REV_0
1\3&267A616A&0&28 28
Base System Device
PCI\VEN_0E11&DEV_B204&SUBSYS_B2060E11&REV_0
1\3&267A616A&0&2A 28

```

[USB]

```

Device      PNP Device ID
Standard OpenHCD USB Host Controller
PCI\VEN_1166&DEV_0220&SUBSYS_02201166&REV_0
5\3&267A616A&0&7A
USB Root Hub USB\ROOT_HUB\4&AF5358C&0

```

[Software Environment]

[Following are sub-categories of this main category]

[Drivers]

Name	Description	File	Type
	Started	Start Mode	State
	Status	Error Control	Accept Pause
	Accept Stop		
abiosdsk	Abiosdsk	Not Available	Kernel Driver
	False	Disabled	Stopped
	Ignore	False	False
abp480n5	abp480n5	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
acpi	Microsoft ACPI Driver	c:\winnt\system32\drivers\acpi.sys	
	Kernel Driver	True	Boot
	Running	OK	Normal
	True		False
acpiec	ACPIEC	c:\winnt\system32\drivers\acpiec.sys	
	Kernel Driver	False	Disabled
	Stopped	OK	Normal
	False		False

adpu160m	adpu160m	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
afd	AFD Networking Support Environment		
	c:\winnt\system32\drivers\afd.sys		
	Kernel Driver	True	Auto
	Running	OK	Normal
	True		False
ahal54x	Ahal54x	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
aic116x	aic116x	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
aic78u2	aic78u2	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
aic78xx	aic78xx	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
alkernel	Altiris Kernel Driver		
	c:\winnt\system32\drivers\alkernel.sys		
	Kernel Driver	False	Manual
	Stopped	OK	Normal
	False		False
ami0nt	ami0nt	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
amsint	amsint	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
asc	asc	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
asc3350p	asc3350p	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
asc3550	asc3550	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
asynmac	RAS Asynchronous Media Driver		
	c:\winnt\system32\drivers\asynmac.sys		
	Kernel Driver	False	Manual
	Stopped	OK	Normal
	False		False
atapi	Standard IDE/ESDI Hard Disk Controller		
	c:\winnt\system32\drivers\atapi.sys		
	Kernel Driver	True	Boot
	Running	OK	Normal
	True		False
atdisk	Atdisk	Not Available	Kernel Driver
	False	Disabled	Stopped
	Ignore	False	False
atirage3	atirage3		
	c:\winnt\system32\drivers\atimpab.sys		
	Kernel Driver	True	Manual
	Running	OK	Ignore
	True		False
atmarpc	ATM ARP Client Protocol		
	c:\winnt\system32\drivers\atmarpc.sys		
	Kernel Driver	False	Manual
	Stopped	OK	Normal
	False		False

audstub	Audio Stub Driver		
	c:\winnt\system32\drivers\audstub.sys		
	Kernel Driver	True	Manual
	Running	OK	Normal
	True		False
beep	Beep		
	c:\winnt\system32\drivers\beep.sys		
	Kernel Driver	True	System
	Running	OK	Normal
	True		False
buslogic	BusLogic	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
cd20xrnt	cd20xrnt	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
cdaudio	Cdaudio		
	c:\winnt\system32\drivers\cdaudio.sys		
	Kernel Driver	False	System
	Stopped	OK	Ignore
	False		False
cdfs	Cdfs		
	c:\winnt\system32\drivers\cdfs.sys		
	File System Driver	True	Disabled
	Running	OK	Normal
	True		False
cdrom	CD-ROM Driver		
	c:\winnt\system32\drivers\cdrom.sys		
	Kernel Driver	True	System
	Running	OK	Normal
	True		False
changer	Changer	Not Available	Kernel Driver
	False	System	Stopped
	Ignore	False	False
cpqarray	CPqarray	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
cpqarray2	CPqarray2	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
cpqcissm	cpqcissm		
	c:\winnt\system32\drivers\cpqcissm.sys		
	Kernel Driver	True	Boot
	Running	OK	Normal
	True		False
cpqfcalm	cpqfcalm	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
cpqfws2e	cpqfws2e	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
dac960nt	dac960nt	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
deckzpsx	deckzpsx	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
dfsdriver	DfsDriver		
	c:\winnt\system32\drivers\dfs.sys		
	File System Driver	True	Boot
	Running	OK	Normal
	True		False
disk	Disk Driver		
	c:\winnt\system32\drivers\disk.sys		

	Kernel Driver	True	Boot
	Running	OK	Normal
	True		False
diskperf	Diskperf		
	c:\winnt\system32\drivers\diskperf.sys		
	Kernel Driver	True	Boot
	Running	OK	Normal
	True		False
dmboot	dmboot		
	c:\winnt\system32\drivers\dmboot.sys		
	Kernel Driver	False	Disabled
	Stopped	OK	Normal
	False		False
dmio	Logical Disk Manager Driver		
	c:\winnt\system32\drivers\dmio.sys		
	Kernel Driver	True	Boot
	Running	OK	Normal
	True		False
dmload	dmload		
	c:\winnt\system32\drivers\dmload.sys		
	Kernel Driver	True	Boot
	Running	OK	Normal
	True		False
efs	EFS	c:\winnt\system32\drivers\efs.sys	
	File System Driver	True	Disabled
	Running	OK	Normal
	True		False
fastfat	Fastfat		
	c:\winnt\system32\drivers\fastfat.sys		
	File System Driver	True	Disabled
	Running	OK	Normal
	True		False
fd16_700	Fd16_700	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
fdc	Floppy Disk Controller Driver		
	c:\winnt\system32\drivers\fdc.sys		
	Kernel Driver	True	Manual
	Running	OK	Normal
	True		False
fips	Fips		
	c:\winnt\system32\drivers\fips.sys		
	Kernel Driver	True	Auto
	Running	OK	Normal
	True		False
fireport	fireport	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
flashpnt	flashpnt	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
flpydisk	Floppy Disk Driver		
	c:\winnt\system32\drivers\flpydisk.sys		
	Kernel Driver	True	Manual
	Running	OK	Normal
	True		False
ftdisk	Volume Manager Driver		
	c:\winnt\system32\drivers\ftdisk.sys		
	Kernel Driver	True	Boot
	Running	OK	Normal
	True		False
gpc	Generic Packet Classifier		
	c:\winnt\system32\drivers\msgpc.sys		

	Kernel Driver	True	Manual
	Running	OK	Normal
	True	False	False
i8042prt	i8042 Keyboard and PS/2 Mouse Port Driver		
	c:\winnt\system32\drivers\i8042prt.sys		
	Kernel Driver	True	System
	Running	OK	Normal
	True	False	False
ini910u	ini910u	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
intellide	IntelIde	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
ipfilterdriver	IP Traffic Filter Driver		
	c:\winnt\system32\drivers\ipfltdrv.sys		
	Kernel Driver	False	Manual
	Stopped	OK	Normal
	True	False	False
ipinip	IP in IP Tunnel Driver		
	c:\winnt\system32\drivers\ipinip.sys		
	Kernel Driver	False	Manual
	Stopped	OK	Normal
	True	False	False
ipnat	IP Network Address Translator		
	c:\winnt\system32\drivers\ipnat.sys		
	Kernel Driver	False	Manual
	Stopped	OK	Normal
	True	False	False
ipsec	IPSEC driver		
	c:\winnt\system32\drivers\ipsec.sys		
	Kernel Driver	False	Manual
	Stopped	OK	Normal
	True	False	False
ipsraidn	ipsraidn	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
isapnp	PnP ISA/EISA Bus Driver		
	c:\winnt\system32\drivers\isapnp.sys		
	Kernel Driver	True	Boot
	Running	OK	Critical
	True	False	False
kbdclass	Keyboard Class Driver		
	c:\winnt\system32\drivers\kbdclass.sys		
	Kernel Driver	True	System
	Running	OK	Normal
	True	False	False
ksecdd	KSecDD		
	c:\winnt\system32\drivers\ksecdd.sys		
	Kernel Driver	True	Boot
	Running	OK	Normal
	True	False	False
lbrtfdc	lbrtfdc	Not Available	Kernel Driver
	False	System	Stopped
	Ignore	False	False
lp6nds35	lp6nds35	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
mmdd	mmdd		
	c:\winnt\system32\drivers\mmdd.sys		
	Kernel Driver	True	System
	Running	OK	Ignore
	True	False	False

modem	Modem		
	c:\winnt\system32\drivers\modem.sys		
	Kernel Driver	False	Manual
	Stopped	OK	Ignore
	True	False	False
mouclass	Mouse Class Driver		
	c:\winnt\system32\drivers\mouclass.sys		
	Kernel Driver	True	System
	Running	OK	Normal
	True	False	False
mountmgr	MountMgr		
	c:\winnt\system32\drivers\mountmgr.sys		
	Kernel Driver	True	Boot
	Running	OK	Normal
	True	False	False
mraid35x	mraid35x	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
mrxsmb	MRXSMB		
	c:\winnt\system32\drivers\mrxsmb.sys		
	File System Driver	True	System
	Running	OK	Normal
	True	False	False
msfs	Msfs		
	c:\winnt\system32\drivers\msfs.sys		
	File System Driver	True	System
	Running	OK	Normal
	True	False	False
mskssrv	Microsoft Streaming Service Proxy		
	c:\winnt\system32\drivers\mskssrv.sys		
	Kernel Driver	False	Manual
	Stopped	OK	Normal
	True	False	False
mspclock	Microsoft Streaming Clock Proxy		
	c:\winnt\system32\drivers\mspclock.sys		
	Kernel Driver	False	Manual
	Stopped	OK	Normal
	True	False	False
mspqm	Microsoft Streaming Quality Manager Proxy		
	c:\winnt\system32\drivers\mspqm.sys		
	Kernel Driver	False	Manual
	Stopped	OK	Normal
	True	False	False
mup	Mup		
	c:\winnt\system32\drivers\mup.sys		
	File System Driver	True	Boot
	Running	OK	Normal
	True	False	False
n100	Compaq Ethernet or Fast Ethernet NIC NT Driver		
	c:\winnt\system32\drivers\n100nt5.sys		
	Kernel Driver	False	Manual
	Stopped	OK	Normal
	True	False	False
ncrc710	Ncrc710	Not Available	Kernel Driver
	False	Disabled	Stopped
	Normal	False	False
ndis	NDIS System Driver		
	c:\winnt\system32\drivers\ndis.sys		
	Kernel Driver	True	Boot
	Running	OK	Normal
	True	False	False
ndistapi	Remote Access NDIS TAPI Driver		
	c:\winnt\system32\drivers\ndistapi.sys		
	Kernel Driver	True	Manual

	Running	OK	Normal	False
	True	False	False	False
ndiswan	Remote Access NDIS WAN Driver			
	c:\winnt\system32\drivers\ndiswan.sys			
	Kernel Driver	True	Manual	
	Running	OK	Normal	False
	True	False	False	False
ndproxy	NDIS Proxy			
	c:\winnt\system32\drivers\ndproxy.sys			
	Kernel Driver	True	Manual	
	Running	OK	Normal	False
	True	False	False	False
netbios	NetBIOS Interface			
	c:\winnt\system32\drivers\netbios.sys			
	File System Driver	True	System	
	Running	OK	Normal	False
	True	False	False	False
netbt	NetBios over Tcpip			
	c:\winnt\system32\drivers\netbt.sys			
	Kernel Driver	True	System	
	Running	OK	Normal	False
	True	False	False	False
netdetect	NetDetect			
	c:\winnt\system32\drivers\netdetect.sys			
	Kernel Driver	False	Manual	
	Stopped	OK	Normal	False
	True	False	False	False
npfs	Npfs			
	c:\winnt\system32\drivers\npfs.sys			
	File System Driver	True	System	
	Running	OK	Normal	False
	True	False	False	False
ntfs	Ntfs			
	c:\winnt\system32\drivers\ntfs.sys			
	File System Driver	True	Disabled	
	Running	OK	Normal	False
	True	False	False	False
null	Null			
	c:\winnt\system32\drivers\null.sys			
	Kernel Driver	True	System	
	Running	OK	Normal	False
	True	False	False	False
nwlnkflt	IPX Traffic Filter Driver			
	c:\winnt\system32\drivers\nwlnkflt.sys			
	Kernel Driver	False	Manual	
	Stopped	OK	Normal	False
	True	False	False	False
nwlnkfld	IPX Traffic Forwarder Driver			
	c:\winnt\system32\drivers\nwlnkfld.sys			
	Kernel Driver	False	Manual	
	Stopped	OK	Normal	False
	True	False	False	False
openhci	Microsoft USB Open Host Controller Driver			
	c:\winnt\system32\drivers\openhci.sys			
	Kernel Driver	True	Manual	
	Running	OK	Normal	False
	True	False	False	False
parallel	Parallel			
	c:\winnt\system32\drivers\parallel.sys			
	Kernel Driver	False	Auto	
	Stopped	OK	Ignore	False
	True	False	False	False

parport	Parport				
	c:\winnt\system32\drivers\parport.sys				
	Kernel Driver	False	Auto		
	Stopped	OK	Ignore	False	
	False				
partmgr	PartMgr				
	c:\winnt\system32\drivers\partmgr.sys				
	Kernel Driver	True	Auto		
	Running	OK	Normal	False	
	True				
parvdm	ParVdm				
	c:\winnt\system32\drivers\parvdm.sys				
	Kernel Driver	False	Auto		
	Stopped	OK	Ignore	False	
	False				
pci	PCI Bus Driver				
	c:\winnt\system32\drivers\pci.sys				
	Kernel Driver	True	Boot		
	Running	OK	Critical	False	
	True				
pcidump	PCIDump	Not Available		Kernel Driver	
	False	System	Stopped	OK	
	Ignore	False	False		
pciide	PCIide				
	c:\winnt\system32\drivers\pciide.sys				
	Kernel Driver	True	Boot		
	Running	OK	Normal	False	
	True				
pcmcia	Pcmcia				
	c:\winnt\system32\drivers\pcmcia.sys				
	Kernel Driver	False	Disabled		
	Stopped	OK	Normal	False	
	False				
pdcomp	PDCOMP	Not Available		Kernel Driver	
	False	Manual	Stopped	OK	
	Ignore	False	False		
pdframe	PDFRAME	Not Available		Kernel Driver	
	False	Manual	Stopped	OK	
	Ignore	False	False		
pdreli	PDRELI	Not Available		Kernel Driver	
	False	Manual	Stopped	OK	
	Ignore	False	False		
pdrframe	PDRFRAME	Not Available		Kernel Driver	
	False	Manual	Stopped	OK	
	Ignore	False	False		
pptpminiport	WAN Miniport (PPTP)				
	c:\winnt\system32\drivers\rasppptp.sys				
	Kernel Driver	True	Manual		
	Running	OK	Normal	False	
	True				
ptilink	Direct Parallel Link Driver				
	c:\winnt\system32\drivers\ptilink.sys				
	Kernel Driver	True	Manual		
	Running	OK	Normal	False	
	True				
q57w2k	Compaq NC7781 Gigabit Server Adapter				
	c:\winnt\system32\drivers\q57w2k.sys				
	Kernel Driver	True	Manual		
	Running	OK	Normal	False	
	True				
ql1080	ql1080	Not Available		Kernel Driver	
	False	Disabled	Stopped	OK	
	Normal	False	False		

ql10wnt	Ql10wnt	Not Available		Kernel Driver	
	False	Disabled	Stopped	OK	
	Normal	False	False		
ql1240	ql1240	Not Available		Kernel Driver	
	False	Disabled	Stopped	OK	
	Normal	False	False		
ql2100	ql2100	Not Available		Kernel Driver	
	False	Disabled	Stopped	OK	
	Normal	False	False		
rasacd	Remote Access Auto Connection Driver				
	c:\winnt\system32\drivers\rasacd.sys				
	Kernel Driver	True	System		
	Running	OK	Normal	False	
	True				
rasl2tp	WAN Miniport (L2TP)				
	c:\winnt\system32\drivers\rasl2tp.sys				
	Kernel Driver	True	Manual		
	Running	OK	Normal	False	
	True				
raspti	Direct Parallel				
	c:\winnt\system32\drivers\raspti.sys				
	Kernel Driver	True	Manual		
	Running	OK	Normal	False	
	True				
rca	Microsoft Streaming Network Raw Channel				
Access	c:\winnt\system32\drivers\rca.sys				
	Kernel Driver	False	Manual		
	Stopped	OK	Normal	False	
	False				
rdbss	Rdbss				
	c:\winnt\system32\drivers\rdbss.sys				
	File System Driver	True	System		
	Running	OK	Normal	False	
	True				
rdpdr	Terminal Server Device Redirector Driver				
	c:\winnt\system32\drivers\rdpdr.sys				
	Kernel Driver	True	Manual		
	Running	OK	Normal	False	
	True				
rdpwd	RDPWD				
	c:\winnt\system32\drivers\rdpwd.sys				
	Kernel Driver	True	Manual		
	Running	OK	Ignore	False	
	True				
redbook	Digital CD Audio Playback Filter Driver				
	c:\winnt\system32\drivers\redbook.sys				
	Kernel Driver	False	System		
	Stopped	OK	Normal	False	
	False				
serenum	Serenum Filter Driver				
	c:\winnt\system32\drivers\serenum.sys				
	Kernel Driver	True	Manual		
	Running	OK	Normal	False	
	True				
serial	Serial port driver				
	c:\winnt\system32\drivers\serial.sys				
	Kernel Driver	True	System		
	Running	OK	Ignore	False	
	True				
sfloppy	Sfloppy				
	c:\winnt\system32\drivers\sfloppy.sys				
	Kernel Driver	False	System		

	Stopped	OK	Ignore	False	
	False				
sglfb	sglfb	Not Available		Kernel Driver	
	False	System	Stopped	OK	
	Normal	False	False		
simbad	Simbad	Not Available		Kernel Driver	
	False	Disabled	Stopped	OK	
	Normal	False	False		
sparrow	Sparrow	Not Available		Kernel Driver	
	False	Disabled	Stopped	OK	
	Normal	False	False		
spud	Special Purpose Utility Driver				
	c:\winnt\system32\drivers\spud.sys				
	Kernel Driver	True	Manual		
	Running	OK	Normal	False	
	True				
srv	Srv				
	c:\winnt\system32\drivers\srv.sys				
	File System Driver	True	Manual		
	Running	OK	Normal	False	
	True				
swenum	Software Bus Driver				
	c:\winnt\system32\drivers\swenum.sys				
	Kernel Driver	True	Manual		
	Running	OK	Normal	False	
	True				
symc810	symc810	Not Available		Kernel Driver	
	False	Disabled	Stopped	OK	
	Normal	False	False		
symc8xx	symc8xx	Not Available		Kernel Driver	
	False	Disabled	Stopped	OK	
	Normal	False	False		
sym_hi	sym_hi	Not Available		Kernel Driver	
	False	Disabled	Stopped	OK	
	Normal	False	False		
tcpip	TCP/IP Protocol Driver				
	c:\winnt\system32\drivers\tcpip.sys				
	Kernel Driver	True	System		
	Running	OK	Normal	False	
	True				
tdasync	TDASYN				
	c:\winnt\system32\drivers\tdasync.sys				
	Kernel Driver	False	Manual		
	Stopped	OK	Ignore	False	
	False				
tdipx	TDIPX				
	c:\winnt\system32\drivers\tdipx.sys				
	Kernel Driver	False	Manual		
	Stopped	OK	Ignore	False	
	False				
tdnetb	TDNETB				
	c:\winnt\system32\drivers\tdnetb.sys				
	Kernel Driver	False	Manual		
	Stopped	OK	Ignore	False	
	False				
tdpipe	TDPIPE				
	c:\winnt\system32\drivers\tdpipe.sys				
	Kernel Driver	False	Manual		
	Stopped	OK	Ignore	False	
	False				
tdspix	TDSPX				
	c:\winnt\system32\drivers\tdspix.sys				
	Kernel Driver	False	Manual		

