Compaq Computer Corporation

TPC Benchmark™ C
Full Disclosure Report
for
ProLiant 7000-6/500-2S
using
Microsoft SQL Server 7.0 Enterprise Edition
and
Windows NT 4.0 Enterprise Edition

First Edition
June 1999

COMPAQ

First Edition – June 1999

Compaq Computer Corporation (Compaq) believes that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. Compaq assumes no responsibility for any errors that may appear in this document. The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, Compaq provides no warranty of the pricing information in this document.

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

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

Printed in U.S.A., 1999
Compaq, NonStop, ProLiant 7000, and ProLiant are registered trademarks of Compaq Computer Corporation.

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

Pentium III Xeon is a registered trademark of Intel.

Performix is a trademark of Pure Atria, Inc.

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

Preface

The TPC Benchmark C was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry. This full disclosure report is based on the TPC Benchmark C Standard Specifications Version 3.4, released August 25, 1998.

TPC Benchmark C Overview

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

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

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

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

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

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

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark C test conducted on the Compaq ProLiant 7000. The operating system used for the benchmark was Windows NT 4.0 Enterprise Edition. The DBMS used was Microsoft SQL Server 7.0 Enterprise Edition.

TPC Benchmark C Metrics

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

25,065.30 tpmC
$18.88 per tpmC

The availability date is August 16, 1999

Standard and Executive Summary Statements

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

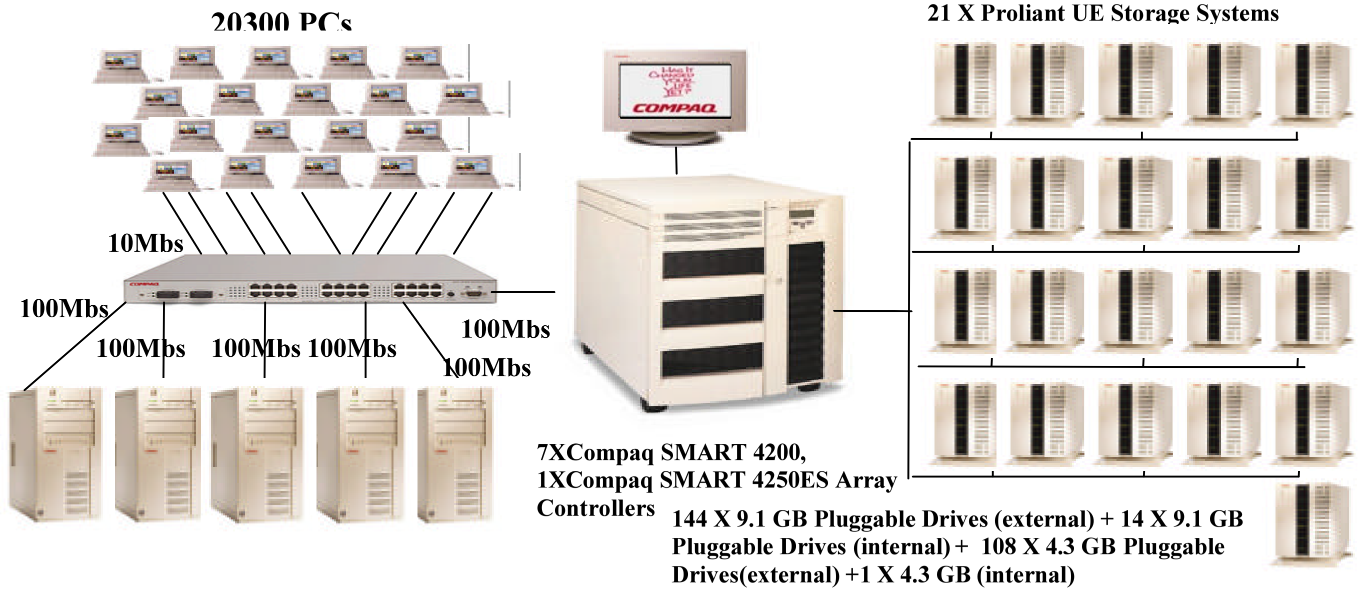
Auditor

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

Compaq Computer Corporation	ProLiant 7000-6/500-2S C/S with 5 ProLiant 1600	TPC-C Rev. 3.4 Report Date:June 8,1999

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$473,346	**25,065.30**	**$18.88**	**August 16, 1999**

Processors	Database Manager	Operating System	Other Software	Number of Users
4 Pentium III Xeon 500 MHz	Microsoft SQL Server 7.0 Enterprise Edition	Windows NT 4.0 Enterprise Edition (SP4)	NonStopTuxedo 1.3 Microsoft Visual C++ MKS Toolkit for NT	**20,300**


20300 PCs
10Mbs
100Mbs
100Mbs
100Mbs 100Mbs
100Mbs
100Mbs
COMPAQ
21 X Proliant UE Storage Systems
7XCompaq SMART 4200,
1XCompaq SMART 4250ES Array
Controllers
144 X 9.1 GB Pluggable Drives (external) + 14 X 9.1 GB Pluggable Drives (internal) + 108 X 4.3 GB Pluggable Drives(external) +1 X 4.3 GB (internal)

	Server		Each Client	
System Components	Quantity	Description	Quantity	Description
Processor	4	500MHz PentiumIII Xeon w/ 2MB Cache	2	450MhzPentium II w/ 512K cache
Memory	32	128MB	4	128MB
Disk Controllers	7	SMART SCSI Array Controller 4200	1	Integrated Ultra SCSI Controller
	1	SMART SCSI Array Controller 4250ES		
Disk Drives	158	9.1GB SCSI Drive	1	4.3GB SCSI Drive
	109	4.3GB SCSI Drive		
Total Storage		1.7593 TeraBytes		4.3GB
Tape Drives	1	4/16 TurboDAT		

Compaq Computer Corporation	ProLiant 7000- 6/500-2S Client/Server	TPC-C Rev. 3.4	
		Report Date:	8-Jun-99

Description	Part Number	Third Party Brand	Pricing	Unit Price	Qty	Extended Price	5 yr. Maint. Price
Server Hardware							
ProLiant 7000-6/500-2S	112893-001		1	28,728	1	28,728	
2 x Pentium II Xeon/500MHz - 2MB Cache							
3 x Integrated Fast-SCSI-3 Controller							
CD-ROM, 4250ES SMART Cont. ,512MB Memory							
10/100 Ethernet Controller							
ProLiant Xeon 500 - 2MB Processor Kit	328814-B21		1	6,720	2	13,440	
512MB EDO DIMM Memory Kit (50ns)	328582-B21		1	1,677	7	11,736	
Compaq SMART Array Controller 4200	295636-B21		1	2,218	7	15,523	
ProLiant Storage System -UE	348710-002		1	1,903	21	39,960	
4.3-GB UW SCSI-3 10,000 RPM Drive	304858-B21		1	503	1	503	
Compaq VGA V55 Color Monitor	308006-001		1	240	1	240	
4/8GB DAT Drive	295353-B21		1	703	1	703	
Netelligent 10/100 TX Controller	317600-B21		1	75	1	75	
4.3 GB Hot-Plug SCSI-3 Drive - 10K	336356-B21		1	503	108	54,311	
9.1 GB Hot-Plug SCSI-3 Drive - 10K	336357-B21		1	839	158	132,543	
ProLiant Enterprise Server - Parts Exchange - 1 Year	FM-HEPRT-12		1	597	2		1,194
ProLiant Storage System - Parts Exchange - 1 Year	FM-STPRT-12		1	255	42		10,710
Onsite 5x8 4 Hr Service Upgrade from Compaq Standard Service		Allstar	1				23,821
					Subtotal	297,763	35,725
Server Software							
Microsoft SQL Server 7.0 Enterprise		Microsoft	2	28,999	1	28,999	10,475
MKS Toolkit for NT		MKS	1	499	1	499	5 Yr WTY
Microsoft Visual C++ 6.0		Microsoft	2	549	1	549	Incl Above
Microsoft Windows NT Enterprise v. 4.0		Microsoft	2	3,999	1	3,999	Incl Above
					Subtotal	34,046	10,475
Client Hardware							
ProLiant 1600 6/450 - 64MB,100TX Nic	313650-001		1	2,832	5	14,162	
ProLiant 1600 6/450 Processor Option Kit	313618-B21		1	1,343	5	6,714	
128-Megabyte SD DIMM Kit	313615-B21		1	327	20	6,541	
Compaq VGA V55 Color Monitor	308006-001		1	240	5	1,198	
4.3GB Wide Ultra2 SCSI Hard Drive	349525-B21		1	391	5	1,954	
ProLiant Workgroup Server - Parts Exchange - 1 Year	FM-LOPRT-12		1	259	10		2,590
Onsite 5x8 4 Hr Service Upgrade from Compaq Standard Service		Allstar	1				2,446
					Subtotal	30,570	5,036
Client Software							
Microsoft Windows NT Server v. 4.0		Microsoft	2	809	5	4,045	Incl. Above
Compaq Non-Stop Tuxedo CS for NT - 1.3				1,950	5	9,750	800
					Subtotal	13,795	800
User Connectivity							
3Com Superstack II 24 Port 10/100 Switch		3Com	1	1,110	4	4,440	See Note 1
8-port 10BaseT hub		Generic	1	29.00	2816	81,664	See Note 1
					Subtotal	86,104	0
Large Purchase and Cash discount	9.0%		1			($37,299)	($3,668)
					Total	$424,979	$48,367

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.

Five-Year Cost of Ownership:	$473,346
tpmC Rating:	25065.30
$ / tpmC:	$18.88

Pricing: 1=Allstar Systems 2= Microsoft

Note 1 = 5 Year warranty with 10% Spares -

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

Numerical Quantities Summary

MQTH, Computed Maximum Qualified Throughput		**25,065.30 tpmC**	
% throughput difference, reported & reproducibility runs			0.10%
Response Times (in seconds)	**Average**	**90%**	**Maximum**
New-Order	0.59	1.02	5.25
Payment	0.47	0.90	4.39
Order-Status	0.50	0.93	3.95
Delivery (interactive portion)	0.11	0.11	2.11
Delivery (deferred portion)	0.51	0.80	2.41
Stock-Level	2.08	2.72	5.15
Menu	0.14	0.12	4.07
Transaction Mix, in percent of total transaction			
New-Order			44.79%
Payment			43.10%
Order-Status			4.04%
Delivery			4.04%
Stock-Level			4.03%
Emulation Delay (in seconds)		**Resp.Time**	**Menu**
New-Order		N/A	N/A
Payment		N/A	N/A
Order-Status		N/A	N/A
Delivery (interactive)		N/A	N/A
Stock-Level		N/A	N/A
Keying/Think Times (in seconds)	**Min.**	**Average**	**Max.**
New-Order	18.00/0.00	18.01/12.04	18.27/121.11
Payment	3.00/0.00	3.00/12.04	3.22/121.35
Order-Status	2.00/0.00	2.01/10.04	2.08/100.90
Delivery (interactive)	2.00/0.00	2.01/5.05	2.21/50.72
Stock-Level	2.00/0.00	2.01/5.11	2.10/50.40
Test Duration			
Ramp-up time			47 minutes
Measurement interval			20 minutes
Transactions (all types) completed during measurement interval			1102286
Ramp down time			26 minutes
Checkpointing			
Number of checkpoints			1
Checkpoint interval			20 minutes

General Items

Test Sponsor

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

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

Application Code and Definition Statements

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

Appendix A contains all source code implemented in this benchmark.

Parameter Settings

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

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

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

Appendix C contains the tunable parameters 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 Configuration

Compaq Proliant 7000 6/500 Model 2S

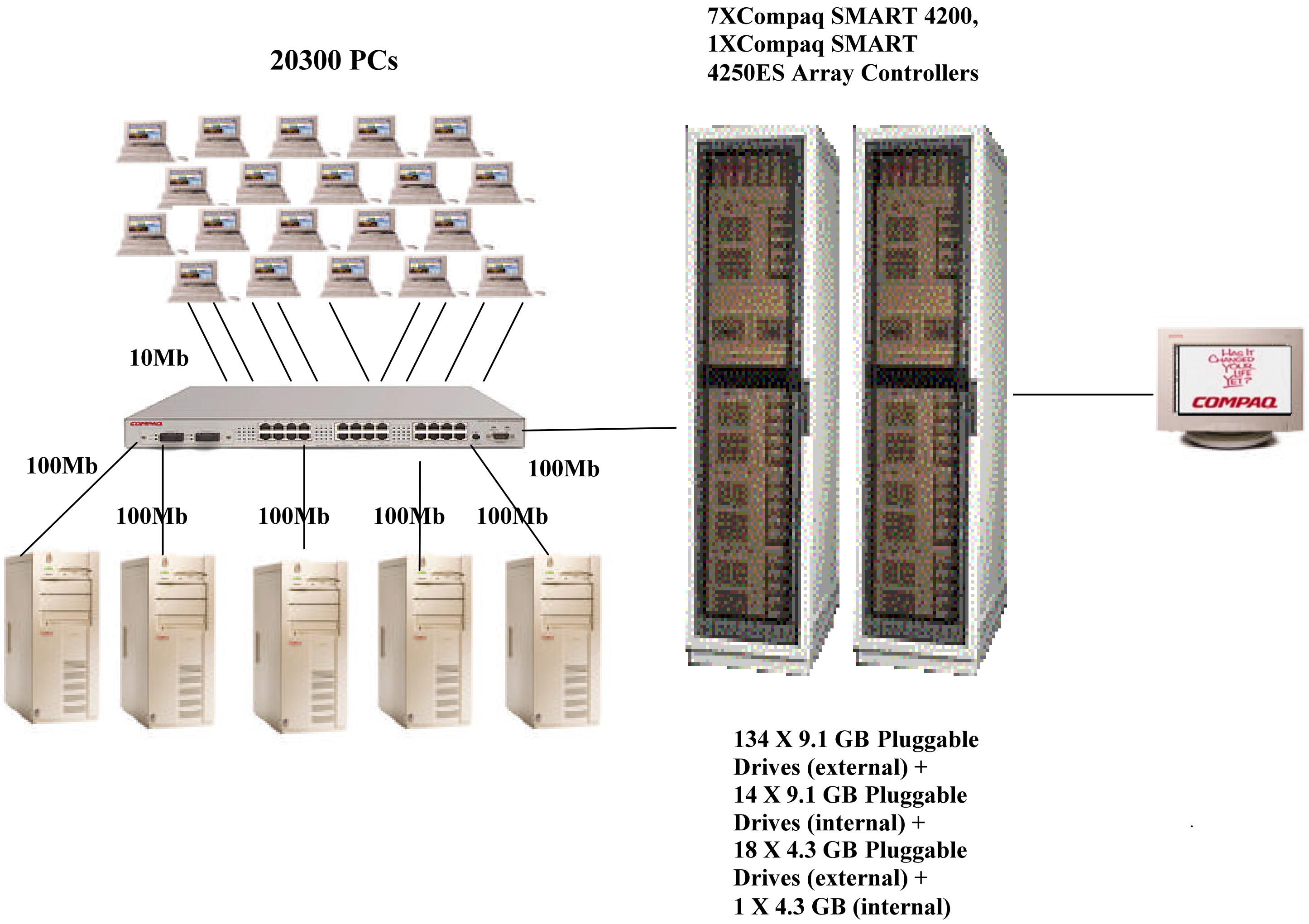
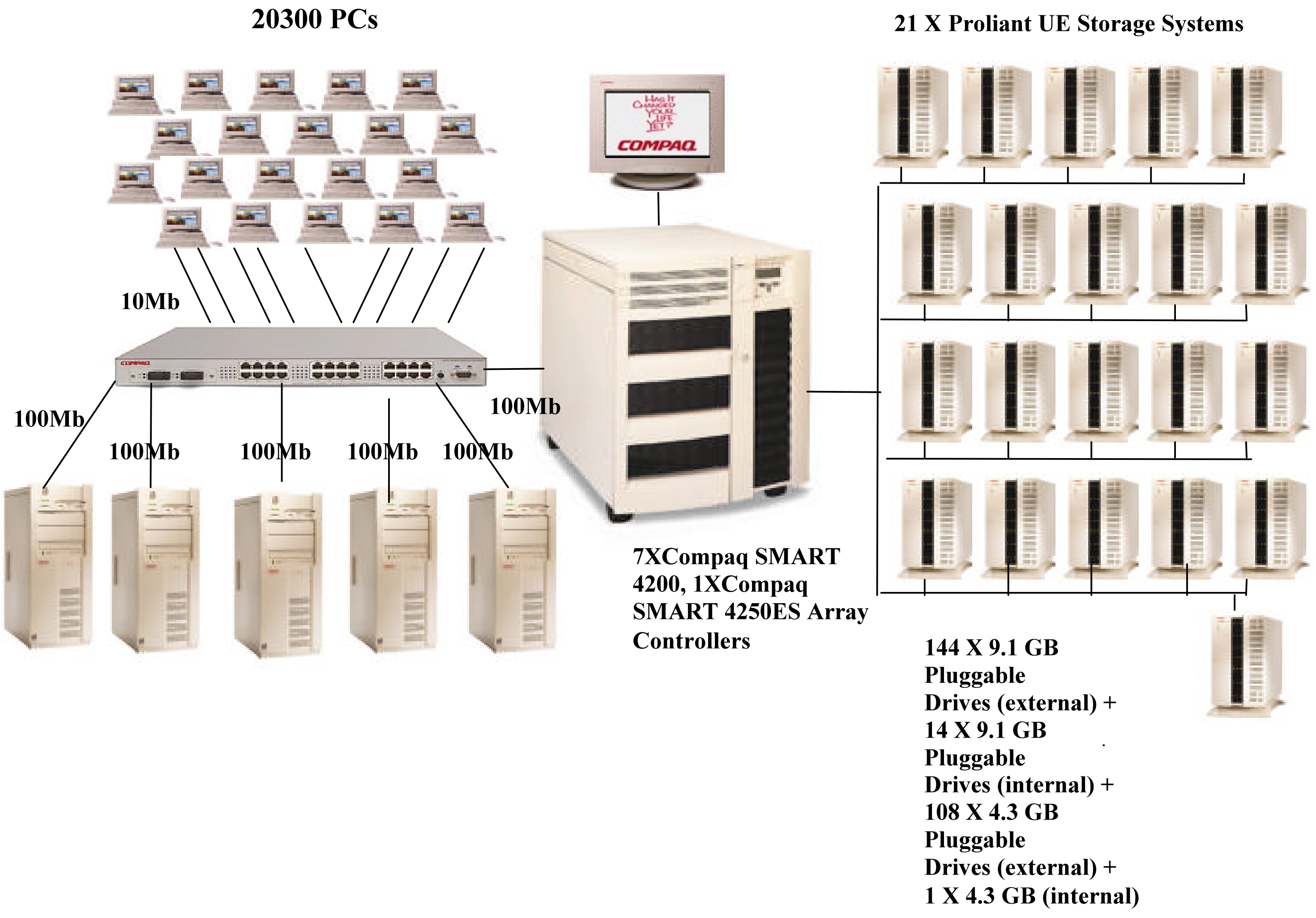


Figure 2. Priced Configuration

Compaq Proliant 7000 6/500 Model 2S


20300 PCs
21 X Proliant UE Storage Systems
10Mb
100Mb
100Mb
100Mb
100Mb
100Mb
100Mb
7XCompaq SMART 4200, 1XCompaq SMART 4250ES Array Controllers
144 X 9.1 GB Pluggable Drives (external) +
14 X 9.1 GB Pluggable Drives (internal) +
108 X 4.3 GB Pluggable Drives (external) +
1 X 4.3 GB (internal)

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: 248 drives at 9.1GB, 18 drives at 4.3GB each and 1 OS drive at 4.3GB each.

Benchmarked Configuration:

Embedded Controller,
(1 – 4.3 GB 10K drive)
EISA UTILITIES PARTITION Total Capacity = 39 MB
Compaq System Configuration Utilities
LOGICAL DRIVE C: Total Capacity = 2047 MB
Microsoft Windows NT v. 4.0
LOGICAL DRIVE D: Total Capacity = 2008 MB
NTFS Filesystem

SMART-4200 Controller, Slot 3, Array A
(36 - 9.1 GB 10K drives)
LOGICAL DRIVE G: Total Capacity = 16897 MB RAID 0
Big1
LOGICAL DRIVE N: Total Capacity = 8000 MB RAID 0
Misc1

SMART-4200 Controller, Slot 4, Array A
(36 - 9.1 GB 10K drives)
LOGICAL DRIVE H: Total Capacity = 16897 MB RAID 0
Big2
LOGICAL DRIVE O: Total Capacity = 8000 MB RAID 0
Misc2

SMART-4200 Controller, Slot 5, Array A
(18 - 9.1 GB 10K drives, 18 – 4.3 GB 10K drives)
LOGICAL DRIVE I: Total Capacity = 16897 MB RAID 0
Big3
LOGICAL DRIVE P: Total Capacity = 8000 MB RAID 0
Misc3

SMART-4200 Controller, Slot 6, Array A
(36 - 9.1 GB 10K drives)
LOGICAL DRIVE J: Total Capacity = 16897 MB RAID 0
Big4

LOGICAL DRIVE Q:	Total Capacity = 8000 MB	RAID 0
Misc4		
LOGICAL DRIVE U:	Total Capacity = 143748 MB	RAID 5
NTFS filesystem for backup		

SMART-4200 Controller, Slot 7, Array A
(36 - 9.1 GB 10K drives)

LOGICAL DRIVE K:	Total Capacity = 16897 MB	RAID 0
Big5		
LOGICAL DRIVE R:	Total Capacity = 8000 MB	RAID 0
Misc5		
LOGICAL DRIVE V:	Total Capacity = 143748 MB	RAID 5
NTFS filesystem for backup		

SMART-4200 Controller, Slot 8, Array A
(36 - 9.1 GB 10K drives)

LOGICAL DRIVE L:	Total Capacity = 16897 MB	RAID 0
Big6		
LOGICAL DRIVE S:	Total Capacity = 8000 MB	RAID 0
Misc6		
LOGICAL DRIVE Y:	Total Capacity = 143748 MB	RAID 5
NTFS filesystem for backup		

SMART-4200 Controller, Slot 9, Array A
(36 - 9.1 GB 10K drives)

LOGICAL DRIVE M:	Total Capacity = 16897 MB	RAID 0
Big7		
LOGICAL DRIVE T:	Total Capacity = 8000 MB	RAID 0
Misc7		
LOGICAL DRIVE Z:	Total Capacity = 143748 MB	RAID 5
NTFS filesystem for backup		

SMART-4250ES Controller, Slot 11, Array A
(14 - 9.1 GB 10K drives)

LOGICAL DRIVE F:	Total Capacity = 52063 MB	RAID 1
Log		

SMART-4250ES Controller, Slot 11, Array B
(2 - 9.1 GB 10K drives)

LOGICAL DRIVE	Total Capacity = 8673 MB	RAID 1
Log		

Priced Configuration vs. Measured Configuration:

The priced configuration and the tested configuration differ in that the measured configuration was run on a ProLiant 7000 Rack Mountable System with Rack Mountable ProLiant Storage Systems, whereas the priced configuration includes all tower (non-rack) ProLiant Storage Systems and a tower ProLiant 7000 server. The measured configuration has the backup devices stored on 9.1GB drives, whereas the priced configuration has tape drives for backup. In addition, the priced configuration includes 3Com Superstack II 24 Port 10/100 switches whereas the tested configuration used Compaq SW3322 24 Port 10/100 switches. The measured configuration had 248 9.1GB 10K drives and 19 4.3GB 10K drives, whereas the priced configuration had 158 9.1GB 10K drives and 109 4.3GB 10K drives.

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.

The driver used the Performix RTE from Pure Atria, Inc. to generate random numbers for the input data. The random number generator received a unique seed for each user.

The Performix RTE computes random integers using an implementation of a linear congruential sequence as described in “The Art of Computer Programming, Volume 2/ Seminumerical Algorithms” by Donald E. Knuth, copyright 1981 by Addison-Wesley Publishing Company.

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

Input/Output Screen Layout

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

All screen layouts followed the specifications exactly.

Priced Terminal Feature Verification

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

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

Presentation Manager or Intelligent Terminal

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

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

Transaction Statistics

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

Table 2. 1 Transaction Statistics

Statistic		Value
New Order	Home warehouse order lines	89.99%
	Remote warehouse order lines	10.01%
	Rolled back transactions	1.01%
	Average items per order	10.00%
Payment	Home warehouse payments	85.09%
	Remote warehouse payments	14.91%

Statistic		Value
	Accessed by last name	60.07%
Order Status	Accessed by last name	60.01%
Transaction Mix	New Order	44.79%
	Payment	43.10%
	Order status	4.04%
	Delivery	4.04%
	Stock level	4.03%

Queuing Mechanism

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

NonStop Tuxedo 1.3 on each client machine served as the queuing mechanism to the database. Each delivery request was submitted to Tuxedo 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 shell script to issue queries to the database. The results of the queries verified that the database was consistent for all four tests.

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

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

Isolation

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

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

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

For Isolation test seven, case A was followed.

Durability

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

Durable Media Failure

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

Loss of Data and Log

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

- The database was backed up to extra disks.
- The total number of New Orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.
- The RTE was started with 100 users.
- The test was allowed to run for a minimum of 10 minutes.
- One log disk was removed from the drive cabinet.
- Since the disk was mirrored, processing was not interrupted. This was verified by checking the users status on the RTE.
- One of the data disks was removed from the drive cabinet.
- When Microsoft SQL Server recorded errors about not being able to access the database, the RTE was shut down.
- A dump of the transaction log was taken and the Microsoft SQL Server was shutdown.
- A new log disk was inserted into the log drive cabinet. A new data disk was inserted into the data drive cabinet. Both cabinets were powered back up and system restarted.
- Microsoft SQL Server was started.
- The database was automatically recovered from the database backup and the transaction log dump.
- Consistency condition #3 was executed and verified.
- Step 2 was repeated and the difference between the first and second counts was noted.
- An RTE report was generated for the entire run time giving the number of NEW-ORDERS successfully returned to the RTE.
- The counts in step 12 and 13 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 2030 warehouses under a full load of 20300 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 20300 users.
- The test was allowed to run for a minimum of 10 minutes.
- A checkpoint was performed.
- System crash and loss of memory were induced by switching the power off. 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 9 and 10 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	2050
District	20,500
Customer	61,500,000
History	61,500,000
Orders	61,500,000
New Order	18,450,000
Order Line	614,997,460
Stock	205,000,000
Item	100,000
Deleted Warehouses	20

Database Layout

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

The benchmarked configuration used 7 SMART-4200 Array controllers, each with 4 SCSI channels. Each controller is capable of accessing up to 48 disk drives, 12 disk drives per each channel, and supports RAID 0, RAID 1 and RAID 5 per each logical drive configured. Three channels of each SMART-4200 Array controller were populated with 12 drives each, for a total of 36 drives per controller. The data tables were stored on 14 logical drive of 36 drives each (i.e. 2 logical volumes on set of 36 drives). Each logical drive was configured with RAID 0 and the Array Accelerator was disabled. One SMART-4250ES controller was configured with one logical drive of 12 9.1GB drives each were configured as a RAID 1 and stored the transaction log, while an additional logical drive of 2 9.1GB drives was available for expansion of the log. All 14 drives connected to the SMART-4250ES controller were located in the internal drive bays of the Proliant 7000-6/500-2S. The transaction log volumes had the Array Accelerator disabled. The operating system was stored on a 4.3GB drive on the embedded channel. 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/1, 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 7.0 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.