```

Stopped OK Ignore False
False
tdtcp TDTCP
c:\winnt\system32\drivers\tdtcp.sys
Kernel Driver True Manual
Running OK Ignore False
True
termdd Terminal Device Driver
c:\winnt\system32\drivers\termdd.sys
Kernel Driver True Auto
Running OK Normal False
True
tga tga Not Available Kernel Driver
False System Stopped OK
Ignore False False
udfs Udfs
c:\winnt\system32\drivers\udfs.sys
File System Driver False Disabled
Stopped OK Normal False
False
ultra66 ultra66 Not Available Kernel Driver
False Disabled Stopped OK
Normal False False
update Microcode Update Driver
c:\winnt\system32\drivers\update.sys
Kernel Driver True Manual
Running OK Normal False
True
usbhub Microsoft USB Standard Hub Driver
c:\winnt\system32\drivers\usbhub.sys
Kernel Driver True Manual
Running OK Normal False
True
vgasave VgaSave c:\winnt\system32\drivers\vga.sys
Kernel Driver True System
Running OK Ignore False
True
wanarp Remote Access IP ARP Driver
c:\winnt\system32\drivers\wanarp.sys
Kernel Driver True Manual
Running OK Normal False
True
wdica WDICA Not Available Kernel Driver
False Manual Stopped OK
Ignore False False

[Environment Variables]

Variable Value User Name
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
Os2LibPath %SystemRoot%\system32\os2\dll;
<SYSTEM>
Path %SystemRoot%\system32;%SystemRoot%;%SystemR
oot%\System32\Wbem;C:\Program Files\Microsoft SQL
Server\80\Tools\BINN <SYSTEM>
windir %SystemRoot% <SYSTEM>
OS Windows_NT <SYSTEM>
PROCESSOR_ARCHITECTURE x86 <SYSTEM>
PROCESSOR_LEVEL 15 <SYSTEM>
PROCESSOR_IDENTIFIER x86 Family 15 Model 2
Stepping 5, GenuineIntel <SYSTEM>
PROCESSOR_REVISION 0205 <SYSTEM>

```

```

NUMBER_OF_PROCESSORS 4 <SYSTEM>
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF
;.WSH <SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>
TMP %USERPROFILE%\Local Settings\Temp
CL8\Administrator
TMP %USERPROFILE%\Local Settings\Temp
CL8\Administrator

[Jobs]

[ Following are sub-categories of this main category ]

[Print]

Document Size Owner Notify Status
Time Submitted Start Time
Until Time Elapsed Time
Pages Printed Job ID Priority
Parameters Driver Name
Print Processor Host Print Queue
Data Type Name
Unknown Unknown Unknown Unknown
Unknown Unknown Unknown Unknown
Unknown Unknown Unknown Unknown
Unknown Unknown Unknown

[Network Connections]

Local Name Remote Name Type
Status User Name
No network connections information

[Running Tasks]

Name Path Process ID Priority Min
Working Set Max Working Set Start Time
Version Size File Date
system idle process Not Available 0 0
Not Available Not Available Not
Available Unknown Unknown Unknown
system Not Available 8 8 0
1413120 Not Available Unknown
Unknown Unknown
smss.exe c:\winnt\system32\smss.exe 192 11
204800 1413120 2/11/2004 3:52:45 PM
5.00.2195.2901 44.27 KB (45,328 bytes)
12/7/1999 7:00:00 AM
csrss.exe Not Available 216 13 Not
Available Not Available 2/11/2004 3:52:48 PM
Unknown Unknown Unknown
winlogon.exe c:\winnt\system32\winlogon.exe
212 13 204800 1413120
2/11/2004 3:52:49 PM
5.00.2195.2953 173.77 KB (177,936
bytes) 12/7/1999 7:00:00 AM
services.exe c:\winnt\system32\services.exe
268 9 204800 1413120

```

```

2/11/2004 3:52:49 PM
5.00.2195.2780 86.77 KB (88,848 bytes)
12/7/1999 7:00:00 AM
lsass.exe c:\winnt\system32\lsass.exe 280 9
204800 1413120 2/11/2004 3:52:49 PM
5.00.2195.2964 32.77 KB (33,552 bytes)
12/7/1999 7:00:00 AM
termsrv.exe c:\winnt\system32\termsrv.exe 392
10 204800 1413120 2/11/2004
3:52:50 PM 5.00.2195.2342 137.27 KB
(140,560 bytes) 9/13/2002 6:09:44 PM
svchost.exe c:\winnt\system32\svchost.exe 476
8 204800 1413120 2/11/2004
3:52:52 PM 5.00.2134.1 7.77 KB
(7,952 bytes) 12/7/1999 7:00:00 AM
msdtc.exe c:\winnt\system32\msdtc.exe 500 8
204800 1413120 2/11/2004 3:52:52 PM
1999.9.3421.3 6.77 KB (6,928 bytes)
9/13/2002 5:45:07 PM
svchost.exe c:\winnt\system32\svchost.exe 680
8 204800 1413120 2/11/2004
3:52:52 PM 5.00.2134.1 7.77 KB
(7,952 bytes) 12/7/1999 7:00:00 AM
rsys.exe c:\benchcraft\rsys.exe 696 8
204800 1413120 2/11/2004 3:52:52 PM
Not Available 32.00 KB (32,768 bytes)
9/17/2002 4:43:40 PM
winmgmt.exe c:\winnt\system32\wbem\winmgmt.exe 732
8 204800 1413120 2/11/2004
3:52:53 PM 1.50.1085.0029 192.08 KB
(196,685 bytes) 9/13/2002 6:09:52 PM
inetinfo.exe c:\winnt\system32\inetinfo.exe 752
8 204800 1413120 2/11/2004
3:52:53 PM 5.00.0984 14.27 KB (14,608 bytes)
9/13/2002 6:10:42 PM
logon.scr c:\winnt\system32\logon.scr 708 4
204800 1413120 2/11/2004 4:08:02 PM
5.00.2195.2104 127.77 KB (130,832
bytes) 9/13/2002 6:09:26 PM
csrss.exe Not Available 768 13 Not
Available Not Available 2/11/2004 4:09:25 PM
Unknown Unknown Unknown
winlogon.exe c:\winnt\system32\winlogon.exe
964 13 204800 1413120
2/11/2004 4:09:25 PM
5.00.2195.2953 173.77 KB (177,936
bytes) 12/7/1999 7:00:00 AM
rdpclip.exe c:\winnt\system32\rdpclip.exe 968
8 204800 1413120 2/11/2004
4:09:31 PM 5.00.2174.1 39.77 KB
(40,720 bytes) 9/13/2002 5:45:10 PM
explorer.exe c:\winnt\explorer.exe
1064 8 204800 1413120
2/11/2004 4:09:32 PM
5.00.3315.2846 237.27 KB (242,960
bytes) 9/13/2002 6:09:47 PM
aclntusr.exe c:\program
files\altiris\aclnt\aclntusr.exe 1120 8
204800 1413120 2/11/2004 4:09:32 PM
5, 6, 0, 50 176.00 KB (180,224
bytes) 6/5/2003 1:56:24 PM

```

```

tardis.exe c:\program files\tardis 2000
v1.4\tardis.exe 1168 8 204800
1413120 2/11/2004 4:09:33 PM 5,
0, 1, 4 308.00 KB (315,392 bytes) 9/13/2002
6:21:25 PM
mmc.exe c:\winnt\system32\mmc.exe 1216 8
204800 1413120 2/11/2004 4:09:38 PM
5.00.2195.2301 589.27 KB (603,408
bytes) 9/13/2002 6:09:26 PM
rsvp.exe c:\winnt\system32\rsvp.exe 1364 8
204800 1413120 2/11/2004 4:10:06 PM
5.00.2167.1 172.77 KB (176,912
bytes) 12/7/1999 7:00:00 AM

[Loaded Modules]

Name Version Size File Date Manufacturer
Path
traffic.dll 5.00.2139.1 30.77 KB
(31,504 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\traffic.dll
rsvp.exe 5.00.2167.1 172.77 KB (176,912
bytes) 12/7/1999 7:00:00 AM Microsoft
Corporation c:\winnt\system32\rsvp.exe
wbemprox.dll 1.50.1085.0045 40.08 KB
(41,040 bytes) 9/13/2002 6:09:52 PM
Microsoft Corporation
c:\winnt\system32\wbem\wbemprox.dll
mlang.dll 5.00.3103.1000 510.77 KB (523,024
bytes) 9/13/2002 6:09:26 PM Microsoft
Corporation c:\winnt\system32\mlang.dll
cabinet.dll 5.00.2147.1 54.77 KB
(56,080 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\cabinet.dll
msinfo32.dll 5.00.2177.1 312.27 KB
(319,760 bytes) 9/13/2002 5:46:00 PM
Microsoft Corporation c:\program
files\common files\microsoft
shared\msinfo\msinfo32.dll
mmcndmgr.dll 5.00.2178.1 815.27 KB
(834,832 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\mmcndmgr.dll
mfc42u.dll 6.00.8665.0 972.05 KB
(995,384 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\mfc42u.dll
mmc.exe 5.00.2195.2301 589.27 KB (603,408
bytes) 9/13/2002 6:09:26 PM Microsoft
Corporation c:\winnt\system32\mmc.exe
rapilib.dll 5.00.2195.2717 24.77 KB
(25,360 bytes) 9/13/2002 6:09:39 PM
Microsoft Corporation
c:\winnt\system32\rapilib.dll
rsvpsp.dll 5.00.2195.2749 74.77 KB
(76,560 bytes) 9/13/2002 6:09:40 PM
Microsoft Corporation
c:\winnt\system32\rsvpsp.dll
comdlg32.dll 5.00.3103.1000 236.77 KB
(242,448 bytes) 12/7/1999 7:00:00 AM

```

```

Microsoft Corporation
c:\winnt\system32\comdlg32.dll
tardis.exe 5, 0, 1, 4 308.00 KB
(315,392 bytes) 9/13/2002 6:21:25 PM
H.C.Mingham-Smith Ltd. c:\program
files\tardis 2000 v1.4\tardis.exe
aclntusr.exe 5, 6, 0, 50 176.00 KB
(180,224 bytes) 6/5/2003 1:56:24 PM
c:\program
files\altiris\aclient\aclntusr.exe
wininet.dll 5.00.3315.1000 456.77 KB
(467,728 bytes) 9/13/2002 6:09:46 PM
Microsoft Corporation
c:\winnt\system32\wininet.dll
linkinfo.dll 5.00.2134.1 15.77 KB
(16,144 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\linkinfo.dll
powrprof.dll 5.00.3103.1000 13.27 KB
(13,584 bytes) 9/13/2002 6:09:38 PM
Microsoft Corporation
c:\winnt\system32\powrprof.dll
batmeter.dll 5.00.3103.1000 20.27 KB
(20,752 bytes) 9/13/2002 6:09:14 PM
Microsoft Corporation
c:\winnt\system32\batmeter.dll
stobject.dll 5.00.2195.2780 79.27 KB
(81,168 bytes) 9/13/2002 6:09:43 PM
Microsoft Corporation
c:\winnt\system32\stobject.dll
msi.dll 1.11.2405.0 1.69 MB (1,767,184
bytes) 9/13/2002 6:09:29 PM Microsoft
Corporation c:\winnt\system32\msi.dll
webcheck.dll 5.00.3315.1000 251.77 KB
(257,808 bytes) 9/13/2002 6:09:45 PM
Microsoft Corporation
c:\winnt\system32\webcheck.dll
ntshrui.dll 5.00.2134.1 46.77 KB
(47,888 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\ntshrui.dll
mydocs.dll 5.00.2920.0000 55.77 KB
(57,104 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\mydocs.dll
browseui.dll 5.00.3315.2846 788.77 KB
(807,696 bytes) 9/13/2002 6:09:14 PM
Microsoft Corporation
c:\winnt\system32\browseui.dll
shdocvw.dll 5.00.3315.2879 1.05 MB
(1,104,144 bytes) 9/13/2002 6:09:42 PM
Microsoft Corporation
c:\winnt\system32\shdocvw.dll
explorer.exe 5.00.3315.2846 237.27 KB
(242,960 bytes) 9/13/2002 6:09:47 PM
Microsoft Corporation
c:\winnt\explorer.exe
rdpclip.exe 5.00.2174.1 39.77 KB
(40,720 bytes) 9/13/2002 5:45:10 PM
Microsoft Corporation
c:\winnt\system32\rdpclip.exe

```