180 Day Space

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

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

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

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

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

Clause 5 Related Items

Throughput

Measured tpmC must be reported

Measured tpmC	25,065.30 tpmC
Price per tpmC	$18.88 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	90th %	Maximum
New-Order	0.59	1.02	5.25
Payment	0.47	0.90	4.39
Order-Status	0.50	0.93	3.95
Interactive Delivery	0.11	0.11	2.11
Deferred Delivery	0.51	0.80	2.41
Stock-Level	2.08	2.72	5.15
Menu	0.14	0.12	4.07

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	18.01	18.27
Payment	3.00	3.00	3.22
Order-Status	2.00	2.01	2.08
Interactive Delivery	2.00	2.01	2.21
Stock-Level	2.00	2.01	2.10

Table 5.4: Think Times

Type	Minimum	Average	Maximum
New-Order	0.00	12.04	121.11
Payment	0.00	12.04	121.35
Order-Status	0.00	10.04	100.90
Interactive Delivery	0.00	5.05	50.72
Stock-Level	0.00	5.11	50.40

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

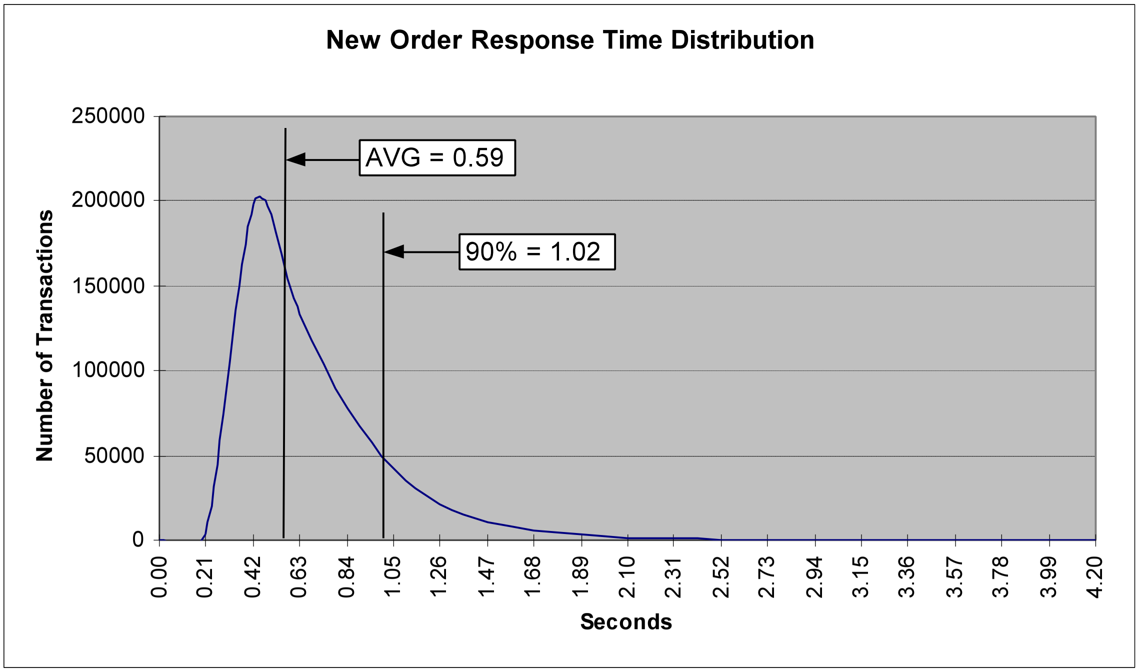


Figure 4. Payment Response Time Distribution

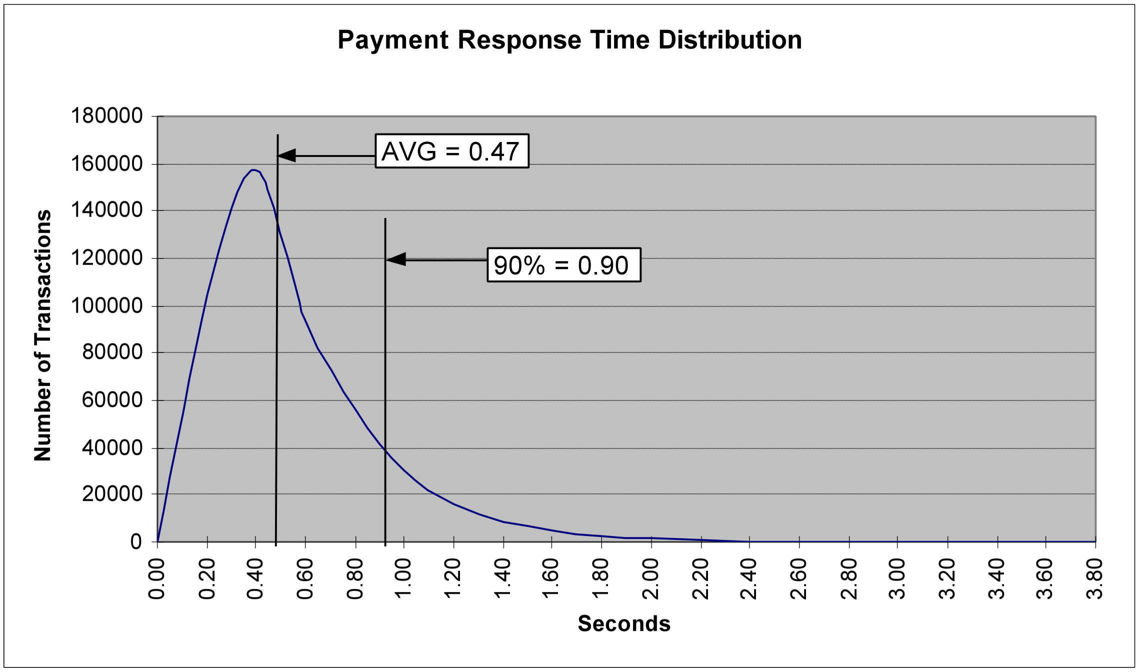


Figure 5. Order Status Response Time Distribution

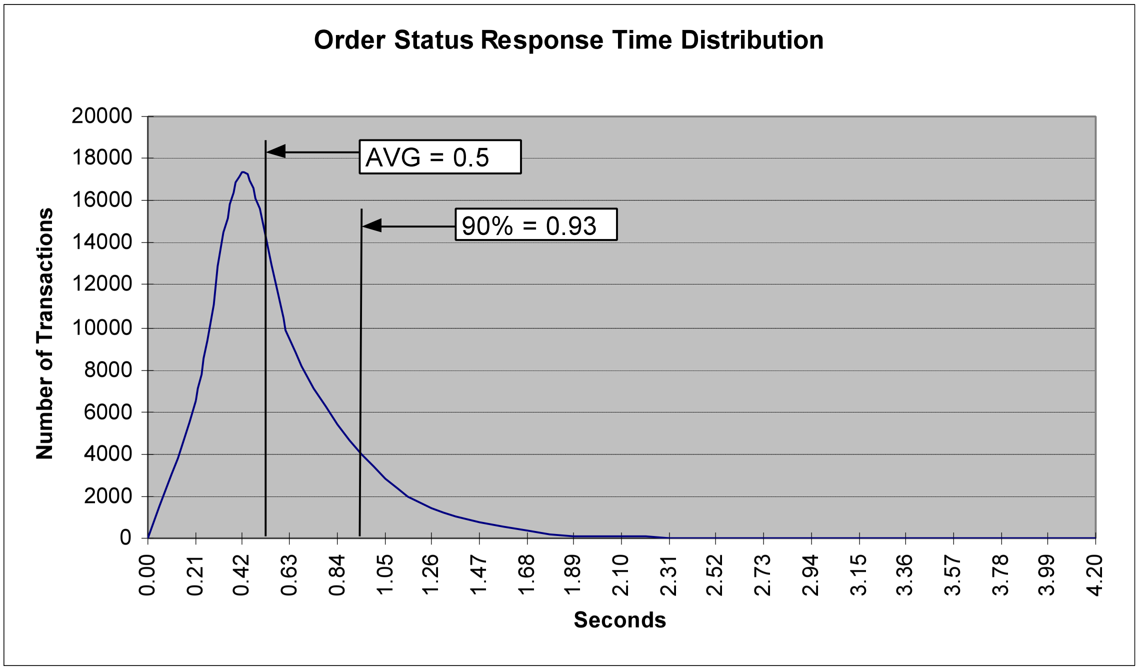


Figure 6. Delivery Response Time Distribution

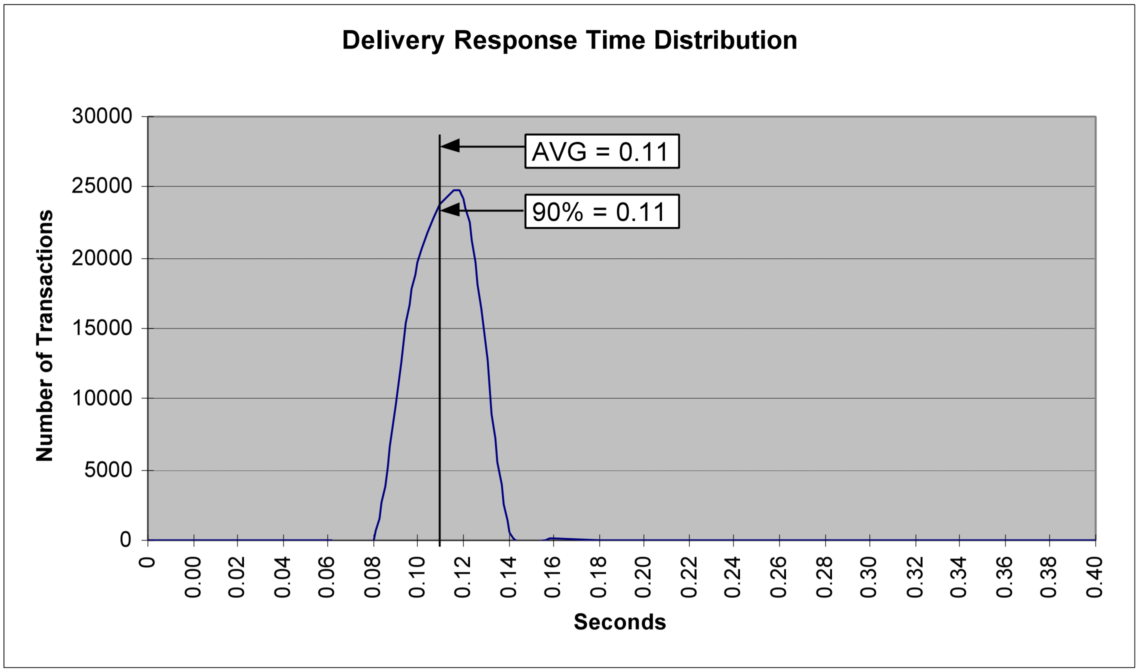


Figure 7. Stock Level Response Time Distribution

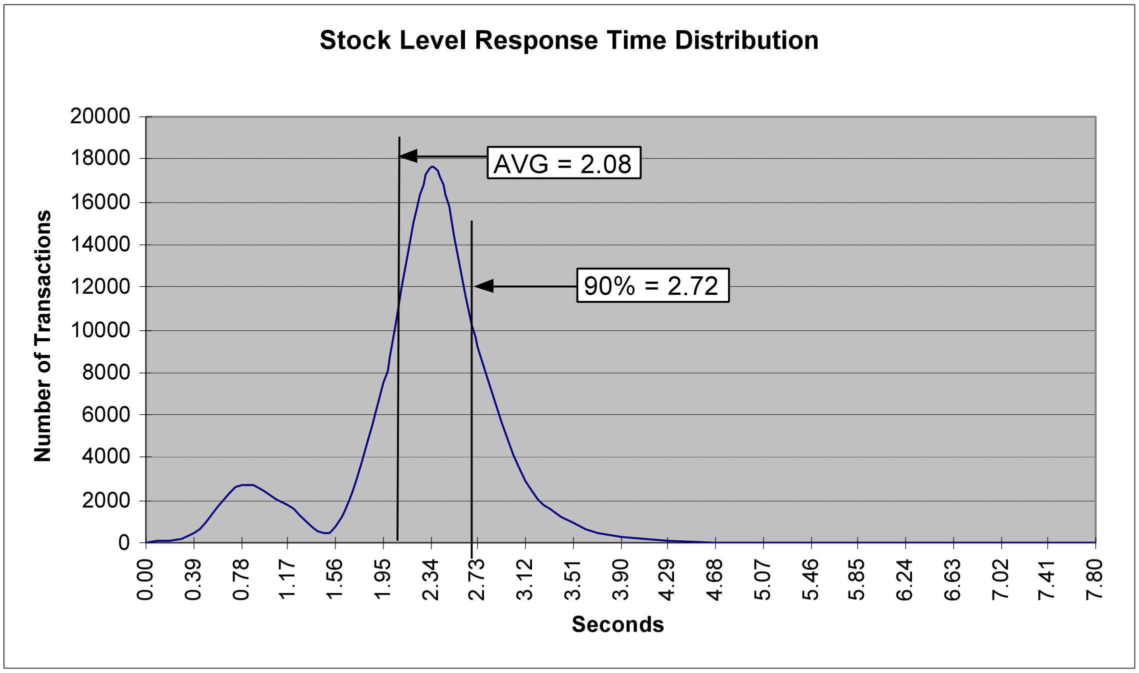


Figure 8. Response Time vs. Throughput

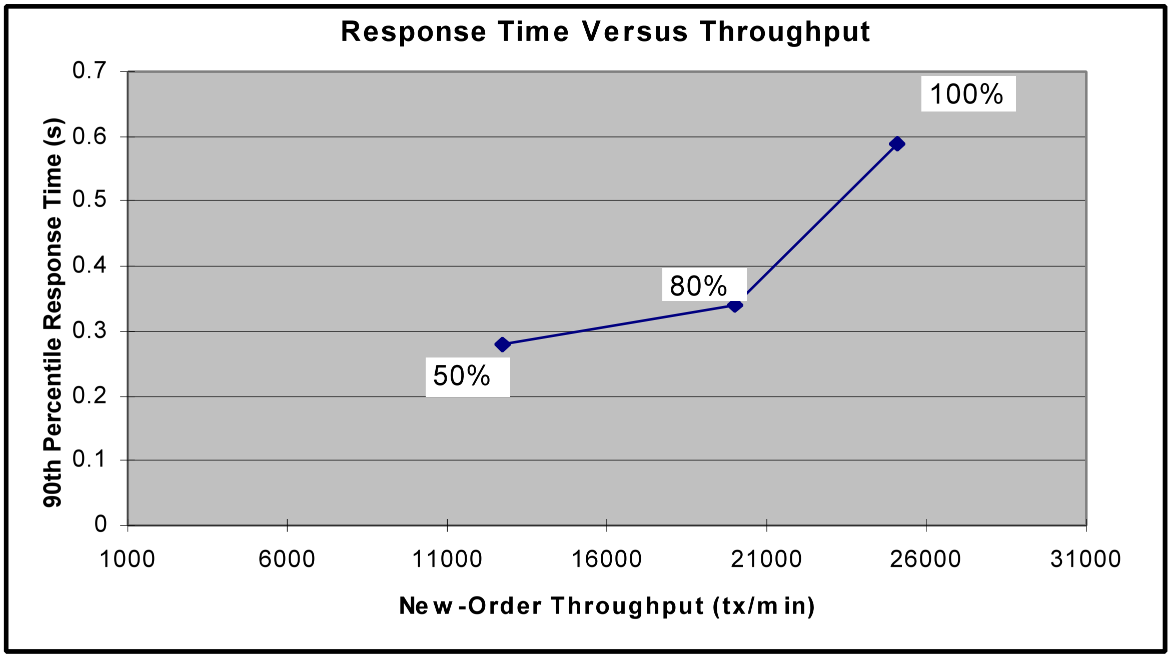


Figure 9. New Order Think Time Distribution

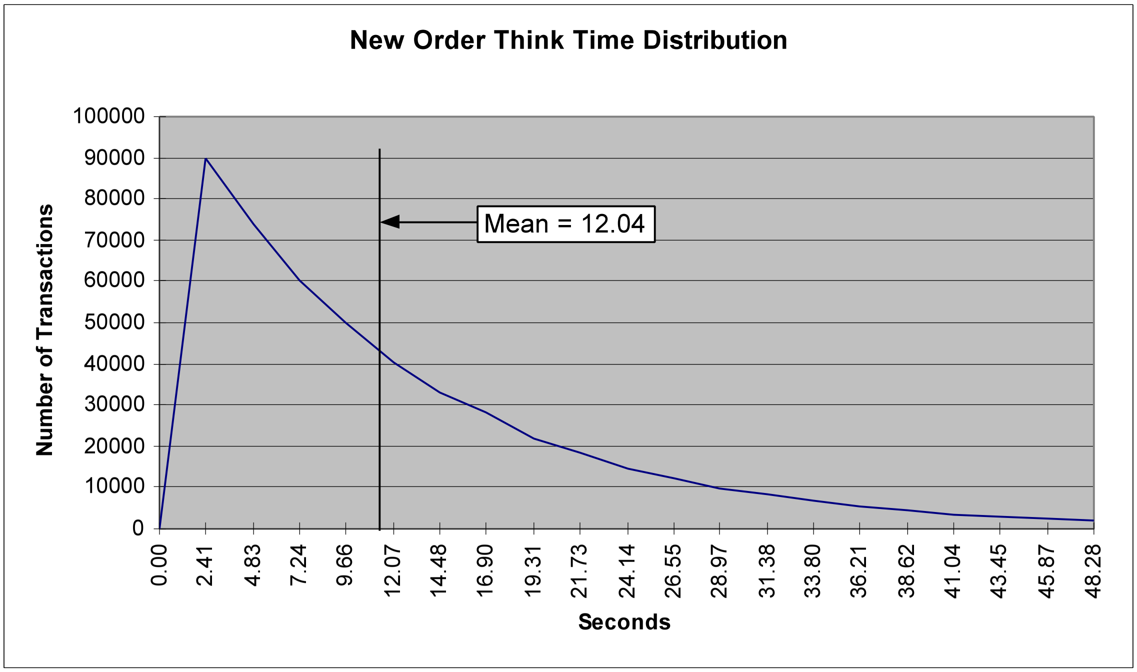
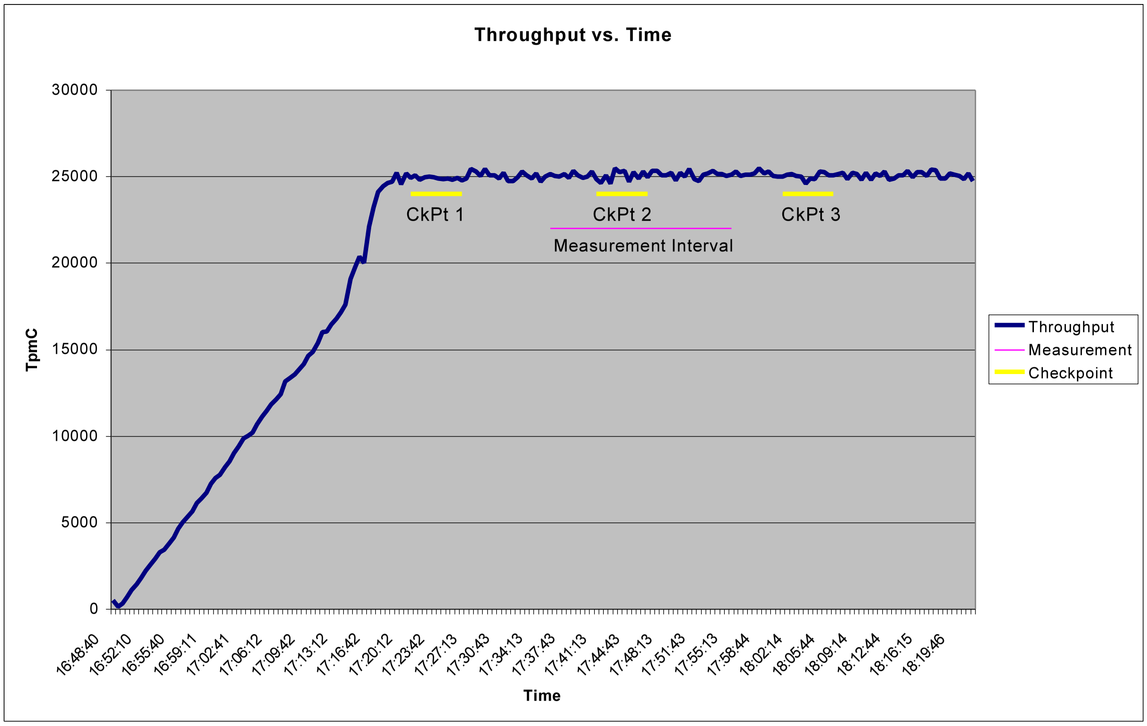


Figure 10. Throughput vs. Time Distribution



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 and captured in RTE log files before being transmitted. There was one log file for each user. The input screen for the requested transaction was returned and it was also captured and timestamped in the RTE log files. The difference between these two timestamps was the menu response time.

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 and captured in RTE log files. The return of the screen with the required response data was timestamped and captured in the RTE log files. The difference between these two timestamps was the response time for that transaction.

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

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

To perform checkpoints at specific intervals, we set SQL Server *recovery interval* to the maximum allowable value and wrote a script to schedule multiple checkpoints at specific intervals. The script included a wait time between each checkpoint equal to the measurement interval which was 20 minutes. The checkpoint script was started manually after the RTE had all users logged in and sending transactions.

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

Reproducibility

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

We allowed the database to warm up and to reach a steady state for approximately 83 minutes. The steady state was sustained for a 20-minute (measurement) interval, and was followed by a ramp-down. The repeatable interval result was within 0.10 % of the reported interval result.

Measurement Period Duration

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

The reported measured interval was exactly 20 minutes long.

Regulation of Transaction Mix

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

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

Transaction Statistics

The percentage of the total mix for each transaction type must be disclosed. The percentage of New-Order transactions rolled back as a result of invalid item number must be disclosed. The average number of order-lines entered per New-Order transaction must be disclosed. The percentage of remote order lines per New-Order transaction must be disclosed. The percentage of remote Payment transactions must be disclosed. The percentage of customer selections by customer last name in the Payment and Order-Status transactions must be disclosed. The percentage of Delivery transactions skipped due to there being fewer than necessary orders in the New-Order table must be disclosed.

Table 5.5: Transaction Statistics

Statistic		Value
New Order	Home warehouse order lines	89.99%
	Remote warehouse order lines	10.01%
	Rolled back transactions	1.01%
	Average items per order	10.00%
Payment	Home warehouse payments	85.09%
	Remote warehouse payments	14.91%
	Accessed by last name	60.07%
Order Status	Accessed by last name	60.01%
Transaction Mix	New Order	44.79%
	Payment	43.10%
	Order status	4.04%
	Delivery	4.04%
	Stock level	4.03%

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 53 minutes after the start of the ramp-up. Subsequent checkpoints occurred every 20 minutes. The checkpoint in the measurement interval lasted 5 minutes and 11 seconds. The measurement interval contains the second checkpoint, and is clear of the guard zones.

Clause 6 Related Items

RTE Descriptions

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

The RTE used was Performix by Pure Atria, Inc. and the code used with Performix is included 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 20 Compaq ProLiant 850R servers. These driver machines were attached to the five Compaq ProLiant 1600 6/450 client machines through a 10Mb/s Ethernet connection to a Compaq SW3322 Dual Speed Switch. Since this configuration is the same connectivity of the priced system, no components were being emulated. Therefore, the test described in Clause 6.6.3.4 was not required.

Functional Diagrams

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

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

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

Networks

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

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

In the tested configuration, 20 driver (RTE) machines were connected to the Compaq SW3322 24-port Dual Speed switches with 10Mb/s Ethernet connections. The five client machines were connected to the switch with 100Mb/s connections. The server (SUT) was connected to the Compaq switch with a 100Mb/s connection.

Operator Intervention

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

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

Clause 7 Related Items

System Pricing

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

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

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

Availability, Throughput, and Price Performance

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

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

• **Maximum Qualified Throughput**	**25,065.30 tpmC**
• **Price per tpmC**	**$18.88 per tpmC**
• **Availability**	**August 16, 1999**

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:

- 5 Microsoft Windows NT 4.0 licenses
- 1 Microsoft Windows NT 4.0 Enterprise Edition license
- One Microsoft SQL Server 7.0 Enterprise Edition (Unlimited User Licenses)
- One Microsoft Visual C++ 6.0
- Compaq Servers include 3 years of support.

Clause 9 Related Items

Auditor’s Report

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

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

Performance Metrics, Inc.
2229 Benita Drive, Suite 101
Rancho Cordova, CA 95670
(phone) (916) 635-2822
(fax) (916) 858-0109
e-mail: lorna@perfmetrics.com

Availability of the Full Disclosure Report

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

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

Transaction Processing Performance Council
c/o Shanley Public Relations
777 North First Street, Suite 6000
San Jose, CA 95112-6311

or

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

PERFORMANCE METRICS INC.
TPC Certified Auditors

June 7, 1999

Mr. Andy Bond
Senior Database Performance Engineer
Compaq Computer Corporation
20555 SH 249
Houston, TX 77070

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

Platform:	ProLiant 7000-6/500
Database Manager:	Microsoft SQL Server 7.0 Enterprise Edition
Operating System:	Microsoft NT version 4.0 Enterprise Edition
Transaction Monitor:	Compaq Nonstop Tuxedo 1.3

Servers: ProLiant 7000 with:				
CPU's	Memory	Disks (total)	90% Response	TpmC
4 Pentium III Xeon @ 500 Mhz	Main: 4096 MB Cache: 2 MB	248 @ 9.1GB 18 @ 4.3 GB	1.02 sec	25,065.30
5 Clients: ProLiant 1600 each with:				
2 Pentium II @ 450 Mhz	Main: 512 MB Cache: 512K	1 @ 4.3GB	na	Na

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

- The transactions were correctly implemented.
- The database files were properly sized and populated.
- The database was properly scaled with 2050 warehouses of which only 2030 were active during the measured interval.
- The ACID properties were successfully demonstrated including isolation and durability across nodes.

- The ACID properties, except for system loss, were demonstrated on a subset of the SUT configured with a database properly populated for 10 warehouses.
- Input data was generated according to the specified percentages.
- Eight hours of mirrored log space was present on the tested system.
- Eight hours of growth space for the dynamic tables was present on the tested system.
- The data for the 180 day space calculation was verified.
- The controller cache was disabled on all disk controllers.
- The steady state portion of the test was 20 minutes.
- One checkpoint was taken before the measured interval.
- One checkpoint was taken during the measured interval.
- The checkpoints were verified to be clear of the guard zone.
- The system pricing was checked for major components and maintenance.
- Third party quotes were verified for compliance.

Auditor Notes: Disk substitution:
The tested configuration included both 4.3 and 9.1 GB disk drives configured and used for database tables. The priced configuration has substituted 90 of the 4.3 disks for the 9.1 disks. This change is in compliance with the requirements for disk substitution. The technical specifications on these drives are identical and performance data collected shows almost equal performance with only a slight advantage towards the 4.3 disks.

Sincerely,

Lorna Livingtree

Lorna Livingtree
Auditor

Appendix A: Source Code

The client source code is listed below.