```

cscui.dll 5.00.2195.2959 228.27 KB (233,744
bytes) 9/13/2002 6:09:17 PM Microsoft
Corporation c:\winnt\system32\cscui.dll
logon.scr 5.00.2195.2104 127.77 KB (130,832
bytes) 9/13/2002 6:09:26 PM Microsoft
Corporation c:\winnt\system32\logon.scr
iislog.dll 5.00.0984 75.27 KB (77,072 bytes)
9/13/2002 6:10:42 PM Microsoft
Corporation
c:\winnt\system32\iislog.dll
httpext.dll 0.9.3940.21 435.27 KB
(445,712 bytes) 9/13/2002 6:10:42 PM
Microsoft Corporation
c:\winnt\system32\iislog\httpext.dll
fpexedll.dll 4.0.2.4324 20.06 KB
(20,541 bytes) 9/13/2002 6:10:33 PM
Microsoft Corporation c:\program
files\common files\microsoft shared\web server
extensions\40\bin\fpexedll.dll
md5filt.dll 5.00.0984 32.77 KB (33,552 bytes)
9/13/2002 6:10:43 PM Microsoft
Corporation
c:\winnt\system32\iislog\md5filt.dll
gzip.dll 5.00.0984 30.27 KB (30,992 bytes)
9/13/2002 6:10:42 PM Microsoft
Corporation
c:\winnt\system32\iislog\gzip.dll
compfilt.dll 5.00.0984 22.77 KB (23,312 bytes)
9/13/2002 6:10:41 PM Microsoft
Corporation
c:\winnt\system32\iislog\compfilt.dll
sspifilt.dll 5.00.0984 43.27 KB (44,304 bytes)
9/13/2002 6:10:43 PM Microsoft
Corporation
c:\winnt\system32\iislog\sspifilt.dll
iscomlog.dll 5.00.0984 24.77 KB (25,360 bytes)
9/13/2002 6:10:43 PM Microsoft
Corporation
c:\winnt\system32\iislog\iscomlog.dll
lonsint.dll 5.00.0984 11.77 KB (12,048 bytes)
9/13/2002 6:10:43 PM Microsoft
Corporation
c:\winnt\system32\iislog\lonsint.dll
inetsloc.dll 5.00.0984 20.27 KB (20,752 bytes)
9/13/2002 6:09:24 PM Microsoft
Corporation c:\winnt\system32\iislog\inetsloc.dll
iisfecnv.dll 5.00.0984 7.27 KB (7,440 bytes)
9/13/2002 5:45:32 PM Microsoft
Corporation
c:\winnt\system32\iislog\iisfecnv.dll
isatq.dll 5.00.0984 60.27 KB (61,712 bytes)
9/13/2002 6:10:43 PM Microsoft
Corporation
c:\winnt\system32\iislog\isatq.dll
infocomm.dll 5.00.0984 238.27 KB (243,984
bytes) 9/13/2002 6:10:43 PM Microsoft
Corporation
c:\winnt\system32\iislog\infocomm.dll
w3svc.dll 5.00.0984 343.27 KB (351,504 bytes)
9/13/2002 6:10:44 PM Microsoft
Corporation
c:\winnt\system32\iislog\w3svc.dll

```

```

security.dll      5.00.2154.1      5.77 KB
(5,904 bytes)    12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\security.dll
svcxext.dll      5.00.0984 39.77 KB (40,720 bytes)
9/13/2002 6:10:44 PM      Microsoft
Corporation
c:\winnt\system32\inetsrv\svcxext.dll
admexs.dll       5.00.0984 27.77 KB (28,432 bytes)
9/13/2002 6:10:41 PM      Microsoft
Corporation
c:\winnt\system32\inetsrv\admexs.dll
wamreg.dll       5.00.0984 45.77 KB (46,864 bytes)
9/13/2002 6:10:44 PM      Microsoft
Corporation
c:\winnt\system32\inetsrv\wamreg.dll
metadata.dll     5.00.0984 68.77 KB (70,416 bytes)
9/13/2002 6:10:43 PM      Microsoft
Corporation
c:\winnt\system32\inetsrv\metadata.dll
iismap.dll       5.00.0984 55.77 KB (57,104 bytes)
9/13/2002 6:09:23 PM      Microsoft
Corporation
c:\winnt\system32\iismap.dll
nsepm.dll        5.00.0984 43.27 KB (44,304 bytes)
9/13/2002 6:10:43 PM      Microsoft
Corporation
c:\winnt\system32\inetsrv\nsepm.dll
admwprox.dll    5.00.0984 31.77 KB (32,528 bytes)
9/13/2002 5:45:33 PM      Microsoft
Corporation
c:\winnt\system32\admwprox.dll
coadmin.dll     5.00.0984 39.27 KB (40,208 bytes)
9/13/2002 6:10:41 PM      Microsoft
Corporation
c:\winnt\system32\inetsrv\coadmin.dll
iisadmin.dll    5.00.0984 15.27 KB (15,632 bytes)
9/13/2002 6:10:42 PM      Microsoft
Corporation
c:\winnt\system32\inetsrv\iisadmin.dll
rpcpref.dll     5.00.0984 4.27 KB (4,368 bytes)
9/13/2002 6:10:43 PM      Microsoft
Corporation
c:\winnt\system32\inetsrv\rpcpref.dll
iisrtl.dll      5.00.0984 119.77 KB (122,640
bytes) 9/13/2002 6:09:23 PM      Microsoft
Corporation
c:\winnt\system32\iisrtl.dll
inetinfo.exe    5.00.0984 14.27 KB (14,608 bytes)
9/13/2002 6:10:42 PM      Microsoft
Corporation
c:\winnt\system32\inetsrv\inetinfo.exe
netui1.dll      5.00.2134.1      210.27 KB
(215,312 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\netui1.dll
netui0.dll      5.00.2134.1      70.27 KB
(71,952 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\netui0.dll
ntlanman.dll    5.00.2157.1      35.27 KB
(36,112 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\ntlanman.dll
wshnetbs.dll   5.00.2134.1      7.77 KB
(7,952 bytes) 12/7/1999 7:00:00 AM

```

```

Microsoft Corporation
c:\winnt\system32\wshnetbs.dll
ntmarta.dll     5.00.2195.2862  98.77 KB
(101,136 bytes) 9/13/2002 6:09:35 PM
Microsoft Corporation
c:\winnt\system32\ntmarta.dll
provthrd.dll   1.50.1085.0000  68.07 KB
(69,708 bytes) 9/13/2002 5:45:53 PM
Microsoft Corporation
c:\winnt\system32\wbem\provthrd.dll
ntevt.dll      1.50.1085.0000  192.06 KB (196,669
bytes) 12/7/1999 7:00:00 AM      Microsoft
Corporation
c:\winnt\system32\wbem\ntevt.dll
perfos.dll     5.00.2155.1      21.27 KB
(21,776 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\perfos.dll
psapi.dll      5.00.2134.1      28.27 KB (28,944 bytes)
12/7/1999 7:00:00 AM      Microsoft
Corporation
c:\winnt\system32\psapi.dll
framedyn.dll  1.50.1085.0000  164.05 KB
(167,992 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\wbem\framedyn.dll
cimwin32.dll   1.50.1085.0038  1.02 MB
(1,073,232 bytes) 9/13/2002 6:09:50 PM
Microsoft Corporation
c:\winnt\system32\wbem\cimwin32.dll
wbemsvc.dll   1.50.1085.0007  40.07 KB
(41,036 bytes) 9/13/2002 6:09:52 PM
Microsoft Corporation
c:\winnt\system32\wbem\wbemsvc.dll
wbemess.dll   1.50.1085.0039  364.07 KB
(372,804 bytes) 9/13/2002 6:09:52 PM
Microsoft Corporation
c:\winnt\system32\wbem\wbemess.dll
fastprox.dll  1.50.1085.0037  144.08 KB
(147,536 bytes) 9/13/2002 6:09:51 PM
Microsoft Corporation
c:\winnt\system32\wbem\fastprox.dll
wbemcore.dll  1.50.1085.0036  628.07 KB
(643,140 bytes) 9/13/2002 6:09:52 PM
Microsoft Corporation
c:\winnt\system32\wbem\wbemcore.dll
wbemcomn.dll  1.50.1085.0021  692.07 KB
(708,675 bytes) 9/13/2002 6:09:51 PM
Microsoft Corporation
c:\winnt\system32\wbem\wbemcomn.dll
winmgmt.exe   1.50.1085.0029  192.08 KB
(196,685 bytes) 9/13/2002 6:09:52 PM
Microsoft Corporation
c:\winnt\system32\wbem\winmgmt.exe
rsys.exe      Not Available    32.00 KB (32,768 bytes)
9/17/2002 4:43:40 PM      Not Available
c:\benchcraft\rsys.exe
wmi.dll       5.00.2191.1      6.27 KB (6,416 bytes)
12/7/1999 7:00:00 AM      Microsoft
Corporation
c:\winnt\system32\wmi.dll
netshell.dll  5.00.2195.2779  457.27 KB
(468,240 bytes) 9/13/2002 6:09:34 PM
Microsoft Corporation
c:\winnt\system32\netshell.dll

```

```

netman.dll     5.00.2195.2779  89.27 KB
(91,408 bytes) 9/13/2002 6:09:34 PM
Microsoft Corporation
c:\winnt\system32\netman.dll
es.dll        2000.2.3471.1  222.27 KB (227,600
bytes) 9/13/2002 6:09:21 PM      Microsoft
Corporation
c:\winnt\system32\es.dll
ntmsdba.dll   5.00.2195.2779  167.27 KB
(171,280 bytes) 9/13/2002 6:09:35 PM
Microsoft Corporation
c:\winnt\system32\ntmsdba.dll
ntmssvc.dll   5.00.2195.2779  391.27 KB
(400,656 bytes) 9/13/2002 6:09:35 PM
Microsoft Corporation
c:\winnt\system32\ntmssvc.dll
mtxoci.dll    2000.2.3471.1  101.77 KB
(104,208 bytes) 9/13/2002 6:09:33 PM
Microsoft Corporation
c:\winnt\system32\mtxoci.dll
resutils.dll  5.00.2195.2787  39.77 KB
(40,720 bytes) 9/13/2002 6:09:40 PM
Microsoft Corporation
c:\winnt\system32\resutils.dll
clusapi.dll   5.00.2195.2104  54.27 KB
(55,568 bytes) 9/13/2002 6:09:16 PM
Microsoft Corporation
c:\winnt\system32\clusapi.dll
msvcp50.dll  5.00.7051 552.50 KB (565,760
bytes) 12/7/1999 7:00:00 AM      Microsoft
Corporation
c:\winnt\system32\msvcp50.dll
xolehlp.dll   1999.9.3421.3  17.27 KB
(17,680 bytes) 9/13/2002 5:45:08 PM
Microsoft Corporation
c:\winnt\system32\xolehlp.dll
msdtclog.dll  1999.9.3421.3  89.77 KB
(91,920 bytes) 9/13/2002 5:45:07 PM
Microsoft Corporation
c:\winnt\system32\msdtclog.dll
mtxclu.dll   2000.2.3471.1  51.27 KB
(52,496 bytes) 9/13/2002 6:09:33 PM
Microsoft Corporation
c:\winnt\system32\mtxclu.dll
msdtcprx.dll  2000.2.3471.1  665.77 KB
(681,744 bytes) 9/13/2002 6:09:27 PM
Microsoft Corporation
c:\winnt\system32\msdtcprx.dll
txfaux.dll   2000.2.3471.1  374.27 KB
(383,248 bytes) 9/13/2002 6:09:44 PM
Microsoft Corporation
c:\winnt\system32\txfaux.dll
msdtctm.dll  2000.2.3471.1  1.07 MB
(1,120,528 bytes) 9/13/2002 6:09:28 PM
Microsoft Corporation
c:\winnt\system32\msdtctm.dll
msdtc.exe    1999.9.3421.3  6.77 KB (6,928 bytes)
9/13/2002 5:45:07 PM      Microsoft
Corporation
c:\winnt\system32\msdtc.exe
rpcss.dll    5.00.2195.2815  231.27 KB (236,816
bytes) 9/13/2002 6:09:40 PM      Microsoft
Corporation
c:\winnt\system32\rpcss.dll
svchost.exe  5.00.2134.1      7.77 KB
(7,952 bytes) 12/7/1999 7:00:00 AM

```



```

Microsoft Corporation
rdpwsx.dll c:\winnt\system32\svchost.exe 94.40 KB
(96,664 bytes) 5.00.2180.1 9/13/2002 5:45:10 PM
Microsoft Corporation
ntlsapi.dll c:\winnt\system32\rdpwsx.dll 6.77 KB
(6,928 bytes) 5.00.2134.1 12/7/1999 7:00:00 AM
Microsoft Corporation
mstlsapi.dll c:\winnt\system32\ntlsapi.dll 24.77 KB
(25,360 bytes) 5.00.2181.1 12/7/1999 7:00:00 AM
Microsoft Corporation
icaapi.dll c:\winnt\system32\mstlsapi.dll 118.77 KB
(121,616 bytes) 5.00.2134.1 9/13/2002 5:45:09 PM
Microsoft Corporation
regapi.dll c:\winnt\system32\icaapi.dll 35.27 KB
(36,112 bytes) 5.00.2155.1 12/7/1999 7:00:00 AM
Microsoft Corporation
termsrv.exe c:\winnt\system32\regapi.dll 137.27 KB
(140,560 bytes) 5.00.2195.2342 9/13/2002 6:09:44 PM
Microsoft Corporation
scecli.dll c:\winnt\system32\termsrv.exe 105.27 KB
(107,792 bytes) 5.00.2195.2780 9/13/2002 6:09:41 PM
Microsoft Corporation
atl.dll c:\winnt\system32\scecli.dll 57.56 KB (58,938 bytes)
12/7/1999 7:00:00 AM Microsoft
Corporation c:\winnt\system32\atl.dll
certcli.dll 5.00.2195.2778 130.77 KB
(133,904 bytes) 9/13/2002 6:09:16 PM
Microsoft Corporation
esent.dll c:\winnt\system32\certcli.dll 1.08 MB (1,135,376
bytes) 6.0.3940.13 9/13/2002 6:09:21 PM
Microsoft
Corporation c:\winnt\system32\esent.dll
ntdsatq.dll 5.00.2195.2878 31.27 KB
(32,016 bytes) 9/13/2002 6:09:35 PM
Microsoft Corporation
ntdsa.dll c:\winnt\system32\ntdsatq.dll 990.77 KB (1,014,544
bytes) 5.00.2195.2899 9/13/2002 6:09:34 PM
Microsoft
Corporation c:\winnt\system32\ntdsa.dll
kdcsvc.dll 5.00.2195.2878 137.77 KB
(141,072 bytes) 9/13/2002 6:09:26 PM
Microsoft Corporation
sfmapi.dll c:\winnt\system32\kdcsvc.dll 38.77 KB
(39,696 bytes) 5.00.2134.1 12/7/1999 7:00:00 AM
Microsoft Corporation
rassfm.dll c:\winnt\system32\sfmapi.dll 21.27 KB
(21,776 bytes) 5.00.2195.2671 9/13/2002 6:09:39 PM
Microsoft Corporation
mpr.dll c:\winnt\system32\rassfm.dll 53.27 KB (54,544 bytes)
5.00.2195.2779 9/13/2002 6:09:27 PM
Microsoft
Corporation c:\winnt\system32\mpr.dll

```

```

rsabase.dll 5.00.2195.2228 128.27 KB
(131,344 bytes) 5/4/2001 12:05:02 PM
Microsoft Corporation
schannel.dll c:\winnt\system32\rsabase.dll 138.27 KB
(141,584 bytes) 5.00.2195.2922 5/4/2001 12:05:02 PM
Microsoft Corporation
netlogon.dll c:\winnt\system32\schannel.dll 357.77 KB
(366,352 bytes) 5.00.2195.2865 9/13/2002 6:09:34 PM
Microsoft Corporation
msvl_0.dll c:\winnt\system32\netlogon.dll 111.77 KB
(114,448 bytes) 5.00.2195.2900 12/7/1999 7:00:00 AM
Microsoft Corporation
kerberos.dll c:\winnt\system32\msvl_0.dll 198.77 KB
(203,536 bytes) 5.00.2195.2913 9/13/2002 6:09:26 PM
Microsoft Corporation
msprivs.dll c:\winnt\system32\kerberos.dll 41.50 KB
(42,496 bytes) 5.00.2154.1 12/7/1999 7:00:00 AM
Microsoft Corporation
samsvr.dll c:\winnt\system32\msprivs.dll 369.77 KB
(378,640 bytes) 5.00.2195.2918 12/7/1999 7:00:00 AM
Microsoft Corporation
lsasrv.dll c:\winnt\system32\samsvr.dll 492.77 KB
(504,592 bytes) 5.00.2195.2964 12/7/1999 7:00:00 AM
Microsoft Corporation
lsass.exe c:\winnt\system32\lsasrv.dll 32.77 KB (33,552 bytes)
12/7/1999 7:00:00 AM Microsoft
Corporation c:\winnt\system32\lsass.exe
wmicore.dll 5.00.2195.2842 72.27 KB
(74,000 bytes) 9/13/2002 6:09:46 PM
Microsoft Corporation
mswsock.dll c:\winnt\system32\wmicore.dll 62.77 KB
(64,272 bytes) 5.00.2195.2871 9/13/2002 6:09:33 PM
Microsoft Corporation
msgsvc.dll c:\winnt\system32\mswsock.dll 34.27 KB
(35,088 bytes) 5.00.2195.2939 12/7/1999 7:00:00 AM
Microsoft Corporation
browser.dll c:\winnt\system32\msgsvc.dll 48.27 KB
(49,424 bytes) 5.00.2195.2778 9/13/2002 6:09:14 PM
Microsoft Corporation
psbase.dll c:\winnt\system32\browser.dll 111.77 KB
(114,448 bytes) 5.00.2195.2779 9/13/2002 6:09:39 PM
Microsoft Corporation
trkwws.dll c:\winnt\system32\psbase.dll 88.77 KB
(90,896 bytes) 5.00.2166.1 12/7/1999 7:00:00 AM
Microsoft Corporation
cryptsvc.dll c:\winnt\system32\trkwws.dll 61.77 KB
(63,248 bytes) 5.00.2181.1 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\cryptsvc.dll

```