MIX

```
mon1, ./norate null mon1.l ./no1.log
mon2@r2, ./norate null mon2.l ./no2.log
mon3@r3, ./norate null mon3.l ./no3.log
mon4@r4, ./norate null mon4.l ./no4.log
mon5@r5, ./norate null mon5.l ./no5.log
mon6@r6, ./norate null mon6.l ./no6.log
mon7@r7, ./norate null mon7.l ./no7.log
mon8@r8, ./norate null mon8.l ./no8.log
mon9@r9, ./norate null mon9.l ./no9.log
mon10@r10, ./norate null mon10.l ./no10.log
mon11@r11, ./norate null mon11.l ./no11.log
mon12@r12, ./norate null mon12.l ./no12.log
mon13@r13, ./norate null mon13.l ./no13.log
mon14@r14, ./norate null mon14.l ./no14.log
mon15@r15, ./norate null mon15.l ./no15.log
mon16@r16, ./norate null mon16.l ./no16.log
mon17@r17, ./norate null mon17.l ./no17.log
mon18@r18, ./norate null mon18.l ./no18.log
mon19@r19, ./norate null mon19.l ./no19.log
mon20@r20, ./norate null mon20.l ./no20.log
user0001, ./tpcc -l 3 log/u0001.l client tigger2 1 1 2030
user0002, ./tpcc -l 3 log/u0002.l client tigger2 1 2 2030
user0003, ./tpcc -l 3 log/u0003.l client tigger2 1 3 2030
user0004, ./tpcc -l 1 log/u0004.l client tigger2 1 4 2030
user0005, ./tpcc -l 1 log/u0005.l client tigger2 1 5 2030
.
.
.
user1016@r2, ./tpcc -l 3 log/u1016.l client tigger2 102 6 2030
user1017@r2, ./tpcc -l 3 log/u1017.l client tigger2 102 7 2030
user1018@r2, ./tpcc -l 3 log/u1018.l client tigger2 102 8 2030
user1019@r2, ./tpcc -l 1 log/u1019.l client tigger2 102 9 2030
user1020@r2, ./tpcc -l 1 log/u1020.l client tigger2 102 10 2030
.
.
.
user2031@r3, ./tpcc -l 3 log/u2031.l client tigger2 204 1 2030
user2032@r3, ./tpcc -l 3 log/u2032.l client tigger2 204 2 2030
user2033@r3, ./tpcc -l 3 log/u2033.l client tigger2 204 3 2030
user2034@r3, ./tpcc -l 1 log/u2034.l client tigger2 204 4 2030
user2035@r3, ./tpcc -l 1 log/u2035.l client tigger2 204 5 2030
.
.
.
user3046@r4, ./tpcc -l 3 log/u3046.l client tigger2 305 6 2030
user3047@r4, ./tpcc -l 3 log/u3047.l client tigger2 305 7 2030
user3048@r4, ./tpcc -l 3 log/u3048.l client tigger2 305 8 2030
user3049@r4, ./tpcc -l 1 log/u3049.l client tigger2 305 9 2030
user3050@r4, ./tpcc -l 1 log/u3050.l client tigger2 305 10 2030
.
.
.
user4061@r5, ./tpcc -l 3 log/u4061.l client tigger2 407 1 2030
user4062@r5, ./tpcc -l 3 log/u4062.l client tigger2 407 2 2030
user4063@r5, ./tpcc -l 3 log/u4063.l client tigger2 407 3 2030
user4064@r5, ./tpcc -l 1 log/u4064.l client tigger2 407 4 2030
user4065@r5, ./tpcc -l 1 log/u4065.l client tigger2 407 5 2030
.
.
.
user5076@r6, ./tpcc -l 3 log/u5076.l client tigger2 508 6 2030
user5077@r6, ./tpcc -l 3 log/u5077.l client tigger2 508 7 2030
user5078@r6, ./tpcc -l 3 log/u5078.l client tigger2 508 8 2030
user5079@r6, ./tpcc -l 1 log/u5079.l client tigger2 508 9 2030
user5080@r6, ./tpcc -l 1 log/u5080.l client tigger2 508 10 2030
.
.
.
user6091@r7, ./tpcc -l 3 log/u6091.l client tigger2 610 1 2030
user6092@r7, ./tpcc -l 3 log/u6092.l client tigger2 610 2 2030
user6093@r7, ./tpcc -l 3 log/u6093.l client tigger2 610 3 2030
user6094@r7, ./tpcc -l 1 log/u6094.l client tigger2 610 4 2030
user6095@r7, ./tpcc -l 1 log/u6095.l client tigger2 610 5 2030
.
.
.
user7106@r8, ./tpcc -l 3 log/u7106.l client tigger2 711 6 2030
user7107@r8, ./tpcc -l 3 log/u7107.l client tigger2 711 7 2030
user7108@r8, ./tpcc -l 3 log/u7108.l client tigger2 711 8 2030
user7109@r8, ./tpcc -l 1 log/u7109.l client tigger2 711 9 2030
user7110@r8, ./tpcc -l 1 log/u7110.l client tigger2 711 10 2030
user7111@r8, ./tpcc -l 1 log/u7111.l client tigger2 712 1 2030
.
.
.
user8122@r9, ./tpcc -l 3 log/u8122.l client tigger2 813 2 2030
user8123@r9, ./tpcc -l 3 log/u8123.l client tigger2 813 3 2030
user8124@r9, ./tpcc -l 1 log/u8124.l client tigger2 813 4 2030
user8125@r9, ./tpcc -l 1 log/u8125.l client tigger2 813 5 2030
.
.
.
user9136@r10, ./tpcc -l 3 log/u9136.l client tigger2 914 6 2030
user9137@r10, ./tpcc -l 3 log/u9137.l client tigger2 914 7 2030
user9138@r10, ./tpcc -l 3 log/u9138.l client tigger2 914 8 2030
user9139@r10, ./tpcc -l 1 log/u9139.l client tigger2 914 9 2030
user9140@r10, ./tpcc -l 1 log/u9140.l client tigger2 914 10 2030
.
.
.
user10151@r11, ./tpcc -l 3 log/u10151.l client tigger2 1016 1 2030
user10152@r11, ./tpcc -l 3 log/u10152.l client tigger2 1016 2 2030
user10153@r11, ./tpcc -l 3 log/u10153.l client tigger2 1016 3 2030
user10154@r11, ./tpcc -l 1 log/u10154.l client tigger2 1016 4 2030
user10155@r11, ./tpcc -l 1 log/u10155.l client tigger2 1016 5 2030
.
.
.
user11166@r12, ./tpcc -l 3 log/u11166.l client tigger2 1117 6 2030
user11167@r12, ./tpcc -l 3 log/u11167.l client tigger2 1117 7 2030
```

```
user11168@r12, ./tpcc -l 3 log/u11168.l client tigger2 1117 8 2030
user11169@r12, ./tpcc -l 1 log/u11169.l client tigger2 1117 9 2030
user11170@r12, ./tpcc -l 1 log/u11170.l client tigger2 1117 10 2030
.
.
.
user12181@r13, ./tpcc -l 3 log/u12181.l client tigger2 1219 1 2030
user12182@r13, ./tpcc -l 3 log/u12182.l client tigger2 1219 2 2030
user12183@r13, ./tpcc -l 3 log/u12183.l client tigger2 1219 3 2030
user12184@r13, ./tpcc -l 1 log/u12184.l client tigger2 1219 4 2030
user12185@r13, ./tpcc -l 1 log/u12185.l client tigger2 1219 5 2030
.
.
.
user13196@r14, ./tpcc -l 3 log/u13196.l client tigger2 1320 6 2030
user13197@r14, ./tpcc -l 3 log/u13197.l client tigger2 1320 7 2030
user13198@r14, ./tpcc -l 3 log/u13198.l client tigger2 1320 8 2030
user13199@r14, ./tpcc -l 1 log/u13199.l client tigger2 1320 9 2030
user13200@r14, ./tpcc -l 1 log/u13200.l client tigger2 1320 10 2030
.
.
.
user14211@r15, ./tpcc -l 3 log/u14211.l client tigger2 1422 1 2030
user14212@r15, ./tpcc -l 3 log/u14212.l client tigger2 1422 2 2030
user14213@r15, ./tpcc -l 3 log/u14213.l client tigger2 1422 3 2030
user14214@r15, ./tpcc -l 1 log/u14214.l client tigger2 1422 4 2030
user14215@r15, ./tpcc -l 1 log/u14215.l client tigger2 1422 5 2030
.
.
.
user15226@r16, ./tpcc -l 3 log/u15226.l client tigger2 1523 6 2030
user15227@r16, ./tpcc -l 3 log/u15227.l client tigger2 1523 7 2030
user15228@r16, ./tpcc -l 3 log/u15228.l client tigger2 1523 8 2030
user15229@r16, ./tpcc -l 1 log/u15229.l client tigger2 1523 9 2030
user15230@r16, ./tpcc -l 1 log/u15230.l client tigger2 1523 10 2030
.
.
.
user16241@r17, ./tpcc -l 3 log/u16241.l client tigger2 1625 1 2030
user16242@r17, ./tpcc -l 3 log/u16242.l client tigger2 1625 2 2030
user16243@r17, ./tpcc -l 3 log/u16243.l client tigger2 1625 3 2030
user16244@r17, ./tpcc -l 1 log/u16244.l client tigger2 1625 4 2030
user16245@r17, ./tpcc -l 1 log/u16245.l client tigger2 1625 5 2030
.
.
.
user17256@r18, ./tpcc -l 3 log/u17256.l client tigger2 1726 6 2030
user17257@r18, ./tpcc -l 3 log/u17257.l client tigger2 1726 7 2030
user17258@r18, ./tpcc -l 3 log/u17258.l client tigger2 1726 8 2030
user17259@r18, ./tpcc -l 1 log/u17259.l client tigger2 1726 9 2030
user17260@r18, ./tpcc -l 1 log/u17260.l client tigger2 1726 10 2030
.
.
.
user18271@r19, ./tpcc -l 3 log/u18271.l client tigger2 1828 1 2030
user18272@r19, ./tpcc -l 3 log/u18272.l client tigger2 1828 2 2030
user18273@r19, ./tpcc -l 3 log/u18273.l client tigger2 1828 3 2030
user18274@r19, ./tpcc -l 1 log/u18274.l client tigger2 1828 4 2030
user18275@r19, ./tpcc -l 1 log/u18275.l client tigger2 1828 5 2030
.
.
.
user19286@r20, ./tpcc -l 3 log/u19286.l client tigger2 1929 6 2030
user19287@r20, ./tpcc -l 3 log/u19287.l client tigger2 1929 7 2030
user19288@r20, ./tpcc -l 3 log/u19288.l client tigger2 1929 8 2030
user19289@r20, ./tpcc -l 1 log/u19289.l client tigger2 1929 9 2030
user19290@r20, ./tpcc -l 1 log/u19290.l client tigger2 1929 10 2030
.
.
.
user20296@r20, ./tpcc -l 1 log/u20296.l client tigger2 2030 6 2030
user20297@r20, ./tpcc -l 1 log/u20297.l client tigger2 2030 7 2030
user20298@r20, ./tpcc -l 1 log/u20298.l client tigger2 2030 8 2030
user20299@r20, ./tpcc -l 1 log/u20299.l client tigger2 2030 9 2030
user20300@r20, ./tpcc -l 1 log/u20300.l client tigger2 2030 10 2030
```

CLIENT MAKEFILE

```
!IF "$(CFG)" == ""
CFG=Release
!MESSAGE No configuration specified.  Defaulting to Debug
!ENDIF

!IF "$(SQL_LOC)" == ""
SQL_LOC=C:\prg_tool.kit\i386
!MESSAGE No SQL_LOC specified.  Defaulting to C:\prg_tool.kit\i386
!ENDIF

!IF "$(CFG)" != "Release" && "$(CFG)" != "Debug"
!MESSAGE Invalid configuration "$(CFG)" specified.
!MESSAGE You can specify a configuration when running NMAKE on this makefile
!MESSAGE by defining the macro CFG on the command line.  For example:
!MESSAGE
!MESSAGE NMAKE CFG="Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "Release"
!MESSAGE "Debug"
!MESSAGE
!ERROR An invalid configuration is specified.
!ENDIF

OUTDIR          = .
SRCDIR          = .\Src
OBJDIR          = .\Objs
OUTDIR          = .\Bin
DBLIB           = $(SQL_LOC)
DBLIBINC        = $(DBLIB)
DBLIBDIR        = $(DBLIB)
TUXDIR              = C:\TUXEDO

!IF "$(CFG)" != "Debug"
LDEBUG       =
CDEBUG       =
LDEBUG_RG    =
CDEBUG_RG    =
DEBUG        =
FLAGS        = /D "WIN32" /D "_WINDOWS" /D "_TMSTHREADS" /D "LOCAL_ALLOC"
OPT          = /G6 /Ox /Gs16384 /F262144
!ELSE
LDEBUG       = /debug /pdb:$(OBJDIR)\tpcc.pdb
CDEBUG       = /Zi /Yd /Fd$(OBJDIR)\tpcc.pdb
```

```
LDEBUG_RG    = /debug /pdb:$(OBJDIR)\install.pdb
CDEBUG_RG    = /Zi /Yd /Fd$(OBJDIR)\install.pdb
FLAGS        = /D "_DEBUG" /D "WIN32" /D "_WINDOWS" /D "_TMSTHREADS" /D
"LOCAL_ALLOC"
OPT          = /Od
!ENDIF

CFLAGS       = /D _TMSTHREADS

LINK32_LIBS     = user32.lib msacm32.lib advapi32.lib $(DBLIBDIR)\ntwdblib.lib
TUX_LIBS  = $(TUXDIR)\lib\libtux.lib $(TUXDIR)\lib\libbuft.lib
$(TUXDIR)\lib\libtux2.lib  \
$(TUXDIR)\lib\libfml.lib $(TUXDIR)\lib\libfml32.lib $(TUXDIR)\lib\libgp.lib
OTHER_LIBS          = wsock32.lib kernel32.lib gdi32.lib comdlg32.lib winspool.lib
LINK32_OBJS     = "$(OBJDIR)\tpcc.obj" "$(OBJDIR)\tpcc.res"
LINK32_DEF      = "$(SRCDIR)\tpcc.def"
LINK32_FLAGS    = /nologo /subsystem:windows /dll /incremental:no $(LDEBUG)
/def:"$(LINK32_DEF)" /out:"$(OUTDIR)\tpcc.dll"

LINK32_LIBS_RG  = user32.lib gdi32.lib advapi32.lib version.lib comctl32.lib
LINK32_OBJS_RG  = "$(OBJDIR)\install.obj" "$(OBJDIR)\install.res"
LINK32_FLAGS_RG = /nologo /subsystem:windows /incremental:no $(LDEBUG_RG)
/out:$(OUTDIR)\install.exe

ALL:    $(OBJDIR)\. $(OUTDIR)\. $(OUTDIR)\tpcc.dll $(OUTDIR)\Neworder.exe \
          $(OUTDIR)\Payment.exe $(OUTDIR)\q.exe \
          $(OUTDIR)\Stocklevel.exe $(OUTDIR)\Orderstatus.exe $(OUTDIR)\Delivery.exe
$(OUTDIR)\Delirpt.exe

touch:
          touch $(SRCDIR)/*.c

$(OBJDIR)\.:
          if not exist $(OBJDIR) md $(OBJDIR)

$(OUTDIR)\.:
          if not exist $(OUTDIR) md $(OUTDIR)

"$(OBJDIR)\tpcc.obj":   "$(SRCDIR)\tpcc.c" "$(SRCDIR)\tpcc.h"
          cl.exe /nologo /MT /W3 $(CDEBUG) $(OPT) /I $(DBLIBINC) /I
$(TUXDIR)\include $(FLAGS) /Fo$(OBJDIR)\tpcc.obj /c "$(SRCDIR)\tpcc.c"

$(OBJDIR)\tpcc.res:     $(SRCDIR)\tpcc.rc
          rc.exe /l 0x409 /fo $(OBJDIR)\tpcc.res $(FLAGS) $(SRCDIR)\tpcc.rc

$(OUTDIR)\tpcc.dll:     $(LINK32_OBJS) $(LINK32_DEF)
          link.exe   $(LINK32_FLAGS) $(LINK32_OBJS) $(LINK32_LIBS) $(TUX_LIBS)
$(OTHER_LIBS)

$(OUTDIR)\q.exe: $(SRCDIR)\q.c
          $(TUXDIR)\bin\buildserver -v -k /f $(SRCDIR)\q.c /o $(OUTDIR)\q.exe \
/s Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10 /l "$(LINK32_LIBS) /I $(DBLIBINC)"
          del q.obj

$(OUTDIR)\Neworder.exe: $(SRCDIR)\neworder.c
          $(TUXDIR)\bin\buildserver -v -k /f $(SRCDIR)\neworder.c /o
$(OUTDIR)\Neworder.exe \
/s NEWORDER:NEWORDER /l "$(LINK32_LIBS) /I $(DBLIBINC)"
          del neworder.obj

$(OUTDIR)\Payment.exe: $(SRCDIR)\payment.c
          $(TUXDIR)\bin\buildserver /f $(SRCDIR)\payment.c /o $(OUTDIR)\Payment.exe
\
/s PAYMENT:PAYMENT /l "$(LINK32_LIBS) /I $(DBLIBINC)"
          del payment.obj

$(OUTDIR)\Orderstatus.exe: $(SRCDIR)\orderstatus.c
          $(TUXDIR)\bin\buildserver /f $(SRCDIR)\orderstatus.c /o
$(OUTDIR)\Orderstatus.exe \
/s ORDERSTATUS:ORDERSTATUS /l "$(LINK32_LIBS) /I  $(DBLIBINC)"
          del orderstatus.obj

$(OUTDIR)\Stocklevel.exe: $(SRCDIR)\stocklevel.c
          $(TUXDIR)\bin\buildserver /f $(SRCDIR)\stocklevel.c /o
$(OUTDIR)\Stocklevel.exe \
/s STOCKLEVEL:STOCKLEVEL /l "$(LINK32_LIBS) /I $(DBLIBINC)"
          del stocklevel.obj

$(OUTDIR)\Delivery.exe: $(SRCDIR)\delivery.c
          $(TUXDIR)\bin\buildserver /f $(SRCDIR)\delivery.c /o
$(OUTDIR)\Delivery.exe \
/s DELIVERY:DELIVERY /l "$(LINK32_LIBS) /I  $(DBLIBINC)"
          del delivery.obj

$(OUTDIR)\Delirpt.exe: $(SRCDIR)\delirpt.c
          cl.exe $(SRCDIR)\delirpt.c /o "$(OUTDIR)\Delirpt.exe"
          del delirpt.obj
```