```

cryptdll.dll 5.00.2135.1 41.27 KB
(42,256 bytes) 12/7/1999 7:00:00 AM
Microsoft Corporation
wkssvc.dll c:\winnt\system32\cryptdll.dll 95.27 KB
(97,552 bytes) 5.00.2195.2780 12/7/1999 7:00:00 AM
Microsoft Corporation
srvsvc.dll c:\winnt\system32\wkssvc.dll 79.27 KB
(81,168 bytes) 5.00.2195.2904 12/7/1999 7:00:00 AM
Microsoft Corporation
cfgmgr32.dll c:\winnt\system32\srvsvc.dll 16.77 KB
(17,168 bytes) 5.00.2134.1 12/7/1999 7:00:00 AM
Microsoft Corporation
dmsrvr.dll c:\winnt\system32\cfgmgr32.dll 11.77 KB
(12,048 bytes) 2195.2778.297.3 9/13/2002 6:09:19 PM
VERITAS Software Corp.
lmhsvc.dll c:\winnt\system32\dmsrvr.dll 9.77 KB
(10,000 bytes) 5.00.2195.2778 12/7/1999 7:00:00 AM
Microsoft Corporation
eventlog.dll c:\winnt\system32\lmhsvc.dll 43.77 KB
(44,816 bytes) 5.00.2178.1 12/7/1999 7:00:00 AM
Microsoft Corporation
ntdsapi.dll c:\winnt\system32\eventlog.dll 55.77 KB
(57,104 bytes) 5.00.2195.2661 9/13/2002 6:09:35 PM
Microsoft Corporation
scesrv.dll c:\winnt\system32\ntdsapi.dll 226.27 KB
(231,696 bytes) 5.00.2195.2780 9/13/2002 6:09:41 PM
Microsoft Corporation
umpnpgm.dll c:\winnt\system32\scesrv.dll 86.27 KB
(88,336 bytes) 5.00.2182.1 12/7/1999 7:00:00 AM
Microsoft Corporation
services.exe c:\winnt\system32\umpnpgm.dll 86.77 KB
(88,848 bytes) 5.00.2195.2780 12/7/1999 7:00:00 AM
Microsoft Corporation
rasadhlp.dll c:\winnt\system32\services.exe 7.27 KB
(7,440 bytes) 5.00.2168.1 12/7/1999 7:00:00 AM
Microsoft Corporation
winrnr.dll c:\winnt\system32\rasadhlp.dll 18.77 KB
(19,216 bytes) 5.00.2160.1 12/7/1999 7:00:00 AM
Microsoft Corporation
rnr20.dll c:\winnt\system32\winrnr.dll 35.77 KB (36,624 bytes)
5.00.2195.2871 9/13/2002 6:09:40 PM
Microsoft
Corporation c:\winnt\system32\rnr20.dll
clbcatq.dll 2000.2.3471.1 496.77 KB
(508,688 bytes) 5.00.2195.2778 9/13/2002 6:09:16 PM
Microsoft Corporation
dhcpcsvc.dll c:\winnt\system32\clbcatq.dll 88.77 KB
(90,896 bytes) 5.00.2195.2778 12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\dhcpcsvc.dll

```

tapi32.dll 5.00.2182.1 123.27 KB
 (126,224 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\tapi32.dll
 rasman.dll 5.00.2195.2780 54.77 KB
 (56,080 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\rasman.dll
 rasapi32.dll 5.00.2195.2671 189.77 KB
 (194,320 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\rasapi32.dll
 rtutils.dll 5.00.2168.1 43.77 KB
 (44,816 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\rtutils.dll
 adslrpc.dll 5.00.2195.2842 127.27 KB
 (130,320 bytes) 9/13/2002 6:09:12 PM
 Microsoft Corporation
 c:\winnt\system32\adslrpc.dll
 activeds.dll 5.00.2195.2778 174.77 KB
 (178,960 bytes) 9/13/2002 6:09:09 PM
 Microsoft Corporation
 c:\winnt\system32\activeds.dll
 oleaut32.dll 2.40.4517 612.27 KB (626,960
 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\oleaut32.dll
 mprapi.dll 5.00.2181.1 79.27 KB
 (81,168 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\mprapi.dll
 icmp.dll 5.00.2134.1 7.27 KB (7,440 bytes)
 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\icmp.dll
 iphlapi.dll 5.00.2173.2 67.77 KB
 (69,392 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\iphlpapi.dll
 wshtcpip.dll 5.00.2195.2104 17.27 KB
 (17,680 bytes) 9/13/2002 6:09:46 PM
 Microsoft Corporation
 c:\winnt\system32\wshtcpip.dll
 msafd.dll 5.00.2195.2779 106.77 KB (109,328
 bytes) 9/13/2002 6:09:27 PM
 Microsoft Corporation
 c:\winnt\system32\msafd.dll
 winspool.drv 5.00.2195.2780 109.77 KB
 (112,400 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\winspool.drv
 winscard.dll 5.00.2134.1 77.27 KB
 (79,120 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\winscard.dll
 wlnotify.dll 5.00.2195.2780 53.77 KB
 (55,056 bytes) 9/13/2002 6:09:46 PM
 Microsoft Corporation
 c:\winnt\system32\wlnotify.dll
 csd.dll 5.00.2195.2401 98.27 KB
 (100,624 bytes) 9/13/2002 6:09:17 PM
 Microsoft Corporation
 c:\winnt\system32\csd.dll

lz32.dll 5.00.2134.1 9.77 KB (10,000 bytes)
 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\lz32.dll
 version.dll 5.00.2134.1 15.77 KB
 (16,144 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\version.dll
 rsaenh.dll 5.00.2195.2228 130.77 KB
 (133,904 bytes) 9/13/2002 6:10:37 PM
 Microsoft Corporation
 c:\winnt\system32\rsaenh.dll
 mscat32.dll 5.131.2134.1 7.77 KB
 (7,952 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\mscat32.dll
 ole32.dll 5.00.2195.2887 969.77 KB (993,040
 bytes) 9/13/2002 6:09:38 PM
 Microsoft Corporation
 c:\winnt\system32\ole32.dll
 imagehlp.dll 5.00.2195.2778 125.77 KB
 (128,784 bytes) 5/4/2001 12:05:02 PM
 Microsoft Corporation
 c:\winnt\system32\imagehlp.dll
 msasn1.dll 5.00.2134.1 51.27 KB
 (52,496 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\msasn1.dll
 crypt32.dll 5.131.2195.2833 451.27 KB
 (462,096 bytes) 9/13/2002 6:09:17 PM
 Microsoft Corporation
 c:\winnt\system32\crypt32.dll
 wintrust.dll 5.131.2195.2779 162.27 KB
 (166,160 bytes) 9/13/2002 6:09:46 PM
 Microsoft Corporation
 c:\winnt\system32\wintrust.dll
 shlwapi.dll 5.00.3315.1000 282.77 KB
 (289,552 bytes) 9/13/2002 6:09:42 PM
 Microsoft Corporation
 c:\winnt\system32\shlwapi.dll
 shell32.dll 5.00.3315.2902 2.25 MB
 (2,359,056 bytes) 9/13/2002 6:09:42 PM
 Microsoft Corporation
 c:\winnt\system32\shell32.dll
 msgina.dll 5.00.2195.2779 324.27 KB
 (332,048 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\msgina.dll
 comctl32.dll 5.81 537.77 KB (550,672
 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\comctl32.dll
 setupapi.dll 5.00.2195.2663 555.77 KB
 (569,104 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\setupapi.dll
 winmm.dll 5.00.2161.1 184.77 KB (189,200
 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\winmm.dll
 winsta.dll 5.00.2195.2386 36.77 KB
 (37,648 bytes) 9/13/2002 6:09:46 PM
 Microsoft Corporation
 c:\winnt\system32\winsta.dll
 wsock32.dll 5.00.2195.2871 21.27 KB
 (21,776 bytes) 9/13/2002 6:09:46 PM

Microsoft Corporation
 c:\winnt\system32\wsock32.dll
 dnsapi.dll 5.00.2195.2785 130.77 KB
 (133,904 bytes) 9/13/2002 6:09:19 PM
 Microsoft Corporation
 c:\winnt\system32\dnsapi.dll
 wldap32.dll 5.00.2195.2797 125.27 KB
 (128,272 bytes) 9/13/2002 6:09:46 PM
 Microsoft Corporation
 c:\winnt\system32\wldap32.dll
 ws2help.dll 5.00.2134.1 17.77 KB
 (18,192 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\ws2help.dll
 ws2_32.dll 5.00.2195.2780 67.77 KB
 (69,392 bytes) 9/13/2002 6:09:46 PM
 Microsoft Corporation
 c:\winnt\system32\ws2_32.dll
 samlib.dll 5.00.2195.2780 49.77 KB
 (50,960 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\samlib.dll
 netrap.dll 5.00.2134.1 11.27 KB
 (11,536 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\netrap.dll
 netapi32.dll 5.00.2195.2808 303.77 KB
 (311,056 bytes) 9/13/2002 6:09:34 PM
 Microsoft Corporation
 c:\winnt\system32\netapi32.dll
 profmap.dll 5.00.2181.1 29.27 KB
 (29,968 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\profmap.dll
 secur32.dll 5.00.2195.2862 46.77 KB
 (47,888 bytes) 9/13/2002 6:09:41 PM
 Microsoft Corporation
 c:\winnt\system32\secur32.dll
 sfc.dll 5.00.2195.2896 92.11 KB (94,320 bytes)
 9/13/2002 6:09:41 PM
 Microsoft Corporation
 c:\winnt\system32\sfc.dll
 nddeapi.dll 5.00.2137.1 15.27 KB
 (15,632 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\nddeapi.dll
 userenv.dll 5.00.2195.2780 361.77 KB
 (370,448 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\userenv.dll
 user32.dll 5.00.2195.2821 392.77 KB
 (402,192 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\user32.dll
 gdi32.dll 5.00.2195.2778 228.77 KB (234,256
 bytes) 12/7/1999 7:00:00 AM
 Microsoft Corporation
 c:\winnt\system32\gdi32.dll
 rpcrt4.dll 5.00.2195.2832 437.27 KB
 (447,760 bytes) 9/13/2002 6:09:40 PM
 Microsoft Corporation
 c:\winnt\system32\rpcrt4.dll
 advapi32.dll 5.00.2195.2867 351.77 KB
 (360,208 bytes) 12/7/1999 7:00:00 AM

```

Microsoft Corporation
c:\winnt\system32\advapi32.dll
kernel32.dll      5.00.2195.2778    714.77 KB
(731,920 bytes)  12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\kernel32.dll
msvcrt.dll       6.10.8924.0      284.05 KB
(290,869 bytes)  5/4/2001 12:05:02 PM
Microsoft Corporation
c:\winnt\system32\msvcrt.dll
winlogon.exe     5.00.2195.2953   173.77 KB
(177,936 bytes)  12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\winlogon.exe
sfcfiles.dll    5.00.2195.2967   948.27 KB
(971,024 bytes)  9/13/2002 6:09:41 PM
Microsoft Corporation
c:\winnt\system32\sfcfiles.dll
ntdll.dll       5.00.2195.2779   478.77 KB (490,256
bytes)           5/4/2001 12:05:02 PM
Microsoft Corporation
c:\winnt\system32\ntdll.dll
smss.exe        5.00.2195.2901   44.27 KB (45,328 bytes)
12/7/1999 7:00:00 AM
Microsoft Corporation
c:\winnt\system32\smss.exe

```

[Services]

Display Name	Name	State	Start Mode
Service Type	Path	Error Control	
Start Name	Tag ID		
Altiris Client Service	AClient	Stopped	
Auto	Own Process		c:\program
files\altiris\aclient\aclient.exe	-service		
Normal	LocalSystem	0	
Alerter	Alerter	Stopped	Auto
Share Process	c:\winnt\system32\services.exe		
Normal	LocalSystem	0	
Application Management	AppMgmt	Stopped	
Manual	Share Process		
c:\winnt\system32\services.exe			
Normal	LocalSystem	0	
Computer Browser	Browser	Running	Auto
Share Process			
c:\winnt\system32\services.exe			
Normal	LocalSystem	0	
Indexing Service	cisvc	Stopped	Manual
Share Process			
c:\winnt\system32\cisvc.exe			Normal
LocalSystem	0		
ClipBook	ClipSrv	Stopped	Manual
Own Process			
c:\winnt\system32\clipsrv.exe			Normal
LocalSystem	0		
Distributed File System	Dfs	Stopped	
Auto	Own Process		
c:\winnt\system32\dfssvc.exe			Normal
LocalSystem	0		
DHCP Client	Dhcp	Stopped	Auto
Share Process			
c:\winnt\system32\services.exe			
Normal	LocalSystem	0	
Logical Disk Manager Administrative Service			
dmadmin	Stopped	Manual	Share Process

```

c:\winnt\system32\dmadmin.exe /com
Normal LocalSystem 0
Logical Disk Manager dmserver Running
Auto Share Process
c:\winnt\system32\services.exe
Normal LocalSystem 0
DNS Client Dnscache Stopped Auto
Share Process
c:\winnt\system32\services.exe
Normal LocalSystem 0
Event Log Eventlog Running Auto Share Process
c:\winnt\system32\services.exe
Normal LocalSystem 0
COM+ Event System EventSystem Running
Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Fax Service Fax Stopped Manual Own
Process c:\winnt\system32\faxsvc.exe Normal
LocalSystem 0
IIS Admin Service IISADMIN Running Auto
Share Process
c:\winnt\system32\inetrv\inetinfo.exe
Normal LocalSystem 0
Intersite Messaging IsmServ Stopped Disabled Own
Process c:\winnt\system32\ismserv.exe Normal
LocalSystem 0
Kerberos Key Distribution Center kdc
Stopped Disabled Share Process
c:\winnt\system32\lsass.exe Normal
LocalSystem 0
Server lanmanserver Running Auto
Share Process
c:\winnt\system32\services.exe
Normal LocalSystem 0
Workstation lanmanworkstation Running
Auto Share Process
c:\winnt\system32\services.exe
Normal LocalSystem 0
License Logging Service LicenseService
Stopped Auto Own Process
c:\winnt\system32\llssrv.exe Normal
LocalSystem 0
TCP/IP NetBIOS Helper Service LmHosts Running
Auto Share Process
c:\winnt\system32\services.exe
Normal LocalSystem 0
Messenger Messenger Running Auto Share Process
c:\winnt\system32\services.exe
Normal LocalSystem 0
NetMeeting Remote Desktop Sharing mnmsrvc
Stopped Manual Own Process
c:\winnt\system32\mnmsrvc.exe Normal
LocalSystem 0
Distributed Transaction Coordinator MSDTC
Running Auto Own Process
c:\winnt\system32\msdtc.exe Normal
LocalSystem 0
Windows Installer MSIServer Stopped Manual
Share Process
c:\winnt\system32\msiexec.exe /v
Normal LocalSystem 0

```