DELIRPT.C

```
/*        FILE:               DELIRPT.C
 *                                      Microsoft TPC-C Kit Ver. 3.00.000
 *
 *                                      Copyright Microsoft, 1996
 *
 *        PURPOSE:  Delivery report processing application
 *        Author:             Philip Durr
 *                                      philipdu@Microsoft.com
 */

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

#define LOGFILE_READ_EOF      0
                              //check log file flag return current state
#define LOGFILE_CLEAR_EOF     1
                              //clear end of log file flag
#define LOGFILE_SET_EOF                 2
                                        //set flag end of log file reached

#define INTERVAL                        .01
                                        //90th percentile calculation bucket
interval

#define ERR_SUCCESS                                                   1000
                              //success no error
#define ERR_READING_LOGFILE                                 1001
                    //io errors occured reading delivery log file
#define ERR_INSUFFICIENT_MEMORY                             1002
                    //insuficient memory to process 90th percentile report
```

```
#define ERR_CANNOT_OPEN_RESULTS_FILE    1005
          //Cannot open delivery results file delilog.

typedef struct _RPTLINE
{
          SYSTEMTIME          start;
                                                  //delilog report line start time
          SYSTEMTIME          end;
                                                  //delilog report line end time
          int                           response;
                                                  //delilog report line time
delivery took in milliseconds
          int                           w_id;
                                                            //delilog report line
warehouse id for delivery
          int                           o_carrier_id;
                                                  //delilog report line carier id
for delivery
          int                           items[10];
                                                            //delilog report line
delivery line items
} RPTLINE, *PRPTLINE;

//error message structure used in ErrorMessage API
typedef struct _SERRORMSG
{
          int                 iError;                             //error id of
message
          char      szMsg[80];                          //message to sent to
browser
} SERRORMSG;

int                                     versionMS = 3;
                              //delirpt version
int                                     versionMM = 0;
int                                     versionLS = 2;
int                                     iReport;
                              //delirpt report to process
int                                     iStartTime;
                                        //begin times to accept for report
int                                     iEndTime;
                              //end times to accept for report
FILE                          *fpLog;
                              //log file stream

//Local function prototypes
void                main(int argc, char *argv[]);
static int          Init(void);
static void         Restore(void);
static int          DoReport(void);
int                           AverageResponse(void);
int                           SkippedDelivery(void);
int                           Percentile90th(void);
BOOL                CheckTimes(PRPTLINE pRptLine);
static int          OpenLogFile(void);
static void         CloseLogFile(void);
static void         ResetLogFile(void);
static BOOL         LogEOF(int iOperation);
static BOOL         ReadReportLine(char *szBuffer, PRPTLINE pRptLine);
static BOOL         ParseReportLine(char *szLine, PRPTLINE pRptLine);
static BOOL         ParseDate(char *szDate, LPSYSTEMTIME pTime);
static BOOL         ParseTime(char *szTime, LPSYSTEMTIME pTime);
static void         ErrorMessage(int iError);
```

```
static BOOL         GetParameters(int argc, char *argv[]);
static void         PrintParameters(void);
static void         PrintHeader(void);
static void         cls(void);
static BOOL         IsNumeric(char *ptr);


/* FUNCTION: int main(int argc, char *argv[])
 *
 * PURPOSE:         This function is the beginning execution point for the delivery
executable.
 *
 * ARGUMENTS:       int                 argc      number of command line arguments
passed to delivery
 *                                      char      *argv[]   array of command line
argument pointers
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

void main(int argc, char *argv[])
{
          int       iError;

          PrintHeader();

          if ( GetParameters(argc, argv) )
          {
                    PrintParameters();
                    return;
          }

          if ( (iError=Init()) != ERR_SUCCESS )
          {
                    ErrorMessage(iError);
                    Restore();
                    return;
          }

          if ( (iError = DoReport()) != ERR_SUCCESS )
                    ErrorMessage(iError);

          Restore();

          return;
}

/* FUNCTION: static int Init(void)
 *
 * PURPOSE:         This function initializes the delirtp application.
 *
 * ARGUMENTS:       None
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

static int Init(void)
```

```
{
          int iError;

          if ( (iError = OpenLogFile()) )
                    return iError;
          return TRUE;
}

/* FUNCTION: static void Restore(void)
 *
 * PURPOSE:         This function cleans up the delirpt application before
termination.
 *
 * ARGUMENTS:       None
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

static void Restore(void)
{
          CloseLogFile();
          return;
}

/* FUNCTION: static int DoReport(void)
 *
 * PURPOSE:                   This function dispatches the requested report.
 *
 * ARGUMENTS:       None
 *
 * RETURNS:                   ERR_SUCCESS if successfull or error code if an error
occurs.
 *
 * COMMENTS:        None
 *
 */

static int DoReport(void)
{
          int iRc;

          switch(iReport)
          {
                    case 1:
                              iRc = AverageResponse();
                              break;
                    case 2:
                              iRc = Percentile90th();
                              break;
                    case 3:
                              iRc = SkippedDelivery();
                              break;
                    case 4:
                              if ( (iRc = AverageResponse()) != ERR_SUCCESS )
                                        break;
                              if ( (iRc = Percentile90th()) != ERR_SUCCESS )
                                        break;
                              if ( (iRc = SkippedDelivery()) != ERR_SUCCESS )
                                        break;
                              break;
```

```
          }
          return iRc;
}

/* FUNCTION: int AverageResponse(void)
 *
 * PURPOSE:                   This function processes the AverageResponse report.
 *
 * ARGUMENTS:       None
 *
 * RETURNS:                   ERR_SUCCESS if successfull or error code if an error
occurs.
 *
 * COMMENTS:        None
 *
 */

int AverageResponse(void)
{
          RPTLINE reportLine;
          int                 iTotalResponse;
          int                 iLines;
          double    fAverage;
          char      szDelivery[128];

          ResetLogFile();

          iTotalResponse = 0;
          iLines = 0;
          printf("\n\n******** Average Response Time Report *******\n");
          while ( !LogEOF(LOGFILE_READ_EOF) )
          {
                    if ( ReadReportLine(szDelivery, &reportLine) )
                              return ERR_READING_LOGFILE;
                    if ( szDelivery[0] == '*' )
                              continue;
                    if ( !LogEOF(LOGFILE_READ_EOF) )
                    {
                              if ( CheckTimes(&reportLine) )
                                        continue;
                              iLines++;
                              iTotalResponse += reportLine.response;

                              if ( iLines % 10 == 0 )
                                        printf("Reading Report Line:\t%d\r",
iLines);
                    }
          }
          printf("                                         \r");
          if ( iLines == 0 )
          {
                    printf("No deliveries found.\n");
          }
          else
          {
                    fAverage = ((double)iTotalResponse /
(double)iLines)/(double)1000;
                    printf("Total Deliveries:      %10.0f\n", (float)iLines);
                    printf("Total Response Times:  %10.3f\n",
((float)iTotalResponse/(float)1000));
                    printf("Average Response Time: %10.3f\n", fAverage);
          }
```

```
        return ERR_SUCCESS;
}

/* FUNCTION: int Percentile90th(void)
 *
 * PURPOSE:             This function processes the 90th percentile report.
 *
 * ARGUMENTS:   None
 *
 * RETURNS:             ERR_SUCCESS if successfull or error code if an error
occurs.
 *
 * COMMENTS:    This function requires enough space to allocate needed
 *                              buckets which will be 2 * max response time
in
 *                              deci-seconds.
 *
 */

int Percentile90th(void)
{
        RPTLINE reportLine;
        int             iBucketSize;
        int             i;
        int             iResponseSeconds;
        int             iMaxSeconds;
        int             iTotalBuckets;
        double  iTotal;
        double  i90thPercent;
        short   *psBuckets;
        char    szDelivery[128];

        printf("\n\n******** 90th Percentile *******\n");
        printf("Calculating Max Response Seconds...\n");

        ResetLogFile();

        iMaxSeconds = -1;
        while ( !LogEOF(LOGFILE_READ_EOF) )
        {
                if ( ReadReportLine(szDelivery, &reportLine) )
                        return ERR_READING_LOGFILE;
                if ( szDelivery[0] == '*' )
                        continue;
                if ( !LogEOF(LOGFILE_READ_EOF) )
                {
                        if ( iMaxSeconds < reportLine.response )
                                iMaxSeconds = reportLine.response;
                }
        }

        iTotalBuckets = iMaxSeconds + 1;

        printf("Allocating Buckets...\n");

        iBucketSize = iTotalBuckets * sizeof(short);

        if ( !(psBuckets = (short *)malloc(iBucketSize)) )
                return ERR_INSUFFICIENT_MEMORY;

        ZeroMemory(psBuckets, iBucketSize);

        iTotal = 0;
```

```
        ResetLogFile();
        printf("Calculating Distribution...\n");

        while ( !LogEOF(LOGFILE_READ_EOF) )
        {
                if ( ReadReportLine(szDelivery, &reportLine) )
                        return ERR_READING_LOGFILE;
                if ( szDelivery[0] == '*' )
                        continue;
                if ( !LogEOF(LOGFILE_READ_EOF) )
                {
                        if ( CheckTimes(&reportLine) )
                                continue;
                        psBuckets[reportLine.response]++;
                        iTotal++;
                }
        }

        i90thPercent = iTotal * .9;

        for(i=0, iTotal = 0.0; iTotal < i90thPercent; iTotal +=
(double)psBuckets[i] )
                i++;

        printf("90th Percentile = %d.%d\n", i/1000, (i % 1000));

        free(psBuckets);

        return ERR_SUCCESS;
}

/* FUNCTION: int SkippedDelivery(void)
 *
 * PURPOSE:             This function processes the Skipped Deliveries report.
 *
 * ARGUMENTS:   None
 *
 * RETURNS:             ERR_SUCCESS if successfull or error code if an error
occurs.
 *
 * COMMENTS:    None
 *
 */

int SkippedDelivery(void)
{
        RPTLINE reportLine;
        char    szDelivery[128];
        int             i;
        int             items[10];

        ResetLogFile();

        printf("\n\n******** Skipped Delivery Report *******\n");
        memset(items, 0, sizeof(items));
        printf("Reading Delivery Log File...");

        while ( !LogEOF(LOGFILE_READ_EOF) )
        {
                if ( ReadReportLine(szDelivery, &reportLine) )
                        return ERR_READING_LOGFILE;
                if ( szDelivery[0] == '*' )
```

```
                              continue;
                    if ( !LogEOF(LOGFILE_READ_EOF) )
                    {
                              if ( CheckTimes(&reportLine) )
                                        continue;
                              for(i=0; i<10; i++)
                              {
                                        if ( !reportLine.items[i] )
                                                  items[i]++;
                              }
                    }
          }
          printf("\n");
          printf("Skipped delivery table.\n");
          printf("  1    2    3    4    5    6    7    8    9   10 \n");
          printf("---- ---- ---- ---- ---- ---- ---- ---- ---- ----\n");
          for(i=0; i<10; i++)
                    printf("%4.4d ", items[i]);
          printf("\n");

          return ERR_SUCCESS;
}

/* FUNCTION: BOOL CheckTimes(PRPTLINE pRptLine)
 *
 * PURPOSE:         This function checks to see of the delilog record falls withing 
the
 *                            begin and end time from the command line.
 *
 * ARGUMENTS:       PRPTLINE  pRptLine  delilog processed report line.
 *
 * RETURNS:                   BOOL      FALSE     if report line is not within the
 *
          requested start and end times.
 *                                                          TRUE      if the report 
line is within the
 *
          requested start and end times.
 *
 * COMMENTS:        If startTime and endTime are both 0 then the user requested
 *                                      the default behavior which is all records in 
delilog are
 *                                      valid.
 */

BOOL CheckTimes(PRPTLINE pRptLine)
{
          int       iRptEndTime;
          int       iRptStartTime;

          iRptStartTime = (pRptLine->start.wHour * 3600000) + (pRptLine-
>start.wMinute * 60000) + (pRptLine->start.wSecond * 1000) + pRptLine-
>start.wMilliseconds;
          iRptEndTime = (pRptLine->end.wHour * 3600000) + (pRptLine->end.wMinute * 
60000) + (pRptLine->end.wSecond * 1000) + pRptLine->end.wMilliseconds;

          if ( iStartTime == 0 && iEndTime == 0 )
                    return FALSE;

          if ( iStartTime <= iRptStartTime && iEndTime >= iRptEndTime )
                    return FALSE;

          return TRUE;
```

```
}

/* FUNCTION: int OpenLogFile(void)
 *
 * PURPOSE:         This function opens the delivery log file for use.
 *
 * ARGUMENTS:       None
 *
 * RETURNS:                   int       ERR_CANNOT_OPEN_RESULTS_FILE  Cannot create 
results log file.
 *                                                ERR_SUCCESS
                                        Log file successfully opened
 *
 *
 * COMMENTS:        None
 *
 */

static int OpenLogFile(void)
{
          fpLog = fopen("delilog.", "rb");

          if ( !fpLog )
                    return ERR_CANNOT_OPEN_RESULTS_FILE;

          return ERR_SUCCESS;
}

/* FUNCTION: int CloseLogFile(void)
 *
 * PURPOSE:         This function closes the delivery log file.
 *
 * ARGUMENTS:       None
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

static void CloseLogFile(void)
{
          if ( fpLog )
                    fclose(fpLog);

          return;
}


/* FUNCTION: static void ResetLogFile(void)
 *
 * PURPOSE:         This function prepares the delilog. file for reading
 *
 * ARGUMENTS:       None
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

static void ResetLogFile(void)
{
```

```
          fseek(fpLog, 0L, SEEK_SET);
          LogEOF(LOGFILE_CLEAR_EOF);

          return;
}

/* FUNCTION: static BOOL LogEOF(int iOperation)
 *
 * PURPOSE:         This function tracks and reports the end of file condition
 *                            on the delilog file.
 *
 * ARGUMENTS:       int iOperation      requested operation this can be:
 *
          LOGFILE_READ_EOF    check log file flag return current state
 *
          LOGFILE_CLEAR_EOF   clear end of log file flag
 *
          LOGFILE_SET_EOF               set flag end of log file reached
 *
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

static BOOL LogEOF(int iOperation)
{
          static BOOL bEOF;

          switch(iOperation)
          {
                    case LOGFILE_READ_EOF:
                              return bEOF;
                              break;
                    case LOGFILE_CLEAR_EOF:
                              bEOF = FALSE;
                              break;
                    case LOGFILE_SET_EOF:
                              bEOF = TRUE;
                              break;
          }
          return FALSE;
}

/* FUNCTION: static BOOL ReadReportLine(char *szBuffer, PRPTLINE pRptLine)
 *
 * PURPOSE:         This function reads a text line from the delilog file.
 *                            on the delilog file.
 *
 * ARGUMENTS:       char                *szBuffer buffer to placed read delilog file
line into.
 *                                      PRPTLINE  pRptLine  returned structure
containing parsed delilog
 *
                    report line.
 *
 * RETURNS:                   FALSE     if successfull or TRUE if an error occurs.
 *
 * COMMENTS:        None
 *
 */
```

```
static BOOL ReadReportLine(char *szBuffer, PRPTLINE pRptLine)
{
          int i = 0;
          int ch;
          int iEof;

          while( i < 128 )
          {
                    ch = fgetc(fpLog);
                    if ( iEof = feof(fpLog) )
                              break;
                    if ( ch == ’\r’ )
                    {
                              if ( i )
                                        break;
                              continue;
                    }
                    if ( ch == ’\n’ )
                              continue;
                    szBuffer[i++] = ch;

          }

          //delivery item format is to long cannot be a valid delivery item
          if ( i >= 128 )
                    return TRUE;

          szBuffer[i] = 0;
          if ( iEof )
          {
                    LogEOF(LOGFILE_SET_EOF);
                    if ( i == 0 )
                              return FALSE;
          }
          if ( szBuffer[0] == ’*’ )
          {
                    //error line ignore
                    return FALSE;
          }
          return ParseReportLine(szBuffer, pRptLine);
}

/* FUNCTION: static BOOL ParseReportLine(char *szLine, PRPTLINE pRptLine)
 *
 * PURPOSE:         This function reads a text line from the delilog file.
 *                            on the delilog file.
 *
 * ARGUMENTS:       char                *szLine             buffer containing the
delilog file line to be parsed.
 *                                      PRPTLINE  pRptLine  returned structure
containing parsed delilog
 *
                    report line values.
 *
 * RETURNS:                   FALSE     if successfull or TRUE if an error occurs.
 *
 * COMMENTS:        None
 *
 */

static BOOL ParseReportLine(char *szLine, PRPTLINE pRptLine)
{
          int i;
          if ( ParseDate(szLine, &pRptLine->start) )
                    return TRUE;

          pRptLine->end.wYear = pRptLine->start.wYear;
          pRptLine->end.wMonth = pRptLine->start.wMonth;
          pRptLine->end.wDay = pRptLine->start.wDay;

          if ( !(szLine = strchr(szLine, ',')) )
                    return TRUE;
          szLine++;

          if ( ParseTime(szLine, &pRptLine->start) )
                    return TRUE;

          if ( !(szLine = strchr(szLine, ',')) )
                    return TRUE;
          szLine++;

          if ( ParseTime(szLine, &pRptLine->end) )
                    return TRUE;

          if ( !(szLine = strchr(szLine, ',')) )
                    return TRUE;
          szLine++;

          if ( !IsNumeric(szLine) )
                    return TRUE;
          pRptLine->response = atoi(szLine);

          if ( !(szLine = strchr(szLine, ',')) )
                    return TRUE;
          szLine++;

          if ( !IsNumeric(szLine) )
                    return TRUE;
          pRptLine->w_id = atoi(szLine);

          if ( !(szLine = strchr(szLine, ',')) )
                    return TRUE;
          szLine++;

          if ( !IsNumeric(szLine) )
                    return TRUE;
          pRptLine->o_carrier_id = atoi(szLine);

          if ( !(szLine = strchr(szLine, ',')) )
                    return TRUE;
          szLine++;

          for(i=0; i<10; i++)
          {
                    if ( !IsNumeric(szLine) )
                              return TRUE;
                    pRptLine->items[i] = atoi(szLine);

                    if ( i<9 && !(szLine = strchr(szLine, ',')) )
                              return TRUE;
                    szLine++;
          }

          return FALSE;
}
```

```
/* FUNCTION: static BOOL ParseDate(char *szDate, LPSYSTEMTIME pTime)
 *
 * PURPOSE:         This function validates and extracts a date string in the format
 *                            yy/mm/dd into an SYSTEMTIME structure.
 *
 * ARGUMENTS:       char                          *szDate            buffer
containing the date to be parsed.
 *                                      LPSYSTEMTIME       pTime
          system time structure where date will be placed.
 *
 * RETURNS:                   FALSE if successfull or TRUE if an error occurs.
 *
 * COMMENTS:        None
 *
 */

static BOOL ParseDate(char *szDate, LPSYSTEMTIME pTime)
{
          if ( !isdigit(*szDate) || !isdigit(*(szDate+1)) || *(szDate+2) != '/' ||
                     !isdigit(*(szDate+3)) || !isdigit(*(szDate+4)) || *(szDate+5)
!= '/' ||
                     !isdigit(*(szDate+6)) || !isdigit(*(szDate+7)) )
                     return TRUE;

          pTime->wYear = atoi(szDate);

          pTime->wMonth = atoi(szDate+3);

          pTime->wDay = atoi(szDate+6);

          if ( pTime->wMonth > 12 || pTime->wMonth < 0 || pTime->wDay > 31 || pTime-
>wDay < 0 )
                    return TRUE;

          return FALSE;
}

/* FUNCTION: static BOOL ParseTime(char *szTime, LPSYSTEMTIME pTime)
 *
 * PURPOSE:         This function validates and extracts a time string in the format
 *                            hh:mm:ss:mmm into an SYSTEMTIME structure.
 *
 * ARGUMENTS:       char                          *szTime            buffer
containing the time to be parsed.
 *                                      LPSYSTEMTIME       pTime
          system time structure where date will be placed.
 *
 * RETURNS:                   FALSE if successfull or TRUE if an error occurs.
 *
 * COMMENTS:        None
 *
 */

static BOOL ParseTime(char *szTime, LPSYSTEMTIME pTime)
{
          if ( !isdigit(*szTime) || !isdigit(*(szTime+1)) || *(szTime+2) != ':' ||
                    !isdigit(*(szTime+3)) || !isdigit(*(szTime+4)) || *(szTime+5) !=
':' ||
                     !isdigit(*(szTime+6)) || !isdigit(*(szTime+7)) || *(szTime+8)
!= ':' ||
                     !isdigit(*(szTime+9)) || !isdigit(*(szTime+10)) ||
!isdigit(*(szTime+11)) )
```

```
                     return TRUE;

          pTime->wHour = atoi(szTime);
          pTime->wMinute = atoi(szTime+3);
          pTime->wSecond = atoi(szTime+6);
          pTime->wMilliseconds  = atoi(szTime+9);

          if ( pTime->wHour > 23 || pTime->wHour < 0 ||
                    pTime->wMinute > 59 || pTime->wMinute < 0 ||
                    pTime->wSecond > 59 || pTime->wSecond < 0 ||
                    pTime->wMilliseconds < 0 )
                    return TRUE;

          if ( pTime->wMilliseconds > 999 )
          {
                    pTime->wSecond += (pTime->wMilliseconds/1000);
                    pTime->wMilliseconds = pTime->wMilliseconds  % 1000;
          }

          return FALSE;
}

/* FUNCTION: void ErrorMessage(int iError)
 *
 * PURPOSE:          This function displays an error message in the delivery
executable’s console window.
 *
 * ARGUMENTS:        int                 iError    error id to be displayed
 *
 * RETURNS:                    None
 *
 * COMMENTS:         None
 *
 */

static void ErrorMessage(int iError)
{
          int i;

          static SERRORMSG errorMsgs[] =
          {
                    {         ERR_SUCCESS,
                              "Success, no error."
                                                                                },
                    {         ERR_CANNOT_OPEN_RESULTS_FILE,
          "Cannot open delivery results file delilog."
                    },
                    {         ERR_READING_LOGFILE,
                    "Reading delivery log file, Delivery item format incorrect."
          },
                    {         ERR_INSUFFICIENT_MEMORY,
          "insufficient memory to process 90th percentile report."              },
                    {         0,
                                                  ""

                                                  }
          };

          for(i=0; errorMsgs[i].szMsg[0]; i++)
          {
                    if ( iError == errorMsgs[i].iError )
                    {
                              printf("\nError(%d): %s", iError, errorMsgs[i].szMsg);
```

```
                              return;
                    }
          }
          printf("Error(%d): %s", errorMsgs[0].szMsg);
          return;
}

/* FUNCTION: BOOL GetParameters(int argc, char *argv[])
 *
 * PURPOSE:          This function parses the command line passed in to the delivery
executable, initializing
 *                             and filling in global variable parameters.
 *
 * ARGUMENTS:        int                 argc      number of command line arguments
passed to delivery
 *                                       char      *argv[]   array of command line
argument pointers
 *
 * RETURNS:                    BOOL      FALSE     parameter read successfull
 *                                                           TRUE      user has
requested parameter information screen be displayed.
 *
 * COMMENTS:         None
 *
 */

static BOOL GetParameters(int argc, char *argv[])
{
          int                           i;
          SYSTEMTIME          startTime;
          SYSTEMTIME          endTime;

          iStartTime = 0;
          iEndTime = 0;
          iReport = 4;

          for(i=0; i<argc; i++)
          {
                    if ( argv[i][0] == ’-’ || argv[i][0] == ’/’ )
                    {
                              switch(argv[i][1])
                              {
                                        case ’S’:
                                        case ’s’:
                                                  if ( ParseTime(argv[i]+2,
&startTime) )
                                                            return TRUE;
                                                  iStartTime = (startTime.wHour *
3600000) + (startTime.wMinute * 60000) + (startTime.wSecond * 1000) +
startTime.wMilliseconds;
                                                  break;
                                        case ’E’:
                                        case ’e’:
                                                  if ( ParseTime(argv[i]+2,
&endTime) )
                                                            return TRUE;
                                                  iEndTime = (endTime.wHour *
3600000) + (endTime.wMinute * 60000) + (endTime.wSecond * 1000) +
endTime.wMilliseconds;
                                                  break;
                                        case ’R’:
                                        case ’r’:
                                                  iReport = atoi(argv[i]+2);
                                                  if ( iReport > 4 || iReport < 1 )
                                                            iReport = 4;
                                                  break;
                                        case ’?’:
                                                  return TRUE;
                              }
                    }
          }
          return FALSE;
}

/* FUNCTION: void PrintParameters(void)
 *
 * PURPOSE:         This function displays the supported command line flags.
 *
 * ARGUMENTS:       None
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

static void PrintParameters(void)
{
          PrintHeader();
          printf("DELIRPT:\n\n");
          printf("Parameter
Default\n");
          printf("-----------------------------------------------------------------
-----\n");
          printf("-S Start Time HH:MM:SS:MMM
All    \n");
          printf("-E End Time HH:MM:SS:MMM
All    \n");
          printf("-R 1)Average Response, 2)90th 3) Skipped 4) All
All    \n");
          printf("-? This help screen\n\n");
          printf("Note:  Command line switches are NOT case sensitive.\n");

          return;
}

/* FUNCTION: void PrintHeader(void)
 *
 * PURPOSE:         This function displays the delivery report applications banner
information.
 *
 * ARGUMENTS:       None
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

static void PrintHeader(void)
{
          cls();

          printf("**************************************************\n");
          printf("*                                                *\n");
          printf("*  Microsoft SQL Server 6.5                      *\n");
```

```
          printf("*                                                *\n");
          printf("*  HTML TPC-C BENCHMARK KIT: Delivery Report     *\n");
          printf("*  Version %d.%2.2d.%3.3d                              *\n",
versionMS, versionMM, versionLS);
          printf("*                                                *\n");
          printf("**************************************************\n\n");

          return;
}

/* FUNCTION: void cls(void)
 *
 * PURPOSE:         This function clears the console window
 *
 * ARGUMENTS:       None
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

static void cls(void)
{
          HANDLE    hConsole;
          COORD     coordScreen = { 0, 0 };                           //here’s where
we’ll home the cursor
          DWORD     cCharsWritten;
          CONSOLE_SCREEN_BUFFER_INFO    csbi;               //to get buffer info
          DWORD                                                       dwConSize;
          //number of character cells in the current buffer

          hConsole = GetStdHandle(STD_OUTPUT_HANDLE);

          //get the number of character cells in the current buffer

          GetConsoleScreenBufferInfo( hConsole, &csbi );
          dwConSize = csbi.dwSize.X * csbi.dwSize.Y;

          //fill the entire screen with blanks
          FillConsoleOutputCharacter( hConsole, (TCHAR) ’ ’, dwConSize, coordScreen,
&cCharsWritten );
          GetConsoleScreenBufferInfo( hConsole, &csbi );

          //now set the buffer’s attributes accordingly
          FillConsoleOutputAttribute( hConsole, csbi.wAttributes,dwConSize,
coordScreen, &cCharsWritten );

          //put the cursor at (0, 0)
          SetConsoleCursorPosition( hConsole, coordScreen );

          return;
}

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

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

          while( *ptr && isdigit(*ptr) )
                    ptr++;
          if ( !*ptr || *ptr == ',' )
                    return TRUE;
          else
                    return FALSE;
}
```

DELIVERY.C

```
/*        FILE:               DELIVERY.C
 *
 *        Based on: Microsoft TPC-C Kit Ver. 3.00.000
 *
 *                                      Copyright Microsoft, 1996
 *                                      Copyright Performance Tuning Corporation,
1997
 *
 *        PURPOSE:  New Order Tuxedo Server.
 *        Author:             Philip Durr
 *                                      philipdu@Microsoft.com
 *
 *  MODIFIED        Changed for modularity and to allow for the Tuxedo TM
 *
 *  Author:                   Edward Whalen
 *                                      Performance Tuning Corporation
 *                                      ewhalen@perftuning.com
 */

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

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

#include "trans.h"                                                //tpckit
transaction header contains definations of structures specific to TPC-C
#include "httpext.h"                                              //ISAPI DLL
information header

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

#include <tmenv.h>
#include <xa.h>
#include <atmi.h>

BOOL      bLog                          = FALSE;
BOOL                                              bFlush;
                                        //Flush delivery log info when written.
BOOL      verbose                       = FALSE;
BOOL      bError                        = FALSE;

int                 iThreads            = 5;
int                 iMaxWareHouses      = 500;
int                 iDelayMs            = 100;
short     iMaxConnections = (short)1;
short     iDeadlockRetry      = (short)3;

DBPROCESS *pdbproc;

char                                              szServer[32];
                    //SQL server name
char                                              szDatabase[32];
                    //tpcc database name
char                                              szUser[32];
                              //user name
char                                              szPassword[32];
                    //user password
int spId;


TERM Term;

static char         szTpccLogPath[256]; //path to html log file if logging turned on
in registry.
static char szErrorLogPath[256];        //path to error log file.

static    CRITICAL_SECTION              CriticalSection;
static    CRITICAL_SECTION              ErrorLogCriticalSection;
static    EXTENSION_CONTROL_BLOCK       *gpECB = NULL;
static    int                                                       bTpccExit;
          //exit delivery disconnect loop as dll exiting.

extern void TMLog();
extern BOOL SQLInit();
extern void UtilStrCpy();
extern void UtilStrCpy();
extern BOOL SQLOpenConnection();
extern BOOL SQLCloseConnection();

//BOOL                                            bDone;
                                        //delivery executable termination request
flag
BOOL                                              bFlush;
                                        //Flush delivery log info when written.

#define ERR_CANNOT_CREATE_THREAD                  1000      //Cannot create thread.
#define ERR_DBGETDATA_FAILED                      1001      //Get data failed.
```

```
#define   ERR_REGISTRY_NOT_SETUP                            1002      //Registry not
setup for tpcc.
#define   ERR_CANNOT_ACCESS_DELIVERY_FN 1003      //Cannot access ReadDelivery
cache.
#define   ERR_CANNOT_ACCESS_REGISTRY              1004      //Cannot access registry
key TPCC.
#define   ERR_CANNOT_CREATE_RESULTS_FILE          1005      //Cannot create results
file.

FILE      *fpLog;

/* FUNCTION: tpsvrinit ( int argc, char *argv[])
 *
 * PURPOSE:         Initialize the Server to Database connection.
 *
 * RETURNS:                   int       0         Success
 *                                                -1        Failure
 *
 *
 * COMMENTS:        None
 *
 */

int tpsvrinit ( int argc, char *argv[] )
{

          if ( GetParameters(argc, argv) )
          {
                    PrintParameters();
                    return -1;
          }

          if ( verbose )
                    TMLog("TPSVRINIT: Delivery: Server %s, Database %s, User %s,
Password %s, Flush %d.",
                              szServer, szDatabase, szUser, szPassword, bFlush);

          if ( ! SQLInit() )
          {
                    TMLog( "DELIVERY: SQLInit Failed" );
                    return -1;
          }

          if ( SQLOpenConnection ( NULL, 0, 0, &pdbproc, szServer, szDatabase,
szUser, szPassword, szDatabase, &spId))
          {
                    TMLog ( "DELIVERY: SQLOpenConnection Failed" );
                    dbexit();
                    return -1;
          }

          OpenLogFile();

          return 0;
}

/* FUNCTION: tpsvrdone ( void )
 *
 * PURPOSE:         Initialize the Server to Database connection.
 *
 * RETURNS:                   int       0         Success
 *                                                -1        Failure
 *
```

```
 *
 * COMMENTS:        None
 *
 */

void tpsvrdone ( void )
{
          SQLCloseConnection( NULL, pdbproc);
          dbexit();
}

/* FUNCTION: DELIVERY ( TPSVCINFO *rqst )
 *
 * PURPOSE:         Process a New Order request.
 *
 * RETURNS:                   int       0         Success
 *                                                -1        Failure
 *
 *
 * COMMENTS:        None
 *
 */

void DELIVERY ( TPSVCINFO *rqst )
{
          PECBINFO pECBInfo = dbgetuserdata(pdbproc);
          DELIVERY_DATA *dp;

          int size = rqst->len;

          if (verbose)
                    TMLog(" DELIVERY: Begin transaction");

          dp = (DELIVERY_DATA *) rqst->data ;

          if (verbose )
          {
                    TMLog(" DELIVERY: w_id %d ", dp->w_id);
                    TMLog(" DELIVERY: d_id %d ", dp->o_carrier_id);
          }

          bError = FALSE;

          dp->retval = SQLDelivery( pdbproc, dp, iDeadlockRetry);

          if (bError == TRUE)
                    dp->retval = -1;

          tpreturn( TPSUCCESS, 0, rqst->data, size, 0);
}


/* FUNCTION: void CalculateElapsedTime(int *pElapsed, LPSYSTEMTIME lpBegin,
LPSYSTEMTIME lpEnd)
 *
 * PURPOSE:         This function calculates the elapsed time a delivery transaction
took.
 *
 * ARGUMENTS:       int                                     *pElapsed pointer to int
variable to receive calculated elapsed
 *
                              time in milliseconds.
 *                                       LPSYSTEMTIME        lpBegin
          Pointer to system time structure containing
 *
                              transaction beginning time.
 *                                       LPSYSTEMTIME        lpEnd
          Pointer to system time structure containing
 *
                              transaction ending time.
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

static void CalculateElapsedTime(int *pElapsed, LPSYSTEMTIME lpBegin, LPSYSTEMTIME
lpEnd)
{
          int                            beginSeconds;
          int                            endSeconds;


          beginSeconds = (lpBegin->wHour * 3600000) + (lpBegin->wMinute * 60000) +
(lpBegin->wSecond * 1000) + lpBegin->wMilliseconds;
          endSeconds = (lpEnd->wHour * 3600000) + (lpEnd->wMinute * 60000) + (lpEnd-
>wSecond * 1000) + lpEnd->wMilliseconds;
          *pElapsed = endSeconds - beginSeconds;

          //check for day boundry, this will function for 24 hour period however it
will not work over 48 hours.
          if ( *pElapsed < 0 )
                    *pElapsed = *pElapsed + (24 * 60 * 60 * 1000);

          return;
}

/* FUNCTION: int SQLDelivery(DBPROCESS *dbproc, DELIVERY *pDelivery, short
deadlock_retry )
 *
 * PURPOSE:         This function processes the delivery transaction.
 *
 * ARGUMENTS:       DELIVERY            *pDelivery                    Pointer to
delivery transaction structure
 *
 * RETURNS:                   int       ERR_DBGETDATA_FAILED
          Delivery get data operation failed.
 *                                                 ERR_SUCCESS
                              Delivery successfull, no error
 *
 *
 * COMMENTS:        None
 *
 */

static int SQLDelivery(DBPROCESS *dbproc, DELIVERY_DATA *pDelivery, short
deadlock_retry)
{
          RETCODE   rc;
          int                 i;
          int                 deadlock_count;
          BYTE      *pData;
          SYSTEMTIME          trans_end;                    //delivery transaction
finished time
```

```
          int       elapsed;                                //delivery transaction
time

          deadlock_count = 0;


          // Start new delivery
          while ( TRUE )
          {
                    if (dbrpcinit(dbproc, "tpcc_delivery", 0) == SUCCEED)
                    {
                              dbrpcparam(dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE
*)&pDelivery->w_id);
                              dbrpcparam(dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE
*)&pDelivery->o_carrier_id);

                              if (dbrpcexec(dbproc) == SUCCEED)
                              {
                                        while (((rc = dbresults(dbproc)) !=
NO_MORE_RESULTS) && (rc != FAIL))
                                        {
                                                  while (((rc = dbnextrow(dbproc))
!= NO_MORE_ROWS) && (rc != FAIL))
                                                  {
                                                            for (i=0;i<10;i++)
                                                            {

          if(pData=dbdata(dbproc, i+1))

          pDelivery->o_id[i] = *((DBINT *)pData);
                                                                      else

          pDelivery->o_id[i] = 0;
                                                            }
                                                  }
                                        }
                              }
                    }
                    if ( !SQLDetectDeadlock(dbproc) )
                              break;
                    deadlock_count++;
                    Sleep(10 * deadlock_count);
          }
          GetLocalTime(&trans_end);

          CalculateElapsedTime(&elapsed, &pDelivery->queue_time, &trans_end);

          fprintf(fpLog,
"%2.2d/%2.2d/%2.2d,%2.2d:%2.2d:%2.2d:%3.3d,%2.2d:%2.2d:%2.2d:%3.3d,%d,%d,%d,%d,%d,%d
,%d,%d,%d,%d,%d,%d,%d\r\n",
                    trans_end.wYear - 1900, trans_end.wMonth, trans_end.wDay,
                    pDelivery->queue_time.wHour, pDelivery->queue_time.wMinute,
                    pDelivery->queue_time.wSecond, pDelivery-
>queue_time.wMilliseconds,
                    trans_end.wHour, trans_end.wMinute, trans_end.wSecond,
trans_end.wMilliseconds,
                    elapsed,
                    pDelivery->w_id, pDelivery->o_carrier_id,
                    pDelivery->o_id[0], pDelivery->o_id[1], pDelivery->o_id[2],
pDelivery->o_id[3],
                    pDelivery->o_id[4], pDelivery->o_id[5], pDelivery->o_id[6],
pDelivery->o_id[7],
                    pDelivery->o_id[8], pDelivery->o_id[9] );
```

```
        if ( bFlush )
                fflush(fpLog);


        return ERR_SUCCESS;
}

/* FUNCTION: BOOL SQLDetectDeadlock(DBPROCESS *dbproc)
 *
 * PURPOSE:       This function is used to check for deadlock conditions.
 *
 * ARGUMENTS:     DBPROCESS       *dbproc         DBPROCESS to check
 *
 * RETURNS:               BOOL    FALSE                                   No
lock condition present
 *                                                TRUE
                Lock condition detected
 *
 * COMMENTS:      None
 *
 */

static BOOL SQLDetectDeadlock(DBPROCESS *dbproc)
{
        if (*((BOOL *) dbgetuserdata(dbproc)) == TRUE)
    {
                *((BOOL *) dbgetuserdata(dbproc)) = FALSE;
                return TRUE;
        }
        return FALSE;
}

/* FUNCTION: int OpenLogFile(void)
 *
 * PURPOSE:       This function opens the delivery log file for use.
 *
 * ARGUMENTS:     None
 *
 * RETURNS:               int     ERR_REGISTRY_NOT_SETUP
        Registry not setup.
 *                                        ERR_CANNOT_CREATE_RESULTS_FILE
        Cannot create results log file.
 *                                        ERR_SUCCESS
                                Log file successfully opened
 *
 *
 * COMMENTS:      None
 *
 */

static int OpenLogFile(void)
{
        HKEY    hKey;
        BOOL    bRc;
        BYTE    szTmp[256];
        char    szKey[256];
        char    szLogPath[256];
        DWORD   size;
        DWORD   sv;
        int             len;
        char    *ptr;
```

```
        szLogPath[0] =  0;
        bRc = TRUE;
        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\Virtual Roots", 0,
KEY_ALL_ACCESS, &hKey) == ERROR_SUCCESS )
        {
                sv = sizeof(szKey);
                size = sizeof(szTmp);

                if ( RegEnumValue(hKey, 0, szKey, &sv, NULL, NULL, szTmp, &size)
== ERROR_SUCCESS )
                {
                        strcpy(szLogPath, szTmp);
                        bRc = FALSE;
                }
                RegCloseKey(hKey);
        }

        if ( bRc )
                return ERR_REGISTRY_NOT_SETUP;

        if ( (ptr = strchr(szLogPath, ',')) )
                *ptr = 0;

        len = strlen(szLogPath);
        if ( szLogPath[len-1] != '\\' )
        {
                szLogPath[len] = '\\';
                szLogPath[len+1] = 0;
        }
        sprintf(szLogPath, "delilog%d", GetCurrentThreadId());

        fpLog = fopen(szLogPath, "ab");

        if ( !fpLog )
                return ERR_CANNOT_CREATE_RESULTS_FILE;

        return ERR_SUCCESS;
}

/*
 *      Common Code for all Servers
 */

 /* FUNCTION: BOOL SQLInit()
 *
 * PURPOSE:       This function initializes SQL Server for later use.
 *
 *
 * RETURNS:               BOOL    FALSE   if successfull
 *                                                TRUE    if an error
occurs and connection cannot be established.
 *
 * COMMENTS:      None
 *
 */
BOOL SQLInit ()
{
        dbinit();

        if ( dbgetmaxprocs() < iMaxConnections )
        {
                if ( dbsetmaxprocs(iMaxConnections) == FAIL )
```

```
                    {
                              //set for fail error message when HttpExtensionProc()
is called because
                              //at this point we don't have a pECB so no way to show
error message.
                              iMaxConnections = -1;
                    }
          }

          // install error and message handlers
          dbmsghandle((DBMSGHANDLE_PROC)msg_handler);
          dberrhandle((DBERRHANDLE_PROC)err_handler);

          return TRUE;
}


/* FUNCTION: BOOL SQLOpenConnection(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId, DBPROCESS **dbproc, char *server, char *database, char *user, char
*password, char *app, int *spid, long *pack_size)
 *
 * PURPOSE:         This function opens the sql connection for use.
 *
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB     passed in structure
pointer from inetsrv.
 *                                   int
          iTermId             terminal id of browser
 *                                   int
          iSyncId             sync id of browser
 *                                   DBPROCESS           **dbproc  pointer to
returned DBPROCESS
 *                                   char                          *server
          SQL server name
 *                                   char                          *database SQL
server database
 *                                   char                          *user
          user name
 *                                   char                          *password user
password
 *                                   char                          *app
          pointer to returned application array
 *                                   int
          *spid               pointer to returned spid
 *                                   long                          *pack_size
          pointer to returned default pack size
 *
 * RETURNS:                    BOOL      FALSE     if successfull
 *                                                           TRUE      if an error
occurs
 *
 * COMMENTS:        None
 *
 */


#ifdef USE_ODBC
          static BOOL SQLOpenConnection(EXTENSION_CONTROL_BLOCK *pECB, int iTermId,
int iSyncId, DBPROCESS **dbproc, char *server, char *database, char *user, char
*password, char *app, int *spid, long *pack_size)
          {

                    RETCODE   rc;
                    char      buffer[30];
```

```
                    *dbproc = (DBPROCESS *)malloc(sizeof(DBPROCESS));
                    if ( !*dbproc )
                              return TRUE;

                    //set pECB data into dbproc
                    (*dbproc)->bDeadlock = FALSE;
                    (*dbproc)->bFailed = FALSE;
                    (*dbproc)->pECB = pECB;
                    (*dbproc)->iTermId = iTermId;
                    (*dbproc)->iSyncId = iSyncId;


                    if ( SQLAllocConnect(henv, &(*dbproc)->hdbc) == SQL_ERROR )
                              return TRUE;

                    if ( SQLSetConnectOption((*dbproc)->hdbc, SQL_PACKET_SIZE,
pack_size) == SQL_ERROR )
                              return TRUE;

                    rc = SQLConnect((*dbproc)->hdbc, server, SQL_NTS, user, SQL_NTS,
password, SQL_NTS);
                    if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                              return TRUE;
                    rc = SQLAllocStmt((*dbproc)->hdbc, &(*dbproc)->hstmt);
                    if (rc == SQL_ERROR)
                              return TRUE;

                    sprintf(buffer,"use %s", Client->database);

                    rc = SQLExecDirect((*dbproc)->hstmt, buffer, SQL_NTS);
                    if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                              return TRUE;

                    SQLFreeStmt((*dbproc)->hstmt, SQL_CLOSE);
                    sprintf(buffer,"set nocount on");
                    rc = SQLExecDirect((*dbproc)->hstmt, buffer, SQL_NTS);
                    if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                              return TRUE;
                    SQLFreeStmt((*dbproc)->hstmt, SQL_CLOSE);

                    sprintf(buffer,"select @@spid");

                    rc = SQLExecDirect((*dbproc)->hstmt, buffer, SQL_NTS);
                    if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                              return TRUE;

                    if ( SQLBindCol((*dbproc)->hstmt, 1, SQL_C_SSHORT, &(*dbproc)-
>spid, 0, NULL) == SQL_ERROR )
                              return TRUE;

                    if ( SQLFetch((*dbproc)->hstmt) == SQL_ERROR )
                              return TRUE;

                    SQLFreeStmt((*dbproc)->hstmt, SQL_CLOSE);

                    return FALSE;
          }

#else
```

```
          static BOOL SQLOpenConnection(EXTENSION_CONTROL_BLOCK *pECB, int iTermId,
int iSyncId, DBPROCESS **dbproc, char *server, char *database, char *user, char
*password, char *app, int *spid)
          {
                    LOGINREC  *login;
                    PECBINFO  pEcbInfo;

                    //set local msg proc for login record
                    //attach pECB record

                    //this is necessary as dblib provides no way to pass user data
in a login structure. So until
                    //there is an allocated dbproc we need to use a static which
means that the login attempt must
                    //be serialized.

                    gpECB = pECB;

                    login = dblogin();
                    if ( !*user )
                              DBSETLUSER(login, "sa");
                    else
                              DBSETLUSER(login, user);

                    DBSETLPWD(login,  password);
                    DBSETLHOST(login, app);

                    DBSETLPACKET(login, (unsigned short)DEFCLPACKSIZE);

                    if ((*dbproc = dbopen(login, server )) == NULL)
                              return TRUE;

                    //set pECB data into dbproc
                    pEcbInfo = (PECBINFO)malloc(sizeof(ECBINFO));
                    pEcbInfo->bDeadlock = FALSE;
                    pEcbInfo->pECB = pECB;
                    pEcbInfo->iTermId = iTermId;
                    pEcbInfo->iSyncId = iSyncId;
                    dbsetuserdata(*dbproc, pEcbInfo);

                    // Use the the right database
                    dbuse(*dbproc, database);

                    dbcmd(*dbproc, "select @@spid");

                    dbsqlexec(*dbproc);
                    while (dbresults(*dbproc) != NO_MORE_RESULTS)
                    {
                              dbbind(*dbproc, 1, SMALLBIND, (DBINT) 0, (BYTE *)
spid);
                              while (dbnextrow(*dbproc) != NO_MORE_ROWS)
                              ;
                    }
                    dbcmd(*dbproc, "set nocount on");

                    dbsqlexec(*dbproc);
                    while (dbresults(*dbproc) != NO_MORE_RESULTS)
                    {
                              while (dbnextrow(*dbproc) != NO_MORE_ROWS)
                              ;
                    }

                    //rollback transaction on abort
```

```
                    dbcmd(*dbproc, "set XACT_ABORT ON");

                    dbsqlexec(*dbproc);
                    while (dbresults(*dbproc) != NO_MORE_RESULTS)
                    {
                              while (dbnextrow(*dbproc) != NO_MORE_ROWS)
                              ;
                    }

                    return FALSE;
          }

#endif

/* FUNCTION: BOOL SQLCloseConnection(EXTENSION_CONTROL_BLOCK *pECB, DBPROCESS
*dbproc)
 *
 * PURPOSE:         This function closes the sql connection.
 *
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB     passed in structure
pointer from inetsrv.
 *                                      DBPROCESS           *dbproc   pointer to
DBPROCESS
 *
 * RETURNS:                   BOOL      FALSE     if successfull
 *                                                          TRUE      if an error
occurs
 *
 * COMMENTS:        None
 *
 */

#ifdef USE_ODBC
          static BOOL SQLCloseConnection(EXTENSION_CONTROL_BLOCK *pECB, DBPROCESS
*dbproc)
          {
                    if ( dbproc )
                    {
                              SQLFreeStmt(dbproc->hstmt, SQL_DROP);
                              SQLDisconnect(dbproc->hdbc);
                              SQLFreeConnect(dbproc->hdbc);
                              free(dbproc);
                              dbproc = NULL;
                    }
                    return FALSE;
          }
#else
          static BOOL SQLCloseConnection(EXTENSION_CONTROL_BLOCK *pECB, DBPROCESS
*dbproc)
          {
                    if (dbclose(dbproc) == FAIL)
                              return TRUE;
                    return FALSE;
          }
#endif

// Lifted from HP FDR since they did such a nice job
void TMLog( char *format, ... )
{
          va_list args;
          char buf[4096];
          int len;
          va_start( args, format );
          _strtime( buf );
          strcat( buf," ");
          len = strlen( buf );
          (void)_vsnprintf( buf+ len, sizeof( buf) - len - 1, format, args);
          buf[sizeof( buf )- 1]= '\0';
          va_end( args );
          userlog( buf );
}

/* 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.
 *
 */

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

          return;
}

/* FUNCTION: int err_handler(DBPROCESS *dbproc, int severity, int dberr, int oserr,
char *dberrstr, char *oserrstr)
 *
 * PURPOSE:         This function handles DB-Library errors
 *
 * ARGUMENTS:       DBPROCESS           *dbproc                       DBPROCESS id
pointer
 *                                      int
          severity            severity of error
 *                                      int
          dberr                         error id
 *                                      int
          oserr                         operating system specific error code
 *                                      char                          *dberrstr
          printable error description of dberr
 *                                      char                          *oserrstr
          printable error description of oserr
 *
 * RETURNS:                   int                                     INT_CONTINUE
          continue if error is SQLETIME else INT_CANCEL action
 *
 * COMMENTS:        None
 *
 */
```

```
int err_handler(DBPROCESS *dbproc, int severity, int dberr, int oserr, char
*dberrstr, char *oserrstr)
{
          PECBINFO                                pEcbInfo;
          EXTENSION_CONTROL_BLOCK       *pECB;
          FILE                                              *fp;
          SYSTEMTIME                                        systemTime;
          char                                              szTmp[256];
          int                                                         iTermId;
          int                                                         iSyncId;

          pEcbInfo = NULL;

          if ((dbproc == NULL) || (DBDEAD(dbproc)))
          {
                    TMLog("DBPROC is invalid");
                    return INT_CANCEL;
          }

          if ( !(pEcbInfo = (PECBINFO)dbgetuserdata(dbproc)) )
          {
                    pECB = gpECB;
                    iTermId = 0;
                    iSyncId = 0;
          }
          else
          {
                    pECB = pEcbInfo->pECB;
                    iTermId = pEcbInfo->iTermId;
                    iSyncId = pEcbInfo->iSyncId;
          }

          if ( pEcbInfo && pEcbInfo->bFailed )
          {
                    bError == FALSE;
                    return INT_CANCEL;
          }

          if ( oserr != DBNOERR )
          {
                    TMLog("DBLIB Error %s", oserrstr);
                    if ( pEcbInfo )
                    {
                              pEcbInfo->bFailed = TRUE;
                              bError = TRUE;
                    }

                    GetLocalTime(&systemTime);
                    fp = fopen(szErrorLogPath, "ab");

                    sprintf(szTmp, "ErrorHandler: DBLIB(%d): %s", oserr, oserrstr);

                    TMLog("%2.2d/%2.2d/%2.2d %2.2d:%2.2d:%2.2d\r\n\r\n%s\r\n\r\n",
                              systemTime.wYear, systemTime.wMonth, systemTime.wDay,
                              systemTime.wHour, systemTime.wMinute,
systemTime.wSecond,
                              szTmp);

                    fclose(fp);
          }

          return INT_CANCEL;
}
```

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

int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int severity, char
*msgtext)
{
          PECBINFO                                pEcbInfo;
          EXTENSION_CONTROL_BLOCK       *pECB;
          FILE                                              *fp;
          SYSTEMTIME                                        systemTime;
          char                                              szTmp[256];
          int                                                         iTermId;
          int                                                         iSyncId;

          if ( !(pEcbInfo = (PECBINFO)dbgetuserdata(dbproc)) )
          {
                    pECB = gpECB;
                    iTermId = 0;
                    iSyncId = 0;
          }
          else
          {
                    pECB = pEcbInfo->pECB;
                    iTermId = pEcbInfo->iTermId;
                    iSyncId = pEcbInfo->iSyncId;
          }


          if ( (msgno == 5701) || (msgno == 2528) || (msgno == 5703) || (msgno ==
6006) )
                    return INT_CONTINUE;

          // deadlock message
          if (msgno == 1205)
          {
                    // set the deadlock indicator
                    if ( pEcbInfo )
                              pEcbInfo->bDeadlock = TRUE;
                    else
                              TMLog("Error, dbgetuserdata returned NULL.");
                    return INT_CONTINUE;
          }
          if ( pEcbInfo && pEcbInfo->bFailed )
          {
                    TMLog("SQL Error ");
                    return INT_CANCEL;
          }

          if (msgno == 0)
                    return INT_CONTINUE;
          else
          {
                    TMLog("MsgHandler: SQL Error %s", msgtext);

                    if ( pEcbInfo )
                              pEcbInfo->bFailed = TRUE;

                    bError = TRUE;

                    sprintf(szTmp, "Error: SQLSVR(%d): %s", msgno, msgtext);

                    TMLog("%2.2d/%2.2d/%2.2d %2.2d:%2.2d:%2.2d\r\n\r\n%s\r\n\r\n",
                              systemTime.wYear, systemTime.wMonth, systemTime.wDay,
                              systemTime.wHour, systemTime.wMinute,
systemTime.wSecond,
                              szTmp);
          }
          return INT_CANCEL;
}


/* FUNCTION: BOOL GetParameters(int argc, char *argv[])
 *
 * PURPOSE:         This function parses the command line passed in to the delivery
executable, initializing
 *                            and filling in global variable parameters.
 *
 * ARGUMENTS:       int                 argc      number of command line arguments
passed to delivery
 *                                      char      *argv[]   array of command line
argument pointers
 *
 * RETURNS:                   BOOL      FALSE     parameter read successfull
 *                                                          TRUE      user has
requested parameter information screen be displayed.
 *
 * COMMENTS:        None
 *
 */

static BOOL GetParameters(int argc, char *argv[])
{
          int i;

          szServer[0]                   = 0;
          szPassword[0]       = 0;
          bFlush                        = FALSE;
          strcpy(szDatabase, "tpcc");
          strcpy(szUser, "sa");

          for(i=0; i<argc; i++)
          {
                    if ( argv[i][0] == ’-’ || argv[i][0] == ’/’ )
                    {
                              switch(argv[i][1])
                              {
                                        case ’S’:
                                        case ’s’:
                                                  strcpy(szServer, argv[i]+2);
                                                  break;
                                        case ’V’:
                                        case ’v’:
                                                  verbose = TRUE;
                                                  break;
                                        case ’F’:
                                        case ’f’:
                                                  bFlush = TRUE;      //turn on
delilog flush when written.
                                                  break;
                                        case ’?’:
                                                  return TRUE;
                              }
                    }
          }
          return FALSE;
}

/* FUNCTION: void PrintParameters(void)
 *
 * PURPOSE:         This function displays the supported command line flags.
 *
 * ARGUMENTS:       None
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

static void PrintParameters(void)
{
          TMLog("Performance Tuning Corporation Tuxedo Kit");
          TMLog("   www.perftuning.com   (281) 251-3495  ");
          TMLog("Delivery: -S Server -D Database -U User -P Password -F" );
          TMLog("Delivery: Server %s, Database %s, User %s, Password %s, Flush %d.",
                    szServer, szDatabase, szUser, szPassword, bFlush);

}
```

GETVAL.H

```
/* MACRO: GetVal(char *pQueryString, char *pKey, char *pValue, int iMax)
 *
 * PURPOSE:         This function parses a http formatted string for specific key
values.
 *
 * ARGUMENTS:       char                          *pQueryString       http string
from client browser
 *                                      char                          *pKey
                    key value to look for
 *                                      char                          *pValue
                    character array into which to place key’s value
 *                                      int                                    iMax
                              maximum length of key value array.
 *
 * RETURNS:                   BOOL      FALSE     key value not found
 *                                                          TRUE      key valud
found
 *
 *
 * COMMENTS:        http keys are formatted either KEY=value& or KEY=value\0. This
DLL formats
 *                                      TPC-C input fields in such a manner that the
keys can be extracted in the
 *                                      above manner.
 */

#define GetVal(s, k, v, m) \
{ int max = m; char * vp = v;\
          if ( (s=strstr(s, k)) ) \
          if ( (s=strchr(s, ’=’)) ) \
          { \
            s++; \
            max--; \
            while( *s && *s != ’&’ && max) \
            { \
                    *vp++ = *s++; \
                    max--; \
            } \
            *vp = 0; \
            s-- ; \
        }\
}
```

HTTPEXT.H

```
/********
*
*  Copyright (c) 1995  Process Software Corporation
*
*  Copyright (c) 1995  Microsoft Corporation
*
*
*  Module Name  : HttpExt.h
*
*  Abstract :
*
*     This module contains  the structure definitions and prototypes for the
*     version 1.0 HTTP Server Extension interface.
*
******************/

#ifndef _HTTPEXT_H_
#define _HTTPEXT_H_

#include <windows.h>

#ifdef __cplusplus
extern "C" {
#endif

#define   HSE_VERSION_MAJOR           1      // major version of this spec
```

```
#define   HSE_VERSION_MINOR           0      // minor version of this spec
#define   HSE_LOG_BUFFER_LEN         80
#define   HSE_MAX_EXT_DLL_NAME_LEN  256

typedef   LPVOID  HCONN;

// the following are the status codes returned by the Extension DLL

#define   HSE_STATUS_SUCCESS                       1
#define   HSE_STATUS_SUCCESS_AND_KEEP_CONN         2
#define   HSE_STATUS_PENDING                       3
#define   HSE_STATUS_ERROR                         4

// The following are the values to request services with the ServerSupportFunction.
//  Values from 0 to 1000 are reserved for future versions of the interface

#define   HSE_REQ_BASE                             0
#define   HSE_REQ_SEND_URL_REDIRECT_RESP           ( HSE_REQ_BASE + 1 )
#define   HSE_REQ_SEND_URL                         ( HSE_REQ_BASE + 2 )
#define   HSE_REQ_SEND_RESPONSE_HEADER             ( HSE_REQ_BASE + 3 )
#define   HSE_REQ_DONE_WITH_SESSION                ( HSE_REQ_BASE + 4 )
#define   HSE_REQ_END_RESERVED                     1000

//
//  These are Microsoft specific extensions
//

#define   HSE_REQ_MAP_URL_TO_PATH                  (HSE_REQ_END_RESERVED+1)
#define   HSE_REQ_GET_SSPI_INFO                    (HSE_REQ_END_RESERVED+2)


//
// passed to GetExtensionVersion
//

typedef struct   _HSE_VERSION_INFO {

    DWORD  dwExtensionVersion;
    CHAR   lpszExtensionDesc[HSE_MAX_EXT_DLL_NAME_LEN];

} HSE_VERSION_INFO, *LPHSE_VERSION_INFO;

//
// passed to extension procedure on a new request
//
typedef struct _EXTENSION_CONTROL_BLOCK {

    DWORD     cbSize;                 // size of this struct.
    DWORD     dwVersion;              // version info of this spec
    HCONN     ConnID;                 // Context number not to be modified!
    DWORD     dwHttpStatusCode;       // HTTP Status code
    CHAR      lpszLogData[HSE_LOG_BUFFER_LEN];// null terminated log info specific
to this Extension DLL

    LPSTR     lpszMethod;             // REQUEST_METHOD
    LPSTR     lpszQueryString;        // QUERY_STRING
    LPSTR     lpszPathInfo;           // PATH_INFO
    LPSTR     lpszPathTranslated;     // PATH_TRANSLATED

    DWORD     cbTotalBytes;           // Total bytes indicated from client
    DWORD     cbAvailable;            // Available number of bytes
    LPBYTE    lpbData;                // pointer to cbAvailable bytes
```

```
    LPSTR     lpszContentType;        // Content type of client data

    BOOL (WINAPI * GetServerVariable) ( HCONN       hConn,
                                        LPSTR       lpszVariableName,


                                                                LPVOID
lpvBuffer,
                                        LPDWORD     lpdwSize );

    BOOL (WINAPI * WriteClient)  ( HCONN      ConnID,
                                   LPVOID     Buffer,
                                   LPDWORD    lpdwBytes,
                                   DWORD      dwReserved );

    BOOL (WINAPI * ReadClient)  ( HCONN      ConnID,
                                  LPVOID     lpvBuffer,
                                  LPDWORD    lpdwSize );

    BOOL (WINAPI * ServerSupportFunction)( HCONN      hConn,
                                           DWORD      dwHSERRequest,
                                           LPVOID     lpvBuffer,
                                           LPDWORD    lpdwSize,
                                           LPDWORD    lpdwDataType );

} EXTENSION_CONTROL_BLOCK, *LPEXTENSION_CONTROL_BLOCK;

//
//  these are the prototypes that must be exported from the extension DLL
//

BOOL  WINAPI   GetExtensionVersion( HSE_VERSION_INFO  *pVer );
DWORD WINAPI   HttpExtensionProc(  EXTENSION_CONTROL_BLOCK *pECB );

// the following type declarations is for the server side

typedef BOOL  (WINAPI * PFN_GETEXTENSIONVERSION)( HSE_VERSION_INFO  *pVer );
typedef DWORD (WINAPI * PFN_HTTPEXTENSIONPROC )( EXTENSION_CONTROL_BLOCK *pECB );

#ifdef __cplusplus
}
#endif

#endif  // end definition _HTTPEXT_H_
```

INSTALL.C

```
/*       FILE:              INSTALL.C
 *                                    Microsoft TPC-C Kit Ver. 3.00.000
 *                                    Audited 08/23/96, By Francois Raab
 *
 *                                    Copyright Microsoft, 1996
 *
 *       PURPOSE:  Automated installation application for TPC-C Web Kit
 *       Author:            Philip Durr
 *                                    philipdu@Microsoft.com
 */

#include <windows.h>
#include <direct.h>
```

```
#include  <io.h>
#include <stdlib.h>
#include <stdio.h>
#include <commctrl.h>
#include "install.h"

HICON               hIcon;
HINSTANCE hInst;

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

static    BOOL      bLog;
static    int                 iThreads;
static    int                 iMaxWareHouse;
static    int                 iDelayMs;
static    int                 iDeadlockRetry;
static    int                 iMaxConnections;
static    int                 iPoolThreadsLimit;
static    int                 iThreadTimeout;
static    int                 iListenBackLog;
static    int                 iAcceptExOutstanding;
static    int                 iQSlotts;

static    int                 iMaxPhysicalMemory;           //max physical memory in
MB

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

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

          hInst = hInstance;

          InitCommonControls();

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

          iRc = DialogBox(hInstance, MAKEINTRESOURCE(IDD_DIALOG1),
GetDesktopWindow(), MainDlgProc);
          if ( iRc )
                              DialogBoxParam(hInstance,
MAKEINTRESOURCE(IDD_DIALOG2), GetDesktopWindow(), UpdatedDlgProc, (LPARAM)iRc);
          DestroyIcon(hIcon);

          return 0;
}
```

```
BOOL CALLBACK UpdatedDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
          switch(uMsg)
          {
                    case WM_INITDIALOG:
                              if ( lParam == 1 )
                                        SetDlgItemText(hwnd, IDC_RESULTS, "HTML TPCC
Installation Successfull");
                              else
                                        SetDlgItemText(hwnd, IDC_RESULTS, "HTML TPCC
Registry Updated");
                              return TRUE;
                    case WM_COMMAND:
                              if ( wParam == IDOK )
                                        EndDialog(hwnd, TRUE);
                              break;
                    default:
                              break;
          }
          return FALSE;
}

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

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

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

                              bLog                                              =
FALSE;
                              iThreads                                = 4;
                              iMaxWareHouse                           = 0;
                              iDelayMs                                = 500;
                              iDeadlockRetry                          = 3;
                              iMaxConnections                         = 25;
                              iPoolThreadsLimit             = iMaxPhysicalMemory *
2;
                              iThreadTimeout                          = 86400;
                              iListenBackLog                          = 15;
                              iAcceptExOutstanding          = 40;

                              ReadRegistrySettings();
```

```
                        GetModuleFileName(hInst, szExePath,
sizeof(szExePath));
                        GetVersionInfo(szDllPath, szExePath);
                        if ( bLog )
                                CheckDlgButton(hwnd, BN_LOG, 1);

                        wsprintf(szTmp, "Version %d.00.%3.3d", versionExeMS,
versionExeLS);
                        SetDlgItemText(hwnd, IDC_VERSION, szTmp);

                        SetDlgItemText(hwnd, IDC_PATH, szDllPath);
                        SetDlgItemInt(hwnd, ED_MAXWARE, iMaxWareHouse, FALSE);
                        SetDlgItemInt(hwnd, ED_THREADS, iThreads, FALSE);
                        SetDlgItemInt(hwnd, ED_MAXCONNECTION, iMaxConnections,
FALSE);
                        SetDlgItemInt(hwnd, ED_IIS_MAX_THEAD_POOL_LIMIT,
iPoolThreadsLimit, FALSE);
                        SetDlgItemInt(hwnd, ED_IIS_THREAD_TIMEOUT,
iThreadTimeout, FALSE);
                        SetDlgItemInt(hwnd, ED_IIS_LISTEN_BACKLOG,
iListenBackLog, FALSE);
                        SetDlgItemInt(hwnd, ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,
iAcceptExOutstanding, FALSE);