```

Network DDE NetDDE Stopped Manual
Share Process
c:\winnt\system32\netdde.exe Normal
LocalSystem 0
Network DDE DSDM NetDDEdsdm Stopped
Manual Share Process
c:\winnt\system32\netdde.exe Normal
LocalSystem 0
Net Logon Netlogon Stopped Manual Share Process
c:\winnt\system32\lsass.exe Normal
LocalSystem 0
Network Connections Netman Running Manual
Share Process
c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
File Replication NtFrs Stopped Manual Own
Process c:\winnt\system32\ntfrs.exe Ignore
LocalSystem 0
NT LM Security Support Provider NtLmSsp
Stopped Manual Share Process
c:\winnt\system32\lsass.exe Normal
LocalSystem 0
Removable Storage NtmsSvc Running Auto
Share Process
c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Plug and Play PlugPlay Running Auto
Share Process
c:\winnt\system32\services.exe
Normal LocalSystem 0
IPSEC Policy Agent PolicyAgent Stopped
Auto Share Process
c:\winnt\system32\lsass.exe Normal
LocalSystem 0
Protected Storage ProtectedStorage Running
Auto Share Process
c:\winnt\system32\services.exe
Normal LocalSystem 0
Remote Access Auto Connection Manager RasAuto
Stopped Manual Share Process
c:\winnt\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Remote Access Connection Manager RasMan
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 Command Service RMSYS Running
Auto Own Process
c:\benchcraft\rsys.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
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
Terminal Services TermService Running
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
Running Auto Share Process
c:\winnt\system32\services.exe
Normal LocalSystem 0
Uninterruptible Power Supply UPS Stopped
Manual Own Process

```

```

c:\winnt\system32\ups.exe Normal
LocalSystem 0
Utility Manager UtilMan Stopped Manual Own
Process c:\winnt\system32\utilman.exe Normal
LocalSystem 0
Windows Time W32Time Stopped Manual
Share Process
c:\winnt\system32\services.exe
Normal LocalSystem 0
World Wide Web Publishing Service W3SVC
Running Auto Share Process
c:\winnt\system32\inetrv\inetinfo.exe
Normal LocalSystem 0
webd webd Stopped Auto Own Process
c:\inetpub\wwwroot\webd.exe http/1.1
Normal LocalSystem 0
Windows Management Instrumentation WinMgmt
Running Auto Own Process
c:\winnt\system32\wbem\winmgmt.exe
Ignore LocalSystem 0
Windows Management Instrumentation Driver Extensions
Wmi Running Manual Share Process
c:\winnt\system32\services.exe
Normal LocalSystem 0

[Program Groups]

Group Name Name User Name
Accessories Default User:Accessories
Default User
Accessories\Accessibility Default
User:Accessories\Accessibility Default User
Accessories\Entertainment Default
User:Accessories\Entertainment Default User
Accessories\System Tools Default
User:Accessories\System Tools Default User
Startup Default User:Startup Default User
Accessories All Users:Accessories All
Users
Accessories\Communications All
Users:Accessories\Communications All Users
Accessories\Entertainment All
Users:Accessories\Entertainment All
Users:Accessories\Microsoft Script Debugger All
Users:Accessories\Microsoft Script Debugger All
Users
Accessories\System Tools All
Users:Accessories\System Tools All Users
Administrative Tools All
Users:Administrative Tools All Users
Microsoft SQL Server All Users:Microsoft SQL
Server All Users
Startup All Users:Startup All Users
Tardis All Users:Tardis All Users
Accessories CL8\Administrator:Accessories
Accessories\Accessibility
CL8\Administrator:Accessories\Accessibility
CL8\Administrator
Accessories\Entertainment
CL8\Administrator:Accessories\Entertainment
CL8\Administrator

```

```

Accessories\System Tools
CL8\Administrator:Accessories\System Tools
CL8\Administrator
Administrative Tools
CL8\Administrator:Administrative Tools
CL8\Administrator
Benchcraft CL8\Administrator:Benchcraft
CL8\Administrator
Startup CL8\Administrator:Startup
CL8\Administrator

[Startup Programs]

Program Command User Name Location
Tardis 2000 c:\progra-1\tardis-1.4\tardis.exe
CL8\Administrator Startup
ACIntUsr c:\program
files\altiris\aclient\aclntusr.exe All Users
HKLM\SOFTWARE\Microsoft\Windows\CurrentVers
ion\Run

[OLE Registration]

Object Local Server
Sound (OLE2) sndrec32.exe
Media Clip mplay32.exe
Video Clip mplay32.exe /avi
MIDI Sequence mplay32.exe /mid
Sound Not Available
Media Clip Not Available
Image Document "C:\Program Files\Windows
NT\Accessories\ImageVue\KodakImg.exe"
WordPad Document "%ProgramFiles%\Windows
NT\Accessories\WORDPAD.EXE"
Windows Media Services DRM Storage object Not
Available
Bitmap Image mspaint.exe

[Internet Explorer 5]

[ Following are sub-categories of this main category
]

[Summary]

Item Value
Version 5.00.3315.1000
Build 53315.1000
Product ID 51876-270-9567332-05753
Application Path C:\Program Files\Internet
Explorer
Language English (United States)
Active Printer Not Available

Cipher Strength 168-bit
Content Advisor Disabled
IEAK Install No

[File Versions]

File Version Size Date Path
Company

```

```

advapi32.dll      5.0.2195.2867      352 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
advpack.dll      5.0.3103.1000     87 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
browsec.dll      5.0.3315.2846     35 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
browseui.dll     5.0.3315.2846     789 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
ckcnv.exe        5.0.2189.1         9 KB
12/7/1999
7:00:00 AM      C:\WINNT\system32 Microsoft
Corporation
comctl32.dll     5.81.3103.1000    538 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
crypt32.dll     5.131.2195.2833   451 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
ehnsig.dll       <File Missing>    Not Available
Not Available   Not Available   Not Available
Available
iemigrat.dll     <File Missing>    Not Available
Not Available   Not Available   Not Available
Available
iesetup.dll     5.0.3103.1000     57 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
iexplore.exe    5.0.2920.0        59 KB
12/7/1999 7:00:00 AM C:\Program
Files\Internet Explorer Microsoft Corporation
imagehlp.dll    5.0.2195.2778    126 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
imghelp.dll     <File Missing>    Not Available
Not Available   Not Available   Not Available
Available
inseng.dll      5.0.3103.1000     72 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
jobexec.dll     5.0.0.1           47 KB
12/7/1999
7:00:00 AM      C:\WINNT\system32 Microsoft
Corporation
jscript.dll     5.1.0.5907        476 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
jsproxy.dll     5.0.2920.0        13 KB
12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation
msaahtml.dll    <File Missing>    Not Available
Not Available   Not Available   Not Available
Available
mshtml.dll      5.0.3315.2870     2290 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
msjava.dll     5.0.3802.0        923 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
msoss.dll       <File Missing>    Not Available
Available Not Available   Not Available

```

```

msxml.dll 8.0.5718.1      493 KB 5/4/2001
11:05:02 AM C:\WINNT\system32 Microsoft
Corporation
occache.dll 5.0.3103.1000      86 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
ole32.dll 5.0.2195.2887    970 KB 5/4/2001
11:05:02 AM C:\WINNT\system32 Microsoft
Corporation
oleaut32.dll 2.40.4517.0        612 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
olepro32.dll 5.0.4517.0         160 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
rsabase.dll 5.0.2195.2228     128 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
rsaenh.dll 5.0.2195.2228     131 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
rsapi32.dll <File Missing> Not Available
Not Available   Not Available   Not Available
Available
rsasig.dll <File Missing> Not Available
Not Available   Not Available   Not Available
Available
schannel.dll 5.1.2195.0         138 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
shdoc401.dll <File Missing> Not Available
Not Available   Not Available   Not Available
Available
shdocv.dll 5.0.3315.2879     1078 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
shell32.dll 5.0.3315.2902     2304 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
shlwapi.dll 5.0.3315.1000     283 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
url.dll 5.0.2920.0     82 KB 12/7/1999
7:00:00 AM C:\WINNT\system32 Microsoft
Corporation
urlmon.dll 5.0.3315.1000    441 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
vbscript.dll 5.1.0.5907        428 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
webcheck.dll 5.0.3315.1000     252 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
win.com 5.0.2134.1        24 KB 12/7/1999
7:00:00 AM C:\WINNT\system32 Microsoft
Corporation
wininet.dll 5.0.3315.1000    457 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
winsock.dll 3.10.0.103         3 KB
12/7/1999 7:00:00 AM
C:\WINNT\system32 Microsoft Corporation

```

```

wintrust.dll 5.131.2195.2779   162 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
wsock.vxd <File Missing> Not Available
Available Not Available Not Available
wsock32.dll 5.0.2195.2871     21 KB
5/4/2001 11:05:02 AM
C:\WINNT\system32 Microsoft Corporation
wsock32n.dll <File Missing> Not Available
Available Not Available Not Available
Not Available

[Connectivity]

Item Value
Connection Preference Never dial
EnableHttp1.1 1
ProxyHttp1.1 0

LAN Settings

AutoConfigProxy wininet.dll
AutoProxyDetectMode Disabled
AutoConfigURL
Proxy Disabled
ProxyServer
ProxyOverride

[Cache]

[ Following are sub-categories of this main category ]

[Summary]

Item Value
Page Refresh Type Automatic
Temporary Internet Files Folder C:\Documents
and Settings\Administrator\Local Settings\Temporary
Internet Files
Total Disk Space 17355 MB
Available Disk Space 14419 MB
Maximum Cache Size 542 MB
Available Cache Size 542 MB

[List of Objects]

Program File Status CodeBase
No cached object information available

[Content]

[ Following are sub-categories of this main category ]

[Summary]

Item Value
Content Advisor Disabled

[Personal Certificates]