                        return TRUE;
                case WM_PAINT:
                        if ( IsIconic(hwnd) )
                        {
                                BeginPaint(hwnd, &ps);
                                DrawIcon(ps.hdc, 0, 0, hIcon);
                                EndPaint(hwnd, &ps);
                                return TRUE;
                        }
                        break;
                case WM_COMMAND:
                        if ( wParam == IDOK )
                        {
                                if ( IsDlgButtonChecked(hwnd, BN_LOG) )
                                        bLog = TRUE;
                                else
                                        bLog = FALSE;
                                iThreads  = GetDlgItemInt(hwnd, ED_THREADS,
&d, FALSE);
                                iMaxWareHouse = GetDlgItemInt(hwnd,
ED_MAXWARE, &d, FALSE);
                                iMaxConnections = GetDlgItemInt(hwnd,
ED_MAXCONNECTION, &d, FALSE);


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

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

```
                                rc = CopyFiles(hDlg, szDllPath);
                                if ( !rc )
                                {
                                        ShowWindow(hwnd, SW_SHOWNA);
                                        DestroyWindow(hDlg);
                                        MessageBox(hwnd, "Error(s) occured
when creating tpcc.dll", NULL, MB_ICONSTOP | MB_OK);
                                        EndDialog(hwnd, 0);
                                        return TRUE;
                                }
                                SetDlgItemText(hDlg, IDC_STATUS, "Updating
Registry.");

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

                                WriteRegistrySettings(szDllPath);

                                Sleep(100);

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

                                EndDialog(hwnd, rc);
                                return TRUE;
                        }
                        if ( wParam == IDCANCEL )
                        {
                                EndDialog(hwnd, FALSE);
                                return TRUE;
                        }
                        break;
                default:
                        break;
        }
        return FALSE;
}

static void ReadRegistrySettings(void)
{
        HKEY      hKey;
        DWORD     size;
        DWORD     type;
        char      szTmp[256];

        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\TPCC", 0,
KEY_READ, &hKey) == ERROR_SUCCESS )
        {
                size = sizeof(szTmp);

                bLog = FALSE;
                if ( RegQueryValueEx(hKey, "LOG", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
                        if ( !stricmp(szTmp, "ON") )
                                bLog = TRUE;
                iThreads = 4;
                size = sizeof(szTmp);
                if ( RegQueryValueEx(hKey, "NumberOfDeliveryThreads", 0, &type,
szTmp, &size) == ERROR_SUCCESS )
                        iThreads = atoi(szTmp);
                if ( iThreads == 0 )
                        iThreads = 4;
```

```
                    iMaxWareHouse = 0;
                    size = sizeof(szTmp);
                    if ( RegQueryValueEx(hKey, "MaximumWarehouses", 0, &type, szTmp,
&size) == ERROR_SUCCESS )
                              iMaxWareHouse = atoi(szTmp);
                    if ( iMaxWareHouse == 0 )
                              iMaxWareHouse = 0;

                    iDelayMs = 500;
                    size = sizeof(szTmp);
                    if ( RegQueryValueEx(hKey, "BackoffDelay", 0, &type, szTmp,
&size) == ERROR_SUCCESS )
                              iDelayMs = atoi(szTmp);
                    if ( iDelayMs == 0 )
                              iDelayMs = 500;

                    iDeadlockRetry = 3;
                    size = sizeof(szTmp);
                    if ( RegQueryValueEx(hKey, "DeadlockRetry", 0, &type, szTmp,
&size) == ERROR_SUCCESS )
                              iDeadlockRetry = atoi(szTmp);
                              if ( !iDeadlockRetry )
                                        iDeadlockRetry = 3;

                    iMaxConnections = 25;
                    size = sizeof(szTmp);
                    if ( RegQueryValueEx(hKey, "MaxConnections", 0, &type, szTmp,
&size) == ERROR_SUCCESS )
                              iMaxConnections = atoi(szTmp);
                              if ( !iMaxConnections )
                                        iMaxConnections = 25;

                    iQSlotts = 3000;
                    size = sizeof(szTmp);
                    if ( RegQueryValueEx(hKey, "QueueSlotts", 0, &type, szTmp,
&size) == ERROR_SUCCESS )
                              iQSlotts = atoi(szTmp);
                    if ( iQSlotts == 0 )
                              iQSlotts = 3000;

                    RegCloseKey(hKey);

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

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

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

                    RegCloseKey(hKey);

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

                    RegCloseKey(hKey);
          }

          return;
}

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

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

                    RegSetValueEx(hKey, "PATH", 0, REG_SZ, szTmp, strlen(szTmp));

                    if ( bLog )
                              RegSetValueEx(hKey, "LOG", 0, REG_SZ, "ON", 2);
                    else
                              RegSetValueEx(hKey, "LOG", 0, REG_SZ, "OFF", 3);

                    itoa(iThreads, szTmp, 10);
                    RegSetValueEx(hKey, "NumberOfDeliveryThreads", 0, REG_SZ, szTmp,
strlen(szTmp));

                    itoa(iMaxWareHouse, szTmp, 10);
                    RegSetValueEx(hKey, "MaximumWarehouses", 0, REG_SZ, szTmp,
strlen(szTmp));

                    itoa(iDelayMs, szTmp, 10);
                    RegSetValueEx(hKey, "BackoffDelay", 0, REG_SZ, szTmp,
strlen(szTmp));

                    itoa(iDeadlockRetry, szTmp, 10);
                    RegSetValueEx(hKey, "DeadlockRetry", 0, REG_SZ, szTmp,
strlen(szTmp));

                    itoa(iMaxConnections, szTmp, 10);
                    RegSetValueEx(hKey, "MaxConnections", 0, REG_SZ, szTmp,
strlen(szTmp));

                    itoa(iQSlotts, szTmp, 10);
                    RegSetValueEx(hKey, "QueueSlotts", 0, REG_SZ, szTmp,
strlen(szTmp));

                    RegFlushKey(hKey);

                    RegCloseKey(hKey);
          }

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

                    RegFlushKey(hKey);
                    RegCloseKey(hKey);
          }

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

                    RegFlushKey(hKey);
                    RegCloseKey(hKey);
          }

          return;
}

BOOL CALLBACK CopyDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
          if ( uMsg == WM_INITDIALOG )
          {
                    SendDlgItemMessage(hwnd, IDC_PROGRESS1, PBM_SETRANGE, 0,
MAKELPARAM(0, 8));
                    SendDlgItemMessage(hwnd, IDC_PROGRESS1, PBM_SETSTEP, (WPARAM)1,
0);
                    return TRUE;
          }
          return FALSE;
}

static int CopyFiles(HWND hDlg, char *szDllPath)
{
          HGLOBAL                       hDLL;
```

```
          HGLOBAL                       hExe;
          HRSRC                         hResInfo;
          BYTE                          *pSrc;
          HANDLE                        hFile;
          DWORD                         dwSize;
          DWORD                         d;
          char                          szTmp[256];
          char                          *ptr;
          BOOL                          bSvcRunning;

          SetDlgItemText(hDlg, IDC_STATUS, "Stopping Web Service.");
          SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
          UpdateDialog(hDlg);

          bSvcRunning = !StopWWWebService();
          SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
          UpdateDialog(hDlg);

          hResInfo = FindResource(hInst, MAKEINTRESOURCE(IDR_TPCCDLL1), "TPCCDLL");
          SetDlgItemText(hDlg, IDC_STATUS, "Copying Files...");
          SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
          UpdateDialog(hDlg);

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

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

          if ( !WriteFile(hFile, pSrc, dwSize, &d, NULL) )
                    return 0;

          CloseHandle(hFile);

          UnlockResource(hDLL);
          FreeResource(hDLL);

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

          hResInfo = FindResource(hInst, MAKEINTRESOURCE(IDR_DELIVERY1),
"DELIVERY");

          dwSize = SizeofResource(hInst, hResInfo);
          hExe = LoadResource(hInst, hResInfo );
          pSrc = (BYTE *)LockResource(hExe);

          strcpy(szTmp, szDllPath);
          ptr = strstr(szTmp, "tpcc");
          if ( ptr )
                    *ptr = 0;
          strcat(szTmp, "delisrv.exe");

          remove(szTmp);

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

          if ( !WriteFile(hFile, pSrc, dwSize, &d, NULL) )
```

```
                return 0;

        CloseHandle(hFile);

        UnlockResource(hExe);
        FreeResource(hExe);

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

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

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

        return 1;
}

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

        szDllPath[0] =  0;
        bRc = TRUE;
        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\Virtual Roots", 0,
KEY_ALL_ACCESS, &hKey) == ERROR_SUCCESS )
        {
                sv = sizeof(szKey);
                size = sizeof(szTmp);

                if ( RegEnumValue(hKey, 0, szKey, &sv, NULL, NULL, szTmp, &size)
== ERROR_SUCCESS )
                {
                        strcpy(szDllPath, szTmp);
                        bRc = FALSE;
                }
                RegCloseKey(hKey);
        }
        if ( (ptr = strchr(szDllPath, ’,’)) )
                *ptr = 0;

        len = strlen(szDllPath);
        if ( szDllPath[len-1] != ’\\’ )
        {
                szDllPath[len] = ’\\’;
                szDllPath[len+1] = 0;
        }
        strcat(szDllPath, "tpcc.dll");
```

```
        return bRc;
}

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

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

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

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

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

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

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

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

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

          CloseServiceHandle(schService);
          return TRUE;

StartWWWebErr:
          CloseServiceHandle(schService);
          return FALSE;
}

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

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

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

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

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

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

          CloseServiceHandle(schService);
          return TRUE;

StopWWWebErr:
          CloseServiceHandle(schService);
          return FALSE;
}

static void UpdateDialog(HWND hDlg)
{
          MSG msg;

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

INSTALL.H

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by install.rc
//
#define IDD_DIALOG1                                   101
#define IDI_ICON1                                     102
#define IDR_TPCCDLL1                                  103
#define IDD_DIALOG2                                   104
#define IDI_ICON2                                     105
#define IDR_DELIVERY1                                 109
#define IDD_DIALOG3                                   110
#define BN_LOG                                        1001
#define ED_KEEP                                       1002
#define ED_THREADS                                    1003
#define ED_THREADS2                                   1004
#define ED_MAXWARE                                    1006
#define IDC_PATH                                      1007
#define IDC_VERSION                                   1009
#define IDC_RESULTS                                   1010
#define IDC_PROGRESS1                                 1011
#define IDC_STATUS                                    1012
#define IDC_BUTTON1                                   1013
#define ED_MAXCONNECTION                              1014
#define ED_IIS_MAX_THEAD_POOL_LIMIT                   1015
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE             1017
#define ED_IIS_THREAD_TIMEOUT                         1018
#define ED_IIS_LISTEN_BACKLOG                         1019

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE                      111
#define _APS_NEXT_COMMAND_VALUE                       40001
#define _APS_NEXT_CONTROL_VALUE                       1015
#define _APS_NEXT_SYMED_VALUE                         101
#endif
#endif
```

INSTALL.RC

```
//Microsoft Developer Studio generated resource script.
//
#include "install.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, 234, 174
STYLE DS_MODALFRAME | DS_CENTER | WS_MINIMIZEBOX | WS_POPUP | WS_CAPTION |
    WS_SYSMENU
CAPTION "TPCC Web Client Installation Utility"
FONT 8, "MS Sans Serif"
BEGIN
    EDITTEXT        ED_MAXWARE,199,37,21,12,ES_NUMBER,WS_EX_RTLREADING
    CONTROL         "",BN_LOG,"Button",BS_AUTOCHECKBOX | BS_LEFTTEXT |
                    BS_LEFT | BS_VCENTER | WS_TABSTOP,205,51,15,13,
                    WS_EX_STATICEDGE
    EDITTEXT        ED_THREADS,205,66,15,12,ES_NUMBER,WS_EX_RTLREADING
    EDITTEXT        ED_MAXCONNECTION,186,80,34,12,ES_NUMBER,WS_EX_RTLREADING
    EDITTEXT        ED_IIS_MAX_THEAD_POOL_LIMIT,186,94,34,12,ES_NUMBER,
                    WS_EX_RTLREADING
    EDITTEXT        ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,186,108,34,12,
                    ES_NUMBER,WS_EX_RTLREADING
    EDITTEXT        ED_IIS_THREAD_TIMEOUT,186,122,34,12,ES_NUMBER,
                    WS_EX_RTLREADING
    EDITTEXT        ED_IIS_LISTEN_BACKLOG,186,136,34,12,ES_NUMBER,
                    WS_EX_RTLREADING
    DEFPUSHBUTTON   "OK",IDOK,59,153,50,14
    PUSHBUTTON      "Cancel",IDCANCEL,125,153,50,14
    EDITTEXT        IDC_PATH,42,22,178,13,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "Max Number of Warehouses:",IDC_STATIC,42,37,115,12,
                    SS_SUNKEN
    LTEXT           "Write HTML To Log file:",IDC_STATIC,42,51,115,12,
                    SS_SUNKEN
    LTEXT           "Number of Delivery Threads:",IDC_STATIC,42,66,115,12,
                    SS_SUNKEN
    LTEXT           "Max Number of Connections:",IDC_STATIC,41,80,115,12,
                    SS_SUNKEN
    CTEXT           "Version 1.00.001",IDC_VERSION,42,6,178,14,SS_SUNKEN |
                    WS_BORDER,WS_EX_CLIENTEDGE
    ICON            IDI_ICON1,IDC_STATIC,9,6,21,20,0,WS_EX_CLIENTEDGE
    LTEXT           "IIS Max Thread Pool Limit:",IDC_STATIC,41,94,115,12,
                    SS_SUNKEN
    LTEXT           "Web Service Backlog Queue Size:",IDC_STATIC,41,108,115,
                    12,SS_SUNKEN
    LTEXT           "IIS Thread Timeout:",IDC_STATIC,41,122,115,12,SS_SUNKEN
    LTEXT           "IIS Listen Backlog:",IDC_STATIC,41,136,115,12,SS_SUNKEN
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 TPCC 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 TPCC Web Service"
FONT 12, "Arial Black"
BEGIN
    CONTROL         "Progress1",IDC_PROGRESS1,"msctls_progress32",WS_BORDER,
                    7,20,77,13
    CTEXT           "Static",IDC_STATUS,7,7,77,12,SS_SUNKEN
END


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

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
    IDD_DIALOG1, DIALOG
    BEGIN
        LEFTMARGIN, 9
        RIGHTMARGIN, 220
        TOPMARGIN, 6
        BOTTOMMARGIN, 167
    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
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_TPCCDLL1            TPCCDLL DISCARDABLE     "tpcc.dll"

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

VS_VERSION_INFO VERSIONINFO
 FILEVERSION 0,3,0,2
 PRODUCTVERSION 0,3,0,2
 FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
 FILEFLAGS 0x1L
#else
 FILEFLAGS 0x0L
#endif
 FILEOS 0x40004L
 FILETYPE 0x1L
 FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904b0"
        BEGIN
            VALUE "CompanyName", "Microsoft\0"
            VALUE "FileDescription", "install\0"
            VALUE "FileVersion", "0, 3, 0, 2\0"
            VALUE "InternalName", "install\0"
            VALUE "LegalCopyright", "Copyright © 1996\0"
            VALUE "OriginalFilename", "install.exe\0"
            VALUE "ProductName", "Microsoft install\0"
            VALUE "ProductVersion", "0, 3, 0, 2\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END

#endif    // !_MAC


/////////////////////////////////////////////////////////////////////////////
//
// DELIVERY
//

IDR_DELIVERY1           DELIVERY DISCARDABLE    "delisrv.exe"
#endif    // English (U.S.) resources
/////////////////////////////////////////////////////////////////////////////



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


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

Q.C

```
/*        FILE:              Q.C
 *
 *        Based on: Microsoft TPC-C Kit Ver. 3.00.000
 *
 *                                        Copyright Microsoft, 1996
 *                                        Copyright Performance Tuning Corporation,
1997
```

```
 *
 *        PURPOSE:  Tuxedo Server.
 *        Orignal Author:               Philip Durr
 *                                      philipdu@Microsoft.com
 *
 *  MODIFIED        Changed for modularity and to allow for the Tuxedo TM
 *
 *  Later Author:             Edward Whalen
 *                                      Performance Tuning Corporation
 *                                      ewhalen@perftuning.com
 *
 * Last Hacker =)
 *        Mike Nikolaiev -  Changed lots of stuff like pointers to make go fast.
 *        Mike Nikolaiev -  combined queues in one
 *                            Copyright (c) 1998 Compaq Computer Corporation
 */

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

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

#include "trans.h"                                                    //tpckit
transaction header contains definations of structures specific to TPC-C
#include "httpext.h"                                                  //ISAPI DLL
information header

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

#include <tmenv.h>
#include <xa.h>
#include <atmi.h>

void NEWORDER();
void PAYMENT();
void ORDERSTATUS();
void STOCKLEVEL();

BOOL      bLog                          = FALSE;
BOOL                                              bFlush;
                                        //Flush delivery log info when written.
BOOL      verbose                       = FALSE;
BOOL      bError                        = FALSE;

int                 iThreads            = 5;
int                 iMaxWareHouses      = 3000;
int                 iDelayMs            = 500;
short     iMaxConnections = (short)8000;
short     iDeadlockRetry      = (short)3;

DBPROCESS *pdbproc;
```

```
static char                                       szServer[32];
                    //SQL server name
static char                                       szDatabase[32];
                    //tpcc database name
static char                                       szUser[32];
                              //user name
static char                                       szPassword[32];
                    //user password
int spId;


TERM Term;

static char         szTpccLogPath[256]; //path to html log file if logging turned on
in registry.
static char szErrorLogPath[256];        //path to error log file.

static    CRITICAL_SECTION              CriticalSection;
static    CRITICAL_SECTION              ErrorLogCriticalSection;
static    EXTENSION_CONTROL_BLOCK       *gpECB = NULL;
static    int                                                   bTpccExit;
          //exit delivery disconnect loop as dll exiting.

extern void TMLog();
extern BOOL SQLInit();
extern void UtilStrCpy();
extern void UtilStrCpy();
extern BOOL SQLOpenConnection();
extern BOOL SQLCloseConnection();
extern BOOL SQLDetectDeadlock();

/* FUNCTION: tpsvrinit ( int argc, char *argv[])
 *
 * PURPOSE:         Initialize the Server to Database connection.
 *
 * RETURNS:                   int       0         Success
 *                                                -1        Failure
 *
 *
 * COMMENTS:        None
 *
 */

int tpsvrinit ( int argc, char *argv[] )
{

          if ( GetParameters(argc, argv) )
          {
                    PrintParameters();
                    return -1;
          }

          if ( verbose )
                    TMLog("TPSVRINIT: Q: Server %s, Database %s, User %s, Password
%s, Flush %d.",
                              szServer, szDatabase, szUser, szPassword, bFlush);


          if ( ! SQLInit() )
          {
                    TMLog( "Q: SQLInit Failed" );
                    return -1;
          }
          if ( SQLOpenConnection ( NULL, 0, 0, &pdbproc, szServer, szDatabase,
szUser, szPassword, szDatabase, &spId))
          {
                    TMLog ( "Q: SQLOpenConnection Failed" );
                    dbexit();
                    return -1;
          }
          return 0;
}

/* FUNCTION: tpsvrdone ( void )
 *
 * PURPOSE:         Initialize the Server to Database connection.
 *
 * RETURNS:                   int       0         Success
 *                                                -1        Failure
 *
 *
 * COMMENTS:        None
 *
 */

void tpsvrdone ( void )
{
          SQLCloseConnection( NULL, pdbproc);
          dbexit();
}

void Q1 ( TPSVCINFO *rqst )
{
          int size = rqst->len;
          switch(size)
          {
                    case sizeof(NEW_ORDER_DATA):
                              NEWORDER( rqst );
                              break;
                    case sizeof(PAYMENT_DATA):
                              PAYMENT( rqst );
                              break;
                    case sizeof(ORDER_STATUS_DATA):
                              ORDERSTATUS( rqst );
                              break;
                    case sizeof(STOCK_LEVEL_DATA):
                              STOCKLEVEL( rqst );
                              break;
                    default:
                              exit;
          }
}

void Q2 ( TPSVCINFO *rqst )
{
          int size = rqst->len;
          switch(size)
          {
                    case sizeof(NEW_ORDER_DATA):
                              NEWORDER( rqst );
                              break;
                    case sizeof(PAYMENT_DATA):
                              PAYMENT( rqst );
                              break;
                    case sizeof(ORDER_STATUS_DATA):
                              ORDERSTATUS( rqst );
                              break;
                    case sizeof(STOCK_LEVEL_DATA):
                              STOCKLEVEL( rqst );
                              break;
                    default:
                              exit;
          }
}

void Q3 ( TPSVCINFO *rqst )
{
          int size = rqst->len;
          switch(size)
          {
                    case sizeof(NEW_ORDER_DATA):
                              NEWORDER( rqst );
                              break;
                    case sizeof(PAYMENT_DATA):
                              PAYMENT( rqst );
                              break;
                    case sizeof(ORDER_STATUS_DATA):
                              ORDERSTATUS( rqst );
                              break;
                    case sizeof(STOCK_LEVEL_DATA):
                              STOCKLEVEL( rqst );
                              break;
                    default:
                              exit;
          }
}

void Q4 ( TPSVCINFO *rqst )
{
          int size = rqst->len;
          switch(size)
          {
                    case sizeof(NEW_ORDER_DATA):
                              NEWORDER( rqst );
                              break;
                    case sizeof(PAYMENT_DATA):
                              PAYMENT( rqst );
                              break;
                    case sizeof(ORDER_STATUS_DATA):
                              ORDERSTATUS( rqst );
                              break;
                    case sizeof(STOCK_LEVEL_DATA):
                              STOCKLEVEL( rqst );
                              break;
                    default:
                              exit;
          }
}

void Q5 ( TPSVCINFO *rqst )
{
          int size = rqst->len;
          switch(size)
          {
                    case sizeof(NEW_ORDER_DATA):
                              NEWORDER( rqst );
                              break;
                    case sizeof(PAYMENT_DATA):
```

```
                              PAYMENT( rqst );
                              break;
                    case sizeof(ORDER_STATUS_DATA):
                              ORDERSTATUS( rqst );
                              break;
                    case sizeof(STOCK_LEVEL_DATA):
                              STOCKLEVEL( rqst );
                              break;
                    default:
                              exit;
          }
}

void Q6 ( TPSVCINFO *rqst )
{
          int size = rqst->len;
          switch(size)
          {
                    case sizeof(NEW_ORDER_DATA):
                              NEWORDER( rqst );
                              break;
                    case sizeof(PAYMENT_DATA):
                              PAYMENT( rqst );
                              break;
                    case sizeof(ORDER_STATUS_DATA):
                              ORDERSTATUS( rqst );
                              break;
                    case sizeof(STOCK_LEVEL_DATA):
                              STOCKLEVEL( rqst );
                              break;
                    default:
                              exit;
          }
}

void Q7 ( TPSVCINFO *rqst )
{
          int size = rqst->len;
          switch(size)
          {
                    case sizeof(NEW_ORDER_DATA):
                              NEWORDER( rqst );
                              break;
                    case sizeof(PAYMENT_DATA):
                              PAYMENT( rqst );
                              break;
                    case sizeof(ORDER_STATUS_DATA):
                              ORDERSTATUS( rqst );
                              break;
                    case sizeof(STOCK_LEVEL_DATA):
                              STOCKLEVEL( rqst );
                              break;
                    default:
                              exit;
          }
}

void Q8 ( TPSVCINFO *rqst )
{
          int size = rqst->len;
          switch(size)
          {
                    case sizeof(NEW_ORDER_DATA):
```

```
                              NEWORDER( rqst );
                              break;
                    case sizeof(PAYMENT_DATA):
                              PAYMENT( rqst );
                              break;
                    case sizeof(ORDER_STATUS_DATA):
                              ORDERSTATUS( rqst );
                              break;
                    case sizeof(STOCK_LEVEL_DATA):
                              STOCKLEVEL( rqst );
                              break;
                    default:
                              exit;
          }
}

void Q9 ( TPSVCINFO *rqst )
{
          int size = rqst->len;
          switch(size)
          {
                    case sizeof(NEW_ORDER_DATA):
                              NEWORDER( rqst );
                              break;
                    case sizeof(PAYMENT_DATA):
                              PAYMENT( rqst );
                              break;
                    case sizeof(ORDER_STATUS_DATA):
                              ORDERSTATUS( rqst );
                              break;
                    case sizeof(STOCK_LEVEL_DATA):
                              STOCKLEVEL( rqst );
                              break;
                    default:
                              exit;
          }
}

/* FUNCTION: NEWORDER ( TPSVCINFO *rqst )
 *
 * PURPOSE:         Process a New Order request.
 *
 * RETURNS:                   int       0         Success
 *                                                -1        Failure
 *
 *
 * COMMENTS:        None
 *
 */

void NEWORDER ( TPSVCINFO *rqst )
{
          PECBINFO pECBInfo = dbgetuserdata(pdbproc);
          NEW_ORDER_DATA      *np;
          int size = rqst->len;

          np = (NEW_ORDER_DATA * ) rqst->data;

          if (verbose )
          {

                    TMLog(" NEWORDER: d_id %d ", np->d_id);
                    TMLog(" NEWORDER: c_id %d ", np->c_id);
                    TMLog(" NEWORDER: o_ol_cnt %d ", np->o_ol_cnt);
                    TMLog(" NEWORDER: o_all_local %d ", np->o_all_local);
          }

          bError = FALSE;

          np->retval = SQLNewOrder( NULL, 0, 0, pdbproc, np, iDeadlockRetry);

          if (bError == TRUE)
                    np->retval = -1;

          if (verbose )
                    TMLog(" NEWORDER: Return Value %d", np->retval);

          tpreturn( TPSUCCESS, 0, rqst->data, size, 0);
}

/* FUNCTION: int SQLNewOrder(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId, int iTermId, int iSyncId, DBPROCESS *dbproc, NEW_ORDER_DATA  *pNewOrder,
short deadlock_retry)
 *
 * PURPOSE:          This function handles the new order transaction.
 *
 * ARGUMENTS:        EXTENSION_CONTROL_BLOCK       *pECB
          passed in structure pointer from inetsrv.
 *                                      int
                    iTermId                       terminal id of browser
 *                                      int
                    iSyncId                       sync id of browser
 *                                      DBPROCESS
          *dbproc                       connection db process id
 *                                      NEW_ORDER_DATA
          *pNewOrder                    pointer to new order structure for
input/output data
 *                                      short
          deadlock_retry      retry count if deadlocked
 *
 * RETURNS:                   int       TRUE      transaction committed
 *                                                FALSE     item number not valid
 *                                                -1                  deadlock max
retry reached
 *
 *
 * COMMENTS:         None
 *
 */

static int SQLNewOrder(EXTENSION_CONTROL_BLOCK    *pECB, int iTermId, int iSyncId,
DBPROCESS *dbproc, NEW_ORDER_DATA  *pNewOrder, short deadlock_retry)
{
          RETCODE                       rc;
          int                                     i;
          DBINT                         commit_flag;
          int                                     tryit;
          char                          printbuf[25];
          char                          tmpbuf[30];
          DBDATETIME                    datetime;
          BYTE                          *pData;
          PECBINFO            pEcbInfo;

          if ( (pEcbInfo = (PECBINFO)dbgetuserdata(dbproc)) )
          {
                    pEcbInfo->pECB = pECB;
```