Issued To Issued By Validity Signature Algorithm

```

Administrator Administrator 9/13/2002 to
8/20/2102 sha1RSA

[Other People Certificates]

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

[Publishers]

Name
No publisher information available

[Security]

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

Microsoft SQL Server 2000 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 SQL Collation binary

sort order/Latin_1_General

Microsoft COM Component Configuration Parameters

The component services tool in Windows 2003 Server
was used to change the queue settings for the
TPCC COM+ single queue component. The
single queue component was 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

single queue component on the client was 200.
Delivery threads were set under the TPCC key
in the registry at 66. The construction string was
Dummy String

Appendix D: 60-Day Space

TPC-C 60 Day Space Requirements						
Warehouses	4,800				TpmC	60,364.00
Table	Rows	Data KB	Index KB	Extra 5% KB	8hr Space	Total Space KB
Warehouse	4,800	512	16	26		554
District	48,000	5,336	16	268		5620
Customer	144,000,000	104,727,280	6,244,648	5,548,596		116520524
History	144,000,000	8,000,008	16		1,619,786	8000024
New_order	43,200,000	683,008	1,568	34,229		718805
Orders	144,000,000	4,413,800	2,007,056		6,854,674	6420856
Order_line	1,439,997,837	89,999,872	190,496		19,964,852	90190368
Item	100,000	9,528	32	478		10038
Stock	480,000,000	153,600,000	286,936	7,694,347		161581283
Total		361,439,344	8,730,784	13,277,944	28,439,312	383,448,072
MB						
Dynamic Space	100,013	Sum of Data for Order, Orderline and History				
Static Space	274,448	Sum of Data+Index+5%-Dynamic Space				
Free Space	na	Total Allocated Spac - (Dynamic + Static Space)				
Daily Growth	20,124	(Dynamic Space/(W*62.5))*tpmc				
Daily Spread	-	(Free Space -1.5*Daily Growth) Zero Assumed				
60 Day Space MB	1,481,889					
60 Day Space GB	1,447.16					
Log Size	119,999.99					
KB Per New Order	4.76					
8 hr log MB	134,650					
8 hr log GB	131.4937					
Space Usage	GB Needed	Disks Measured	GB Priced	Disk Size	Formatted Size	
60 Day Space DB	1,447.16	280	4732.00	18.2GB	16.900	
Total DB			4732.00			
8-hr log + mirror	262.9874	10	339.20	36.4GB	33.92	
OS, Swap	3	1	16.90	18.2GB	16.900	
Total Storage	1,713.14	GB	5,088.10	GB		

	MSSQL_misc_fg	MSSQL_cs_fg
	554	
	5620	
		116520524
	9619810	
	718805	
	13275530	
	110155220	
	10038	
		161581283
	133,785,576	278,101,807
files=	1	4
size=	19,187,200	9,984,000
Total=	19,187,200	39,936,000
8K blocks	153,497,600	319,488,000
	OK	OK

tpmC	60,364.00									
	Data Before KB	Index Before KB	Data After KB	Index After KB	Data Grow KB	Index Grow KB	Total Grow KB	KB/New-Order	8-Hr Growth KB	8-Hr Growth MB
History	8,000,008	16	8,747,808	80	747,800	64	747,864	0.0559	1,619,786.09	1,581.82
Order	4,413,800	2,007,056	5,558,608	4,027,088	1,144,808	2,020,032	3,164,840	0.2366	6,854,673.84	6,694.02
Order-Line	89,999,872	190,496	99,027,296	380,952	9,027,424	190,456	9,217,880	0.6890	19,964,851.58	19,496.93
										27,772.77
	sum(*) Before		sum(*) After		Num New-					
d_next_o_id	144,048,000		157,425,785		13,377,785					
	Before MB		After MB		Grow MB					
Log	1847.78		64016.20		62168.42			KB/New-Order	8-Hr Growth MB	8-Hr Growth GB
								4.7587	134,649.53	131.49
								4,872.8777	bytes	
	119999.99	1.5398178	53.346836							
Database tpcc log used (%)										

Appendix E:

Third Party Letters



Phone Orders
800.287.2323

**THE NUMBER ONE SOURCE
FOR COMPUTER PERIPHERALS**

Home About Us Product Return Customer Service Contact Us Phone Orders

IMPORTANT:
Closed for
Holiday
Oct. 10-Oct. 17
Click Here

DEAL! HP Compaq H2215 / IPAQ Pocket PC 2003...\$354
New! Sandisk 256MB Ultra II Compact Flash Card - SDCFH256901...\$56
WOW! TDK AID+420 DVD-RW Retail Box ONLY! \$135
NEW! Sony Color CLIE PEG-UX50/U Handheld...\$590

shop by product

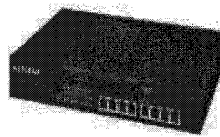
shop by brand

SEARCH STORE

GO!

- PRODUCTS**
- CDROMs
 - CDR/CDRW
 - Camera Accessories
 - CASES
 - CLOSEOUTS
 - CONTROLLER CARDS
 - CPU's
 - DIGITAL CAMERAS
 - DVDs
 - FLASH CARDS
 - FLOPPY DRIVES
 - GPS
 - HANDHELDS/PDA's
 - HARD DRIVES
 - HUBS
 - KEYBOARD/MOUSE
 - MEMORY
 - MEMORY STICK
 - MICROSOFT
 - MODEMS
 - MONITORS
 - MOTHERBOARDS
 - MP3 PLAYERS
 - NETWORKING
 - OPERATING SYSTEMS
 - POWER SUPPLIES
 - PRINTERS
 - REMOVABLE DRIVES
 - ROUTERS
 - SCANNERS
 - SECURE DIGITAL CARDS
 - SMART MEDIA
 - SOFTWARE
 - SOUND CARDS
 - SPEAKERS

NETGEAR NETGEAR GS508TNA 8 PORT GIGABIT COPPER SWITCH 10/100/1000 MBPS



- Price: \$485.00
- In Stock!

tell a friend
ADD TO CART
BUY NOW

DESCRIPTION:

The NETGEAR GS508T Gigabit over Copper Switch is a high performance network switch that provides back-bone connectivity for power workgroups, data centers, and server farms. [More Info & Product Specification](#)

More Info & Product Specification

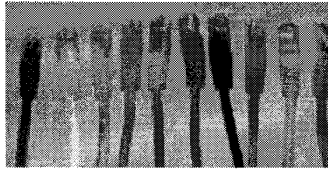
- Includes switch, power cord, rack-mount kit, and manual;
- CONNECTOR(s): (8) 10BaseT/100BaseTX/1000BaseT/RJ45 ports;
- INDICATORS: Unit, power, Per network port, link, activity, full duplex/collision;
- PERFORMANCE: Switching fabric (9.6 gigabit per sec), Forward rate (100 Mbps port) 148,000 packet per sec, Forward rate (1000 Mbps port) 1,480,000 packet per sec, Latency (100 to 1000 Mbps) 8 usec max;
- MAC addresses: 8,000;
- Gigabit buffer memory: 8MB for 8 ports;
- APPROVALS: CE, FCC A, EN55022 A,VCCI A,UL,TUV;
- POWER: Autosensing internal 100 ~ 240V, 50/60Hz; Consumption 25 watts;
- SIZE: 13.0"w x 1.7"h x 8.2"d;
- Five Year Warranty!
- Pictures are for illustration purposes only.

FEATURE	
<input type="checkbox"/> MXP Hc OEM	\$78.00
<input type="checkbox"/> PTEAC Read Guar	\$6.95
<input type="checkbox"/> E1250I Scanr	\$29.95

- Home
- Network Cards
- Network Cables & MISC Cat5e
- Crossover Cables
- Print Servers
- Barcode Readers
- Extension Cables
- Miscellaneous
- TEST
- WE ARE ANTI SPAM
- Blacklisted Brands
- gaming
- Cables - Misc
- SCSI Cables & devices
- Boneyard Cables
- 6ft 4 wire black molded
As low as 34 cents each
- network patch cable
- supports 10 / 100 mbps networks
Order quantities over 5 ONLY

- Show Order
- Privacy Policy
- Info &
Shipping Notes
& Ways to delay
Processing of order
- Search
- Index
- Y? SHOPPING**

LanAdapters.com



7ft Cat 5e Network Patch Cables. (compatible with cat 5)

7ft Category 5 and (Cat5e)Enhanced Network patch cables MOLDED. 35C Twist Pair supports fast ethernet. These cat 5 e cables are backwards compatible with cat 5

green purple come with booted snagless ends
blue light gray white black red yellow orange come with compact m snagless

Availability: Usually ships the same business day.

CBLC57 \$1.00, 125/\$118.75 Color:



click here to try our
NEW & IMPROVED
Navigation

**CHECK OUT THIS WEEK'S
SUPERIOR SAVINGS!** >>

- PC Computers
- Apple Computers
- Notebooks & Tablets
- PDA & Handhelds
- Hard Drives & Memory
- Electronic Com
- Networking
- Software
- Games
- DVD Movies
- Audio - Video
- Cameras & Telescopes
- Appliances
- Science & Toys

Customer Service | Home | Express Account | Your Privacy | Track Your Order | Returns | Warranty | 1.877.OUTPOST

Search in

[view your cart](#) & checkout



AirLink+ Mini Desktop KVM Switch PS/2

Airlink+:

Outpost #: 3891817

In stock, same day shipping



You can't go wrong!

- Best customer service on the web
- Secure, no-hassle shopping
- Our customers love us!

[Detailed Description](#) | [Tell a friend](#) | [Warranty Info](#)

Price: **\$ 29.99**

[add to cart](#) & buy it!

Detailed Description [\[return to top\]](#)

Need help placing your order? Let us help!

1-877-OUTPOST
(1-877-688-7678)
INTL: 1-860-927-2050

[add to cart](#) & buy it!

[Home](#) | [Site Map](#) | [Buying Guides](#)

Search

Copyright © 1995-2004 Fry's Electronics, Incorporated. All rights reserved.
Fry's Electronics and Outpost.com are registered trademarks of Fry's Electronics, Incorporated.

February 26, 2004

Hewlett-Packard
Company
Paul Cao
MS150402
20555 SH 249
Houston, TX 77070

Mr. Cao:

Here is the information you requested regarding pricing for several Microsoft products to be used in conjunction with your TPC-C benchmark testing.

All pricing shown is in US Dollars (\$).

Part Number	Description	Unit Price	Quantity	Price
810-00845	SQL Server 2000 Enterprise Edition <i>Per processor licensing</i> <i>Discount Schedule: Open Program Level B</i> <i>Unit Price reflects a 14% discount from the retail unit price of \$19,999.</i>	\$17,279	2	\$34,558
C11-00821	Windows 2000 Server <i>Server license only - No CALs</i> <i>Discount Schedule: Open Program - No Level</i> <i>Unit Price reflects a 8% discount from the retail unit price of \$799.</i>	\$738	1	\$738
P72-00264	Windows Server 2003, Enterprise Edition <i>Server license only - No CALs</i> <i>Discount Schedule: Open Program - No Level</i> <i>Unit Price reflects a 40% discount from the retail unit price of \$3,999.</i>	\$2,399	1	\$2,399
254-00170	Visual C++ Standard <i>No discounts applied</i>	\$109	1	\$109
PRO-PRORS-16U-01	Database Server Support Package <i>1 Year Term</i>	\$1,950	3	\$5,850

All products are currently orderable through Microsoft's normal distribution channels.

This quote is valid for the next 90 days.

If we can be of any further assistance, please contact Jamie Reding at (425) 703-0510 or jamiere@microsoft.com.