```
                    pEcbInfo->bFailed = FALSE;
                    pEcbInfo->iTermId = iTermId;
                    pEcbInfo->iSyncId = iSyncId;
          }

          pNewOrder->num_deadlocks = 0;

          strcpy(tmpbuf, "tpcc_neworder");

          for (tryit=0; tryit < deadlock_retry; tryit++)
          {
                    if (dbrpcinit(dbproc, tmpbuf, 0) == SUCCEED)
                    {
                              dbrpcparam(dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&pNewOrder->w_id);
                              dbrpcparam(dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&pNewOrder->d_id);
                              dbrpcparam(dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *)
&pNewOrder->c_id);
                              dbrpcparam(dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&pNewOrder->o_ol_cnt);
                              dbrpcparam(dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&pNewOrder->o_all_local);

                              for (i = 0; i < pNewOrder->o_ol_cnt; i++)
                              {
                                        dbrpcparam(dbproc, NULL, 0, SQLINT4, -1, -1,
(BYTE *) &pNewOrder->Ol[i].ol_i_id);
                                        dbrpcparam(dbproc, NULL, 0, SQLINT2, -1, -1,
(BYTE *) &pNewOrder->Ol[i].ol_supply_w_id);
                                        dbrpcparam(dbproc, NULL, 0, SQLINT2, -1, -1,
(BYTE *) &pNewOrder->Ol[i].ol_quantity);
                              }

                              if (dbrpcexec(dbproc) == SUCCEED)
                              {
                                        pNewOrder->total_amount=0;

                                        // Get resutls from order line
                                        for (i = 0; i<pNewOrder->o_ol_cnt; i++)
                                        {
                                                  if (((rc = dbresults(dbproc)) !=
NO_MORE_RESULTS) && (rc != FAIL))
                                                  {
                                                            if (DBROWS(dbproc) &&
(dbnumcols(dbproc) == 5))
                                                            {
                                                                      while
(dbnextrow(dbproc) != NO_MORE_ROWS)
                                                                      {

          if(pData=dbdata(dbproc, 1))

          UtilStrCpy(pNewOrder->Ol[i].ol_i_name, pData, dbdatlen(dbproc, 1));

          if(pData=dbdata(dbproc, 2))

          pNewOrder->Ol[i].ol_stock = (*(DBSMALLINT *) pData);

          if(pData=dbdata(dbproc, 3))

          UtilStrCpy(pNewOrder->Ol[i].ol_brand_generic, pData, dbdatlen(dbproc, 3));
          if(pData=dbdata(dbproc, 4))

          pNewOrder->Ol[i].ol_i_price = (*(DBFLT8 *) pData);


          if(pData=dbdata(dbproc, 5))


          pNewOrder->Ol[i].ol_amount = (*(DBFLT8 *) pData);




          pNewOrder->total_amount = pNewOrder->total_amount + pNewOrder-
>Ol[i].ol_amount;
                                                                      }
                                                            }
                                                  }
                                        }
                                        while (((rc = dbresults(dbproc)) !=
NO_MORE_RESULTS) && (rc != FAIL))
                                        {
                                                  if (DBROWS(dbproc) &&
(dbnumcols(dbproc) == 8))
                                                  {
                                                            while (((rc =
dbnextrow(dbproc)) != NO_MORE_ROWS) && (rc != FAIL))
                                                            {

          if(pData=dbdata(dbproc, 1))



          pNewOrder->w_tax = (*(DBFLT8 *) pData);






          if(pData=dbdata(dbproc, 2))


          pNewOrder->d_tax = (*(DBFLT8 *) pData);







          if(pData=dbdata(dbproc, 3))

          pNewOrder->o_id = (*(DBINT *) pData);

          if(pData=dbdata(dbproc, 4))

          UtilStrCpy(pNewOrder->c_last, pData, dbdatlen(dbproc, 4));

          if(pData=dbdata(dbproc, 5))

          pNewOrder->c_discount = (*(DBFLT8 *) pData);
```

```
          if(pData=dbdata(dbproc, 6))

          UtilStrCpy(pNewOrder->c_credit, pData, dbdatlen(dbproc, 6));

          if(pData=dbdata(dbproc, 7))
                                                                      {

          datetime = *((DBDATETIME *) pData);

          dbdatecrack(dbproc, &pNewOrder->o_entry_d, &datetime);
                                                                      }

          if(pData=dbdata(dbproc, 8))commit_flag = (*(DBTINYINT *) pData);
                                                            }
                                                  }
                              }
                              }
                    }
                    if (SQLDetectDeadlock(dbproc))
                    {
                              pNewOrder->num_deadlocks++;
                              sprintf(printbuf,"deadlock: retry: %d",pNewOrder-
>num_deadlocks);
                              Sleep(DEADLOCKWAIT*tryit);
                    }
                    else
                    {
                              if (commit_flag == 1)
                              {
                                        pNewOrder->total_amount = pNewOrder-
>total_amount * ((1 + pNewOrder->w_tax + pNewOrder->d_tax) * (1 - pNewOrder-
>c_discount));
                                        strcpy(pNewOrder-
>execution_status,"Transaction commited.");
                                        return TRUE;
                              }
                              else
                              {
                                        strcpy(pNewOrder->execution_status,"Item
number is not valid.");
                                        return FALSE;
                              }
                    }
          }

          // If we reached here, it means we quit after MAX_RETRY deadlocks
          strcpy(pNewOrder->execution_status,"Hit deadlock max.  ");
          if ( verbose )
                    TMLog(" NEWORDER: SQLNewOrder Max Deadlocks %d", tryit);

          return -1;          //        "deadlock max retry reached!"
}

/************************************************************************************
**************/

/* FUNCTION: PAYMENT ( TPSVCINFO *rqst )
```

```
 *
 * PURPOSE:          Process a Payment request.
 *
 * RETURNS:                    int       0         Success
 *                                                 -1        Failure
 *
 *
 * COMMENTS:         None
 *
 */

void PAYMENT ( TPSVCINFO *rqst )
{
          PECBINFO pECBInfo = dbgetuserdata(pdbproc);
          PAYMENT_DATA *pp;
          int size = rqst->len;

          pp = (PAYMENT_DATA *) rqst->data ;

          if (verbose )
          {
                    TMLog(" PAYMENT: w_id %d ", pp->w_id);
                    TMLog(" PAYMENT: d_id %d ", pp->d_id);
                    TMLog(" PAYMENT: c_id %d ", pp->c_id);
          }

          bError = FALSE;

          pp->retval = SQLPayment( NULL, 0, 0, pdbproc, pp, iDeadlockRetry);

          if (bError == TRUE)
                    pp->retval = -1;

          if (verbose )
                    TMLog(" PAYMENT: Return Value %d", pp->retval);

          tpreturn( TPSUCCESS, 0, rqst->data, size, 0);
}

/* FUNCTION: int SQLPayment(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int iSyncId,
DBPROCESS *dbproc, PAYMENT_DATA *pPayment, short deadlock_retry)
 *
 * PURPOSE:          This function handles the payment transaction.
 *
 * ARGUMENTS:        EXTENSION_CONTROL_BLOCK       *pECB
          passed in structure pointer from inetsrv.
 *                                      int
                    iTermId                       terminal id of browser
 *                                      int
                    iSyncId                       sync id of browser
 *                                      DBPROCESS
          *dbproc                       connection db process id
 *                                      PAYMENT_DATA
          *pPayment           pointer to payment input/output data structure
 *                                      short
          deadlock_retry      deadlock retry count
 *
 * RETURNS:                    int       TRUE                success
 *                                                 -1                            max
deadlocked reached
 *
 * COMMENTS:         None
 *
```

```
 */

static int SQLPayment(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int iSyncId,
DBPROCESS *dbproc, PAYMENT_DATA *pPayment, short deadlock_retry)
{
          RETCODE             rc;
          int                           tryit;
          char                printbuf[26];
          BOOL                by_name;
          DBDATETIME          datetime;
          BYTE                *pData;
          PECBINFO  pEcbInfo;

          if ( (pEcbInfo = (PECBINFO)dbgetuserdata(dbproc)) )
          {
                    pEcbInfo->pECB = pECB;
                    pEcbInfo->bFailed = FALSE;
                    pEcbInfo->iTermId = iTermId;
                    pEcbInfo->iSyncId = iSyncId;
          }

          pPayment->num_deadlocks = 0;

          if (pPayment->c_id == 0)
                    by_name = TRUE;
          else
                    by_name = FALSE;

          for (tryit=0; tryit < deadlock_retry; tryit++)
          {
                    if (dbrpcinit(dbproc, "tpcc_payment", 0) == SUCCEED)
                    {
                              dbrpcparam(dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&pPayment->w_id);
                              dbrpcparam(dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&pPayment->c_w_id);
                              dbrpcparam(dbproc, NULL, 0, SQLFLT8, -1, -1, (BYTE *)
&pPayment->h_amount);
                              dbrpcparam(dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&pPayment->d_id);
                              dbrpcparam(dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&pPayment->c_d_id);
                              dbrpcparam(dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *)
&pPayment->c_id);
                              if (pPayment->c_id == 0)
                              {
                                        dbrpcparam(dbproc, NULL, 0, SQLCHAR, -1,
strlen(pPayment->c_last), pPayment->c_last);
                              }
                    }
                    if (dbrpcexec(dbproc) == SUCCEED)
                    {
                              while (((rc = dbresults(dbproc)) != NO_MORE_RESULTS)
&& (rc != FAIL))
                              {
                                        if (DBROWS(dbproc) && (dbnumcols(dbproc) ==
27))
                                        {
                                                  while (((rc = dbnextrow(dbproc))
!= NO_MORE_ROWS) && (rc != FAIL))
                                                  {
                                                            if(pData=dbdata(dbproc,
1))
```

```
                                                                      pPayment->c_id
= *((DBINT *) pData);
                                                            if(pData=dbdata(dbproc,
2))

          UtilStrCpy(pPayment->c_last, pData, dbdatlen(dbproc, 2));
                                                            if(pData=dbdata(dbproc,
3))
                                                            {
                                                                      datetime =
*((DBDATETIME *) pData);

          dbdatecrack(dbproc, &pPayment->h_date, &datetime);
                                                            }
                                                            if(pData=dbdata(dbproc,
4))

          UtilStrCpy(pPayment->w_street_1, pData, dbdatlen(dbproc, 4));
                                                            if(pData=dbdata(dbproc,
5))

          UtilStrCpy(pPayment->w_street_2, pData, dbdatlen(dbproc, 5));
                                                            if(pData=dbdata(dbproc,
6))

          UtilStrCpy(pPayment->w_city, pData, dbdatlen(dbproc, 6));
                                                            if(pData=dbdata(dbproc,
7))

          UtilStrCpy(pPayment->w_state, pData, dbdatlen(dbproc, 7));
                                                            if(pData=dbdata(dbproc,
8))

          UtilStrCpy(pPayment->w_zip, pData, dbdatlen(dbproc, 8));
                                                            if(pData=dbdata(dbproc,
9))

          UtilStrCpy(pPayment->d_street_1, pData, dbdatlen(dbproc, 9));
                                                            if(pData=dbdata(dbproc,
10))

          UtilStrCpy(pPayment->d_street_2, pData, dbdatlen(dbproc, 10));
                                                            if(pData=dbdata(dbproc,
11))

          UtilStrCpy(pPayment->d_city, pData, dbdatlen(dbproc, 11));
                                                            if(pData=dbdata(dbproc,
12))

          UtilStrCpy(pPayment->d_state, pData, dbdatlen(dbproc, 12));
                                                            if(pData=dbdata(dbproc,
13))

          UtilStrCpy(pPayment->d_zip, pData, dbdatlen(dbproc, 13));
                                                            if(pData=dbdata(dbproc,
14))

          UtilStrCpy(pPayment->c_first, pData, dbdatlen(dbproc, 14));
                                                            if(pData=dbdata(dbproc,
15))

          UtilStrCpy(pPayment->c_middle, pData, dbdatlen(dbproc, 15));
```

```
                                                            if(pData=dbdata(dbproc,
16))

          UtilStrCpy(pPayment->c_street_1, pData, dbdatlen(dbproc, 16));
                                                            if(pData=dbdata(dbproc,
17))

          UtilStrCpy(pPayment->c_street_2, pData, dbdatlen(dbproc, 17));
                                                            if(pData=dbdata(dbproc,
18))

          UtilStrCpy(pPayment->c_city, pData, dbdatlen(dbproc, 18));
                                                            if(pData=dbdata(dbproc,
19))

          UtilStrCpy(pPayment->c_state, pData, dbdatlen(dbproc, 19));
                                                            if(pData=dbdata(dbproc,
20))

          UtilStrCpy(pPayment->c_zip, pData, dbdatlen(dbproc, 20));
                                                            if(pData=dbdata(dbproc,
21))

          UtilStrCpy(pPayment->c_phone, pData, dbdatlen(dbproc, 21));
                                                            if(pData=dbdata(dbproc,
22))
                                                            {
                                                                      datetime =
*((DBDATETIME *) pData);

          dbdatecrack(dbproc, &pPayment->c_since, &datetime);
                                                            }
                                                            if(pData=dbdata(dbproc,
23))

          UtilStrCpy(pPayment->c_credit, pData, dbdatlen(dbproc, 23));
                                                            if(pData=dbdata(dbproc,
24))
                                                                      pPayment-
>c_credit_lim = (*(DBFLT8 *) pData);
                                                            if(pData=dbdata(dbproc,
25))
                                                                      pPayment-
>c_discount = (*(DBFLT8 *) pData);
                                                            if(pData=dbdata(dbproc,
26))
                                                                      pPayment-
>c_balance = (*(DBFLT8 *) pData);
                                                            if(pData=dbdata(dbproc,
27))

          UtilStrCpy(pPayment->c_data, pData, dbdatlen(dbproc, 27));
                                                  }
                                        }
                              }
                    }
                    if (SQLDetectDeadlock(dbproc))
                    {
                              pPayment->num_deadlocks++;
                              sprintf(printbuf,"deadlock: retry: %d",pPayment-
>num_deadlocks);
                              Sleep(DEADLOCKWAIT*tryit);
                    }
                    else
                    {
                              if ( pPayment->c_id == 0 )
                              {
                                        strcpy(pPayment->execution_status,"Invalid
Customer id,name.");
                                        return 0;
                              }
                              else
                                        strcpy(pPayment-
>execution_status,"Transaction commited.");
                              return TRUE;
                    }
          }
          // If we reached here, it means we quit after MAX_RETRY deadlocks
          strcpy(pPayment->execution_status,"Hit deadlock max.  ");
          return -1; //"deadlock max retry reached!"
}

/*********************************************************************************
**************/

void STOCKLEVEL ( TPSVCINFO *rqst )
{
          PECBINFO pECBInfo = dbgetuserdata(pdbproc);
          STOCK_LEVEL_DATA *sp;
          int size = rqst->len;

          sp = (STOCK_LEVEL_DATA *) rqst->data;

          if (verbose )
          {
                    TMLog(" STOCKLEVEL: w_id %d ", sp->w_id);
                    TMLog(" STOCKLEVEL: d_id %d ", sp->d_id);
                    TMLog(" STOCKLEVEL: c_id %d ", sp->thresh_hold);
          }

          bError = FALSE;

          sp->retval = SQLStockLevel( NULL, 0, 0, pdbproc, sp, iDeadlockRetry);


          if (bError == TRUE)
                    sp->retval = -1;

          if ( verbose )
                    TMLog(" STOCKLEVEL: Return Value %d", sp->retval);

          tpreturn( TPSUCCESS, 0, rqst->data, size, 0);
}

/* FUNCTION: SQLStockLevel(EXTENSION_CONTROL_BLOCK          *pECB, int iTermId, int
iSyncId, DBPROCESS *dbproc, STOCK_LEVEL_DATA *pStockLevel, short deadlock_retry)
 *
 * PURPOSE:         This function handles the stock level transaction.
 *
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB
          passed in structure pointer from inetsrv.
 *                                      int
                    iTermId                       terminal id of browser
 *                                      int
                    iSyncId                       sync id of browser
 *                                      DBPROCESS
          *dbproc                       connection db process id
 *                                      STOCK_LEVEL_DATA              *pStockLevel
          stock level input / output data structure
 *                                      short
          deadlock_retry      retry count if deadlocked
 *
 * RETURNS:                   BOOL      FALSE               if successfull
 *                                                          TRUE                if
deadlocked
 *
 * COMMENTS:        None
 *
 */

static BOOL SQLStockLevel(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int iSyncId,
DBPROCESS *dbproc, STOCK_LEVEL_DATA *pStockLevel, short deadlock_retry)
{
          int                           tryit;
          RETCODE             rc;
          char                printbuf[25];
          BYTE                *pData;
          PECBINFO  pEcbInfo;

          //update pECB and bFailed flag
          if ( (pEcbInfo = (PECBINFO)dbgetuserdata(dbproc)) )
          {
                    pEcbInfo->pECB = pECB;
                    pEcbInfo->bFailed = FALSE;
                    pEcbInfo->iTermId = iTermId;
                    pEcbInfo->iSyncId = iSyncId;
          }

          pStockLevel->num_deadlocks = 0;

          for (tryit=0; tryit < deadlock_retry; tryit++)
          {
                    if (dbrpcinit(dbproc, "tpcc_stocklevel", 0) == SUCCEED)
                    {
                              dbrpcparam(dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&pStockLevel->w_id);
                              dbrpcparam(dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&pStockLevel->d_id);
                              dbrpcparam(dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&pStockLevel->thresh_hold);

                              if (dbrpcexec(dbproc) == SUCCEED)
                              {
                                        while (((rc = dbresults(dbproc)) !=
NO_MORE_RESULTS) && (rc != FAIL))
                                        {
                                                  if (DBROWS(dbproc))
                                                  {
                                                            while (((rc =
dbnextrow(dbproc)) != NO_MORE_ROWS) && (rc != FAIL))
                                                            {

          if(pData=dbdata(dbproc, 1))

          pStockLevel->low_stock = *((long *) pData);
                                                            }
                                                  }
                                        }
                              }
                    }
                    if (SQLDetectDeadlock(dbproc))
                    {
                              pStockLevel->num_deadlocks++;
                              sprintf(printbuf,"deadlock: retry: %d",pStockLevel-
>num_deadlocks);
                              Sleep(10 * tryit);
                    }
                    else
                    {
                              strcpy(pStockLevel->execution_status, "Transaction
commited.");
                              return TRUE;
                    }
          }

          // If we reached here, it means we quit after MAX_RETRY deadlocks
          strcpy(pStockLevel->execution_status, "Hit deadlock max. ");
          return -1;
}


/************************************************************************************
**************/


/* FUNCTION: ORDERSTATUS ( TPSVCINFO *rqst )
 *
 * PURPOSE:         Process an Order Status request.
 *
 * RETURNS:                   int       0         Success
 *                                                -1        Failure
 *
 *
 * COMMENTS:        None
 *
 */

void ORDERSTATUS ( TPSVCINFO *rqst )
{
          PECBINFO pECBInfo = dbgetuserdata(pdbproc);
          ORDER_STATUS_DATA *op ;
          int size = rqst->len;

          op = (ORDER_STATUS_DATA *) rqst->data;

          if (verbose )
          {
                    TMLog(" ORDERSTATUS: w_id %d ", op->w_id);
                    TMLog(" ORDERSTATUS: d_id %d ", op->d_id);
                    TMLog(" ORDERSTATUS: c_id %d ", op->c_id);
          }

          bError = FALSE;

          op->retval = SQLOrderStatus( NULL, 0, 0, pdbproc, op, iDeadlockRetry);

          if (bError == TRUE)
                    op->retval = -1;

          if ( verbose )
                    TMLog(" ORDERSTATUS: Return Value %d", op->retval);
```

```
          if (verbose )
          {
                    TMLog(" ORDERSTATUS: w_id %d ", op->w_id);
                    TMLog(" ORDERSTATUS: d_id %d ", op->d_id);
                    TMLog(" ORDERSTATUS: c_id %d ", op->c_id);
                    TMLog(" ORDERSTATUS: c_balance %f ", op->c_balance);
                    TMLog(" ORDERSTATUS: o_id %d ", op->o_id);
                    TMLog(" ORDERSTATUS: o_carrier_id %d ", op->o_carrier_id);
                    TMLog(" ORDERSTATUS: execution_status %s ", op-
>execution_status);
          }

          tpreturn( TPSUCCESS, 0, rqst->data, size, 0);
}

/* FUNCTION: int SQLOrderStatus(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId, DBPROCESS *dbproc, ORDER_STATUS_DATA *pOrderStatus, short deadlock_retry)
 *
 * PURPOSE:         This function processes the Order Status transaction.
 *
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB
          passed in structure pointer from inetsrv.
 *                                      int
                    iTermId                       terminal id of browser
 *                                      int
                    iSyncId                       sync id of browser
 *                                      DBPROCESS
          *dbproc                       connection db process id
 *                                      ORDER_STATUS_DATA             *pOrderStatus
          pointer to Order Status data input/output structure
 *                                      short
          deadlock_retry      deadlock retry count
 *
 * RETURNS:                   int       -1                  max deadlock reached
 *                                                0                  No orders
found for customer
 *                                                1                  Transaction
successfull
 *
 * COMMENTS:        None
 *
 */

static int SQLOrderStatus(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int iSyncId,
DBPROCESS *dbproc, ORDER_STATUS_DATA *pOrderStatus, short deadlock_retry)
{
          RETCODE             rc;
          int                           tryit;
          int                           i;
          char                printbuf[25];
          BOOL                by_name;
          DBDATETIME          datetime;
          BYTE                *pData;
          PECBINFO  pEcbInfo;

          if ( (pEcbInfo = (PECBINFO)dbgetuserdata(dbproc)) )
          {
                    pEcbInfo->pECB = pECB;
                    pEcbInfo->bFailed = FALSE;
                    pEcbInfo->iTermId = iTermId;
                    pEcbInfo->iSyncId = iSyncId;
          }
          pOrderStatus->num_deadlocks = 0;
          if (pOrderStatus->c_id == 0)
                    by_name = TRUE;
          else
                    by_name = FALSE;

          for (tryit=0; tryit < deadlock_retry; tryit++)
          {
                    if (dbrpcinit(dbproc, "tpcc_orderstatus", 0) == SUCCEED)
                    {
                              dbrpcparam(dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&pOrderStatus->w_id);
                              dbrpcparam(dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *)
&pOrderStatus->d_id);
                              dbrpcparam(dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *)
&pOrderStatus->c_id);
                              if (pOrderStatus->c_id == 0)
                              {
                                        dbrpcparam(dbproc, NULL, 0, SQLCHAR, -1,
strlen(pOrderStatus->c_last), pOrderStatus->c_last);
                              }
                    }
                    if (dbrpcexec(dbproc) == SUCCEED)
                    {
                              while (((rc = dbresults(dbproc)) != NO_MORE_RESULTS)
&& (rc != FAIL))
                              {
                                        if (DBROWS(dbproc) && (dbnumcols(dbproc) ==
5))
                                        {
                                                  i=0;
                                                  while (((rc = dbnextrow(dbproc))
!= NO_MORE_ROWS) && (rc != FAIL))
                                                  {
                                                            if(pData=dbdata(dbproc,
1))
                                                                      pOrderStatus-
>OlOrderStatusData[i].ol_supply_w_id = (*(DBSMALLINT *) pData);
                                                            if(pData=dbdata(dbproc,
2))
                                                                      pOrderStatus-
>OlOrderStatusData[i].ol_i_id = (*(DBINT *) pData);
                                                            if(pData=dbdata(dbproc,
3))
                                                                      pOrderStatus-
>OlOrderStatusData[i].ol_quantity = (*(DBSMALLINT *) pData);
                                                            if(pData=dbdata(dbproc,
4))
                                                                      pOrderStatus-
>OlOrderStatusData[i].ol_amount = (*(DBFLT8 *) pData);
                                                            if(pData=dbdata(dbproc,
5))
                                                            {
                                                                      datetime =
*((DBDATETIME *) pData);

          dbdatecrack(dbproc, &pOrderStatus->OlOrderStatusData[i].ol_delivery_d,
&datetime);
                                                            }
                                                            i++;
                                                  }
                                                  pOrderStatus->o_ol_cnt = i;
```

```
                                        }
                                        else if (DBROWS(dbproc) &&
(dbnumcols(dbproc) == 8))
                                        {
                                                  while (((rc = dbnextrow(dbproc))
!= NO_MORE_ROWS) && (rc != FAIL))
                                                  {
                                                            if(pData=dbdata(dbproc,
1))
                                                                      pOrderStatus-
>c_id = (*(DBINT *) pData);
                                                            if(pData=dbdata(dbproc,
2))

          UtilStrCpy(pOrderStatus->c_last, pData, dbdatlen(dbproc,2));
                                                            if(pData=dbdata(dbproc,
3))

          UtilStrCpy(pOrderStatus->c_first, pData, dbdatlen(dbproc,3));
                                                            if(pData=dbdata(dbproc,
4))

          UtilStrCpy(pOrderStatus->c_middle, pData, dbdatlen(dbproc, 4));
                                                            if(pData=dbdata(dbproc,
5))

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

          dbdatecrack(dbproc, &pOrderStatus->o_entry_d, &datetime);
                                                            }
                                                            if(pData=dbdata(dbproc,
6))
                                                                      pOrderStatus-
>o_carrier_id = (*(DBSMALLINT *) pData);
                                                            if(pData=dbdata(dbproc,
7))
                                                                      pOrderStatus-
>c_balance = (*(DBFLT8 *) pData);
                                                            if(pData=dbdata(dbproc,
8))
                                                                      pOrderStatus-
>o_id = (*(DBINT *) pData);
                                                  }
                                        }
                                        if (i==0)
                                                  return 0; //"No orders found for
customer"
                              }
                    }
                    if (SQLDetectDeadlock(dbproc))
                    {
                              pOrderStatus->num_deadlocks++;
                              sprintf(printbuf,"deadlock: retry: %d",pOrderStatus-
>num_deadlocks);
                              Sleep(DEADLOCKWAIT*tryit);
                    }
                    else
                    {
                              if (pOrderStatus->c_id == 0 && pOrderStatus->c_last[0]
== 0)
                                        strcpy(pOrderStatus-
>execution_status,"Invalid Customer id,name.");
```

```
                              else
                                        strcpy(pOrderStatus-
>execution_status,"Transaction commited.");
                              return 1;
                    }
          }
          // If we reached here, it means we quit after MAX_RETRY deadlocks
          strcpy(pOrderStatus->execution_status,"Hit deadlock max.  ");
          return -1; //"deadlock max retry reached!"
}


/*********************************************************************************
**************/


/*
 *        Common Code for all Servers
 */

 /* FUNCTION: BOOL SQLInit()
 *
 * PURPOSE:         This function initializes SQL Server for later use.
 *
 *
 * RETURNS:                   BOOL      FALSE     if successfull
 *                                                          TRUE      if an error
occurs and connection cannot be established.
 *
 * COMMENTS:        None
 *
 */
BOOL SQLInit ()
{

          dbinit();

          if ( dbgetmaxprocs() < iMaxConnections )
          {
                    if ( dbsetmaxprocs(iMaxConnections) == FAIL )
                    {
                              //set for fail error message when HttpExtensionProc()
is called because
                              //at this point we don't have a pECB so no way to show
error message.
                              iMaxConnections = -1;
                    }
          }

          // install error and message handlers
          dbmsghandle((DBMSGHANDLE_PROC)msg_handler);
          dberrhandle((DBERRHANDLE_PROC)err_handler);

          return TRUE;
}


/* FUNCTION: BOOL SQLOpenConnection(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId, DBPROCESS **dbproc, char *server, char *database, char *user, char
*password, char *app, int *spid, long *pack_size)
 *
 * PURPOSE:         This function opens the sql connection for use.
 *
```

```
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB     passed in structure
pointer from inetsrv.
 *                                      int
          iTermId             terminal id of browser
 *                                      int
          iSyncId             sync id of browser
 *                                      DBPROCESS           **dbproc  pointer to
returned DBPROCESS
 *                                      char                          *server
          SQL server name
 *                                      char                          *database SQL
server database
 *                                      char                          *user
          user name
 *                                      char                          *password user
password
 *                                      char                          *app
          pointer to returned application array
 *                                      int
          *spid               pointer to returned spid
 *                                      long                          *pack_size
          pointer to returned default pack size
 *
 * RETURNS:                   BOOL      FALSE     if successfull
 *                                                          TRUE      if an error
occurs
 *
 * COMMENTS:        None
 *
 */


#ifdef USE_ODBC
          static BOOL SQLOpenConnection(EXTENSION_CONTROL_BLOCK *pECB, int iTermId,
int iSyncId, DBPROCESS **dbproc, char *server, char *database, char *user, char
*password, char *app, int *spid, long *pack_size)
          {

                    RETCODE   rc;
                    char      buffer[30];

                    *dbproc = (DBPROCESS *)malloc(sizeof(DBPROCESS));
                    if ( !*dbproc )
                              return TRUE;

                    //set pECB data into dbproc
                    (*dbproc)->bDeadlock = FALSE;
                    (*dbproc)->bFailed = FALSE;
                    (*dbproc)->pECB = pECB;
                    (*dbproc)->iTermId = iTermId;
                    (*dbproc)->iSyncId = iSyncId;


                    if ( SQLAllocConnect(henv, &(*dbproc)->hdbc) == SQL_ERROR )
                              return TRUE;

                    if ( SQLSetConnectOption((*dbproc)->hdbc, SQL_PACKET_SIZE,
pack_size) == SQL_ERROR )
                              return TRUE;

                    rc = SQLConnect((*dbproc)->hdbc, server, SQL_NTS, user, SQL_NTS,
password, SQL_NTS);
                    if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
```

```
                              return TRUE;
                    rc = SQLAllocStmt((*dbproc)->hdbc, &(*dbproc)->hstmt);
                    if (rc == SQL_ERROR)
                              return TRUE;

                    sprintf(buffer,"use %s", Client->database);

                    rc = SQLExecDirect((*dbproc)->hstmt, buffer, SQL_NTS);
                    if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                              return TRUE;

                    SQLFreeStmt((*dbproc)->hstmt, SQL_CLOSE);
                    sprintf(buffer,"set nocount on");
                    rc = SQLExecDirect((*dbproc)->hstmt, buffer, SQL_NTS);
                    if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                              return TRUE;
                    SQLFreeStmt((*dbproc)->hstmt, SQL_CLOSE);

                    sprintf(buffer,"select @@spid");

                    rc = SQLExecDirect((*dbproc)->hstmt, buffer, SQL_NTS);
                    if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                              return TRUE;

                    if ( SQLBindCol((*dbproc)->hstmt, 1, SQL_C_SSHORT, &(*dbproc)-
>spid, 0, NULL) == SQL_ERROR )
                              return TRUE;

                    if ( SQLFetch((*dbproc)->hstmt) == SQL_ERROR )
                              return TRUE;

                    SQLFreeStmt((*dbproc)->hstmt, SQL_CLOSE);

                    return FALSE;
          }

#else

          static BOOL SQLOpenConnection(EXTENSION_CONTROL_BLOCK *pECB, int iTermId,
int iSyncId, DBPROCESS **dbproc, char *server, char *database, char *user, char
*password, char *app, int *spid)
          {
                    LOGINREC  *login;
                    PECBINFO  pEcbInfo;

                    //set local msg proc for login record
                    //attach pECB record

                    //this is necessary as dblib provides no way to pass user data
in a login structure. So until
                    //there is an allocated dbproc we need to use a static which
means that the login attempt must
                    //be serialized.

                    gpECB = pECB;

                    login = dblogin();
                    if ( !*user )
                              DBSETLUSER(login, "sa");
                    else
                              DBSETLUSER(login, user);

                    DBSETLPWD(login,  password);
```

```
                    DBSETLHOST(login, app);

                    DBSETLPACKET(login, (unsigned short)DEFCLPACKSIZE);

                    if ((*dbproc = dbopen(login, server )) == NULL)
                              return TRUE;

                    //set pECB data into dbproc
                    pEcbInfo = (PECBINFO)malloc(sizeof(ECBINFO));
                    pEcbInfo->bDeadlock = FALSE;
                    pEcbInfo->pECB = pECB;
                    pEcbInfo->iTermId = iTermId;
                    pEcbInfo->iSyncId = iSyncId;
                    dbsetuserdata(*dbproc, pEcbInfo);

                    // Use the the right database
                    dbuse(*dbproc, database);

                    dbcmd(*dbproc, "select @@spid");

                    dbsqlexec(*dbproc);
                    while (dbresults(*dbproc) != NO_MORE_RESULTS)
                    {
                              dbbind(*dbproc, 1, SMALLBIND, (DBINT) 0, (BYTE *)
spid);
                              while (dbnextrow(*dbproc) != NO_MORE_ROWS)
                              ;
                    }
                    dbcmd(*dbproc, "set nocount on");

                    dbsqlexec(*dbproc);
                    while (dbresults(*dbproc) != NO_MORE_RESULTS)
                    {
                              while (dbnextrow(*dbproc) != NO_MORE_ROWS)
                              ;
                    }

                    //rollback transaction on abort
                    dbcmd(*dbproc, "set XACT_ABORT ON");

                    dbsqlexec(*dbproc);
                    while (dbresults(*dbproc) != NO_MORE_RESULTS)
                    {
                              while (dbnextrow(*dbproc) != NO_MORE_ROWS)
                              ;
                    }

                    return FALSE;
          }

#endif

/* FUNCTION: BOOL SQLCloseConnection(EXTENSION_CONTROL_BLOCK *pECB, DBPROCESS
*dbproc)
 *
 * PURPOSE:         This function closes the sql connection.
 *
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB     passed in structure
pointer from inetsrv.
 *                                      DBPROCESS           *dbproc   pointer to
DBPROCESS
 *
 * RETURNS:                   BOOL      FALSE     if successfull
 *                                                          TRUE      if an error
occurs
 *
 * COMMENTS:        None
 *
 */

#ifdef USE_ODBC
          static BOOL SQLCloseConnection(EXTENSION_CONTROL_BLOCK *pECB, DBPROCESS
*dbproc)
          {
                    if ( dbproc )
                    {
                              SQLFreeStmt(dbproc->hstmt, SQL_DROP);
                              SQLDisconnect(dbproc->hdbc);
                              SQLFreeConnect(dbproc->hdbc);
                              free(dbproc);
                              dbproc = NULL;
                    }
                    return FALSE;
          }
#else
          static BOOL SQLCloseConnection(EXTENSION_CONTROL_BLOCK *pECB, DBPROCESS
*dbproc)
          {
                    if (dbclose(dbproc) == FAIL)
                              return TRUE;
                    return FALSE;
          }
#endif

/* FUNCTION: BOOL SQLDetectDeadlock(DBPROCESS *dbproc)
 *
 * PURPOSE:         This function checks to see if a sql server deadlock condition
exists.
 *
 * ARGUMENTS:       DBPROCESS                     *dbproc
          connection db process id to check
 *
 * RETURNS:                   BOOL      FALSE               no deadlock detected
 *                                                          TRUE
          deadlock condition exists
 *
 * COMMENTS:        None
 *
 */

BOOL SQLDetectDeadlock(DBPROCESS *dbproc)
{
          PECBINFO  pEcbInfo;

          if ( (pEcbInfo = (PECBINFO)dbgetuserdata(dbproc)) )
          {
                    if ( pEcbInfo->bDeadlock )
                    {
                              pEcbInfo->bDeadlock = FALSE;
                              return TRUE;
                    }
          }
          return FALSE;
}

// Lifted from HP FDR since they did such a nice job
```

```
void TMLog( char *format, ... )
{
          va_list args;
          char buf[4096];
          int len;
          va_start( args, format );
          _strtime( buf );
          strcat( buf," ");
          len = strlen( buf );
          (void)_vsnprintf( buf+ len, sizeof( buf) - len - 1, format, args);
          buf[sizeof( buf )- 1]= ’\0’;
          va_end( args );
          userlog( buf );
}

/* 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.
 *
 */

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

          return;
}

/* FUNCTION: int err_handler(DBPROCESS *dbproc, int severity, int dberr, int oserr,
char *dberrstr, char *oserrstr)
 *
 * PURPOSE:         This function handles DB-Library errors
 *
 * ARGUMENTS:       DBPROCESS           *dbproc                       DBPROCESS id
pointer
 *                                      int
          severity            severity of error
 *                                      int
          dberr                         error id
 *                                      int
          oserr                         operating system specific error code
 *                                      char                          *dberrstr
          printable error description of dberr
 *                                      char                          *oserrstr
          printable error description of oserr
 *
 * RETURNS:                   int                                     INT_CONTINUE
          continue if error is SQLETIME else INT_CANCEL action
 *
 * COMMENTS:        None
 *
 */


int err_handler(DBPROCESS *dbproc, int severity, int dberr, int oserr, char
*dberrstr, char *oserrstr)
{
          PECBINFO                                pEcbInfo;
          EXTENSION_CONTROL_BLOCK       *pECB;
          FILE                                              *fp;
          SYSTEMTIME                                        systemTime;
          char                                              szTmp[256];
          int                                                         iTermId;
          int                                                         iSyncId;

          pEcbInfo = NULL;

          if ((dbproc == NULL) || (DBDEAD(dbproc)))
          {
                    TMLog("DBPROC is invalid");
                    return INT_CANCEL;
          }

          if ( !(pEcbInfo = (PECBINFO)dbgetuserdata(dbproc)) )
          {
                    pECB = gpECB;
                    iTermId = 0;
                    iSyncId = 0;
          }
          else
          {
                    pECB = pEcbInfo->pECB;
                    iTermId = pEcbInfo->iTermId;
                    iSyncId = pEcbInfo->iSyncId;
          }

          if ( pEcbInfo && pEcbInfo->bFailed )
          {
                    bError == FALSE;
                    return INT_CANCEL;
          }

          if ( oserr != DBNOERR )
          {
                    TMLog("DBLIB Error %s", oserrstr);
                    if ( pEcbInfo )
                    {
                              pEcbInfo->bFailed = TRUE;
                              bError = TRUE;
                    }

                    GetLocalTime(&systemTime);
                    fp = fopen(szErrorLogPath, "ab");

                    sprintf(szTmp, "ErrorHandler: DBLIB(%d): %s", oserr, oserrstr);

                    TMLog("%2.2d/%2.2d/%2.2d %2.2d:%2.2d:%2.2d\r\n\r\n%s\r\n\r\n",
                              systemTime.wYear, systemTime.wMonth, systemTime.wDay,
                              systemTime.wHour, systemTime.wMinute,
systemTime.wSecond,
                              szTmp);
```

```
                    fclose(fp);
          }

          return INT_CANCEL;
}


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

int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int severity, char
*msgtext)
{
          PECBINFO                                pEcbInfo;
          EXTENSION_CONTROL_BLOCK       *pECB;
          FILE                                              *fp;
          SYSTEMTIME                                        systemTime;
          char                                              szTmp[256];
          int                                                         iTermId;
          int                                                         iSyncId;

          if ( !(pEcbInfo = (PECBINFO)dbgetuserdata(dbproc)) )
          {
                    pECB = gpECB;
                    iTermId = 0;
                    iSyncId = 0;
          }
          else
          {
                    pECB = pEcbInfo->pECB;
                    iTermId = pEcbInfo->iTermId;
                    iSyncId = pEcbInfo->iSyncId;
          }


          if ( (msgno == 5701) || (msgno == 2528) || (msgno == 5703) || (msgno ==
6006) )
                    return INT_CONTINUE;
          // deadlock message
          if (msgno == 1205)
          {
                    // set the deadlock indicator
                    if ( pEcbInfo )
                              pEcbInfo->bDeadlock = TRUE;
                    else
                              TMLog("Error, dbgetuserdata returned NULL.");
                    return INT_CONTINUE;
          }
          if ( pEcbInfo && pEcbInfo->bFailed )
          {
                    TMLog("SQL Error ");
                    return INT_CANCEL;
          }

          if (msgno == 0)
                    return INT_CONTINUE;
          else
          {
                    TMLog("MsgHandler: SQL Error %s", msgtext);

                    if ( pEcbInfo )
                              pEcbInfo->bFailed = TRUE;

                    bError = TRUE;

                    GetLocalTime(&systemTime);

                    sprintf(szTmp, "Error: SQLSVR(%d): %s", msgno, msgtext);

                    TMLog("%2.2d/%2.2d/%2.2d %2.2d:%2.2d:%2.2d\r\n\r\n%s\r\n\r\n",
                              systemTime.wYear, systemTime.wMonth, systemTime.wDay,
                              systemTime.wHour, systemTime.wMinute,
systemTime.wSecond,
                              szTmp);
          }
          return INT_CANCEL;
}


/* FUNCTION: BOOL GetParameters(int argc, char *argv[])
 *
 * PURPOSE:         This function parses the command line passed in to the delivery
executable, initializing
 *                            and filling in global variable parameters.
 *
 * ARGUMENTS:       int                 argc      number of command line arguments
passed to delivery
 *                                      char      *argv[]   array of command line
argument pointers
 *
 * RETURNS:                   BOOL      FALSE     parameter read successfull
 *                                                          TRUE      user has
requested parameter information screen be displayed.
 *
 * COMMENTS:        None
 *
 */

static BOOL GetParameters(int argc, char *argv[])
{
          int i;
```

```
          szServer[0]                   = 0;
          szPassword[0]       = 0;
          bFlush                        = FALSE;
          strcpy(szDatabase, "tpcc");
          strcpy(szUser, "sa");

          for(i=0; i<argc; i++)
          {
                    if ( argv[i][0] == ’-’ || argv[i][0] == ’/’ )
                    {
                              switch(argv[i][1])
                              {
                                        case ’S’:
                                        case ’s’:
                                                  strcpy(szServer, argv[i]+2);
                                                  break;
                                        case ’V’:
                                        case ’v’:
                                                  verbose = TRUE;
                                                  break;
                                        case ’?’:
                                                  return TRUE;
                              }
                    }
          }
          return FALSE;
}

/* FUNCTION: void PrintParameters(void)
 *
 * PURPOSE:         This function displays the supported command line flags.
 *
 * ARGUMENTS:       None
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

static void PrintParameters(void)
{
          TMLog("Performance Tuning Corporation Tuxedo Kit");
          TMLog("   www.perftuning.com   (281) 251-3495  ");
          TMLog("NewOrder: -S Server -D Database -U User -P Password" );
          TMLog("NewOrder: Server %s, Database %s, User %s, Password %s.",
                    szServer, szDatabase, szUser, szPassword);

}
```

RESOURCE.H

```
//{{NO_DEPENDENCIES}}

// Microsoft Developer Studio generated include file.

// Used by TPCC.rc

//
```

```
// Next default values for new objects

//

#ifdef APSTUDIO_INVOKED

#ifndef APSTUDIO_READONLY_SYMBOLS

#define _APS_NEXT_RESOURCE_VALUE        101

#define _APS_NEXT_COMMAND_VALUE         40001

#define _APS_NEXT_CONTROL_VALUE         1000

#define _APS_NEXT_SYMED_VALUE           101

#endif

#endif
```

TESTGV.C

```
#include "getval.h"
main()
{
char s1[]= {"Test String DID*=08& CID*=123& IID*=54321& The end of test"};
char s2[20];

char * ptr;

  ptr = s1;

  GetVal(ptr, "DID*", s2, sizeof(s2))
  printf("Value of DID = %s\n",s2);

  GetVal(ptr, "CID*", s2, sizeof(s2))
  printf("Value of CID = %s\n",s2);

  GetVal(ptr, "IID*", s2, sizeof(s2))
  printf("Value of IID = %s\n",s2);

}
```

TPCC.C

```
/*        FILE:               TPCC.C
 *
 *        Based on: Microsoft TPC-C Kit Ver. 3.00.000
 *
 *                                      Copyright Microsoft, 1996
 *                                      Copyright Performance Tuning Corporation,
1997
 *
 *        PURPOSE:  TPC-C main program.
 *        Original Author:              Philip Durr
 *                                      philipdu@Microsoft.com
 *
 *  MODIFIED        Changed for modularity and to allow for the Tuxedo TM
 *
 *  Later Author:             Edward Whalen
 *                                        Performance Tuning Corporation
 *                                        ewhalen@perftuning.com
 *
 *        LAST HACKER =)
 *                                        Mike Nikolaiev - Copywrite (C) 1998 Compaq
Computer Corporation
*/

#define LOCAL_ALLOC 1
#define QQQ 1

#define PERFORMIX
//#define BOIL

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

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

#include "trans.h"                                                    //tpckit
transaction header contains definations of structures specific to TPC-C
#include "httpext.h"                                                  //ISAPI DLL
information header
#include "getval.h"

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

#include <tmenv.h>
#include <xa.h>
#include <atmi.h>

static TPINIT       *tpinf;
static DWORD        TLSIsTpInitedKey;
static DWORD        TLSNewOrderKey;
static DWORD        TLSPaymentKey;
static DWORD        TLSOrderStatusKey;
static DWORD        TLSDeliveryKey;
static DWORD        TLSStockLevelKey;
static int                    ThrTpInit();

char      qname[9][4]= {"Q1","Q2","Q3","Q4","Q5","Q6","Q7","Q8","Q9"};
char      szServer[32]        = "EDW";            //global variables used with this
DLL
char      szUser[32]                    = "sa";
char      szPassword[32]      = "";
char      szDatabase[32]      = "tpcc";
char      szIID[16][8]        =
{"IID00*","IID01","IID02","IID03","IID04","IID05","IID06","IID07","IID08","IID09","I
ID10","IID11","IID12","IID13","IID14","IID15"};
char      szSP[16][8]         =
{"SP00*","SP01","SP02","SP03","SP04","SP05","SP06","SP07","SP08","SP09","SP10","SP11
","SP12","SP13","SP14","SP15"};
```

```
char      szQty[16][8]        =
{"Qty00*","Qty01","Qty02","Qty03","Qty04","Qty05","Qty06","Qty07","Qty08","Qty09","Q
ty10","Qty11","Qty12","Qty13","Qty14","Qty15"};

BOOL      bLog                          = FALSE;
BOOL      dLog                          = FALSE;

int                 iThreads            = 5;
int                 iMaxWareHouses      = 3000;
int                 iDelayMs            = 500;
short     iDeadlockRetry      = (short)3;
short     iMaxConnections = (short)8000;
int                 iErrVal                       = 0;

char buffer[256];

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

//defined command string functions, called via CMD=command http string from html
client.

void (*DoCmd[])(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int
iSyncId) =
{
          NewOrderForm,
          PaymentForm,
          DeliveryForm,
          OrderStatusForm,
          StockLevelForm,
          Exitcmd,
          SubmitCmd,
          BeginCmd,
          ProcessCmd,
          MenuCmd,
          ClearCmd
};



//Terminal client id structure and interface defination
TERM      Term = { 0, 0, 0, FALSE, NULL, TermInit, TermAllocate, TermRestore,
TermAdd, TermDelete };

//welcome to tpc-c html form buffer, this is first form client sees.
static char         *szWelcomeForm =    "<HTML>"

          "<HEAD><TITLE>Welcome To TPC-C</TITLE></HEAD><BODY>"

          "Please Identify your Warehouse and District for this session.<BR>"

          "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"

          "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"

          "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"

          "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"1\">"
```

```
          "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"-2\">"

          "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"0\">"

          "Warehouse ID <INPUT NAME=\"w_id\" SIZE=4><BR>"

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

          "<HR>"

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

          "</FORM><BODY>"

          "</HTML>";


static char         szTpccLogPath[256]; //path to html log file if logging turned on
in registry.
static char szErrorLogPath[256];        //path to error log file.

static    CRITICAL_SECTION              CriticalSection;
static    CRITICAL_SECTION              ErrorLogCriticalSection;

static    EXTENSION_CONTROL_BLOCK       *gpECB;
static    int                                                     bTpccExit;
          //exit delivery disconnect loop as dll exiting.

/* FUNCTION: BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call, LPVOID
lpReserved)
 *
 * PURPOSE:         This function is the entry point for the DLL this implementation
is baised on the
 *                            fact that DLL_PROCESS_ATTACH is only called from the
inet service once. Connections
 *                            are sent to this function as thread attachments.
 *
 * ARGUMENTS:       HANDLE    hModule                           module handle
 *                                      DWORD     ul_reason_for_call  reason for
call
 *                                      LPVOID    lpReserved
          reserved for future use
 *
 * RETURNS:                   BOOL      FALSE
          errors occured in initialization
 *                                                          TRUE
                    DLL successfully initialized
 *
 * COMMENTS:        None
 *
 */

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
          int                                               i;
          static SECURITY_ATTRIBUTES    sa;
          static PSECURITY_DESCRIPTOR pSD;

          switch( ul_reason_for_call )
          {
                    case DLL_PROCESS_ATTACH:
                              if ( ReadRegistrySettings() )
                              {
                                        MessageBox(NULL, "Cannot Find TPCC Key in
registry (run install.exe).", "Init", MB_OK | MB_ICONSTOP);
                                        return FALSE;
                              }

                              InitializeCriticalSection(&CriticalSection);
                              InitializeCriticalSection(&ErrorLogCriticalSection);

                              (*Term.Init)();
                              if ( !(*Term.Allocate)() )
                              {
                                        MessageBox(NULL, "Error Trm.Allocate().",
"Init", MB_OK | MB_ICONSTOP);
                                        return FALSE;
                              }
                              for(i=Term.iNext; i<Term.iAvailable; i++)
                                        Term.pClientData[i].inUse = 0;
                              Term.pClientData[0].inUse = 1;

                              TLSIsTpInitedKey = TlsAlloc(); // check for failure
later
                              TLSNewOrderKey = TlsAlloc();
                              TLSPaymentKey = TlsAlloc();
                              TLSOrderStatusKey = TlsAlloc();
                              TLSDeliveryKey = TlsAlloc();
                              TLSStockLevelKey = TlsAlloc();
                              // assumption:value inited to 0
                              break;
                    case DLL_THREAD_ATTACH:
                              break;
                    case DLL_THREAD_DETACH:
                              break;
                    case DLL_PROCESS_DETACH:
                              if ( pSD )
                                        free( pSD );

                              bTpccExit = TRUE;

                              (*Term.Restore)();

                              DeleteCriticalSection(&CriticalSection);
                              DeleteCriticalSection(&ErrorLogCriticalSection);

                              TlsFree(TLSIsTpInitedKey);
                              TlsFree(TLSNewOrderKey);
                              TlsFree(TLSPaymentKey);
                              TlsFree(TLSOrderStatusKey);
                              TlsFree(TLSDeliveryKey);
                              TlsFree(TLSStockLevelKey);

                              break;
          }
          return TRUE;
}


/* FUNCTION: BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO *pVer)
 *
 * PURPOSE:           This function is called by the inet service when the DLL is
first loaded.
 *
```

```
 * ARGUMENTS:         HSE_VERSION_INFO    *pVer     passed in structure in which to
place expected version number.
 *
 * RETURNS:                     TRUE      inet service expected return value.
 *
 * COMMENTS:          None
 *
 */

BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO *pVer)
{

          pVer->dwExtensionVersion = MAKELONG(HSE_VERSION_MINOR, HSE_VERSION_MAJOR);
          lstrcpyn(pVer->lpszExtensionDesc, "TPC-C Server.",
HSE_MAX_EXT_DLL_NAME_LEN);

          return TRUE;
}

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

DWORD WINAPI HttpExtensionProc(EXTENSION_CONTROL_BLOCK *pECB)
{
          int iCmd, FormId, TermId, iSyncId;
          FILE *fp;

          static BOOL bReadRegistry = FALSE;

          if ( iMaxConnections == -1 )
          {
                    ErrorMessage(pECB, ERR_CAN_NOT_SET_MAX_CONNECTIONS,
ERR_TYPE_WEBDLL, NULL, -1, -1);
                    return HSE_STATUS_SUCCESS;
          }

          //if registry setting is for html logging then show http string passed in.
          if ( bLog )
          {
                    SYSTEMTIME          systemTime;

                    fp = fopen(szTpccLogPath, "ab");

                    GetLocalTime(&systemTime);
                    fprintf(fp, "* QUERY * %2.2d/%2.2d/%2.2d
%2.2d:%2.2d:%2.2d\r\n\r\n%s\r\n\r\n",
                              systemTime.wYear, systemTime.wMonth, systemTime.wDay,
                              systemTime.wHour, systemTime.wMinute,
systemTime.wSecond,
                              pECB->lpszQueryString);
                    fclose(fp);
          }


          //process http query
          if ( !ProcessQueryString(pECB, &iCmd, &FormId, &TermId, &iSyncId) )
          {
                    if ( TermId < 0 )
                              ErrorMessage(pECB, ERR_INVALID_TERMID,
ERR_TYPE_WEBDLL, NULL, TermId, iSyncId);
                    else
                              ErrorMessage(pECB, ERR_COMMAND_UNDEFINED,
ERR_TYPE_WEBDLL, NULL, TermId, iSyncId);
                    return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
          }

          if ( TermId != 0 )
          {
                    if ( !IsValidTermId(TermId) )
                    {
                              ErrorMessage(pECB, ERR_INVALID_TERMID,
ERR_TYPE_WEBDLL, NULL, TermId, iSyncId);
                              return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
                    }

                    //must have a valid syncid here since termid is valid
                    if ( iSyncId < 1 || iSyncId != Term.pClientData[TermId].iSyncId
)
                    {
                              ErrorMessage(pECB, ERR_INVALID_SYNC_CONNECTION,
ERR_TYPE_WEBDLL, NULL, TermId, iSyncId);
                              return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
                    }

          }

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

          //go execute http: command
          (*DoCmd[iCmd])(pECB, FormId, TermId, iSyncId);

          //finish up and keep connection
          return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}

/* FUNCTION: static BOOL IsValidTermId(int TermId)
 *
 * PURPOSE:         This function checks to see of the passed in terminal id is
valid.
 *
 * ARGUMENTS:       int                                     TermId
                    client terminal id
 *
 * RETURNS:                   BOOL      FALSE
                    Terminal ID Invalid
```

```
 *                                                          TRUE
                                        Terminal ID valid
 *
 * COMMENTS:        None
 *
 */

static BOOL IsValidTermId(int TermId)
{
          return (BOOL) ( TermId > 0 && TermId <= Term.iAvailable &&
Term.pClientData[TermId].inUse );
}

/* FUNCTION: BOOL ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd, int
*pFormId, int *pTermId, int *pSyncId)
 *
 * PURPOSE:         This function extracts the relevent information out of the http
command passed in from
 *                            the browser.
 *
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB               structure
pointer to passed in internet
 *
                                                  service information.
 *                                      int
          *pCmd                                   returned command id
 *                                      int
          *pFormId                      returned active form client browser is on
 *                                      int
          *pTermId                      returned client terminal id
 *
 * RETURNS:                   BOOL      FALSE
                    success
 *                                                          TRUE
                                        command passed in is invalid
 *
 * COMMENTS:        If this is the initial connection i.e. client is at welcome
screen then
 *                                      there will not be a terminal id or current
form id if this is the case
 *                                      then the pTermid and pFormid return values
are undefined.
 */

BOOL ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd, int *pFormId, int
*pTermId, int *pSyncId)
{
          char *ptr;
          char szBuffer[25];
          char szTmp[25];
          char *dest = szBuffer;
          int i;

          if ( (ptr = strstr(pECB->lpszQueryString, "FORMID=")) )
                    *pFormId = *(ptr+7) & 0x0F;

          if ( (ptr = strstr(pECB->lpszQueryString, "TERMID=")) )
          {
                    *pTermId = atoi((ptr+7));
                    if ( *pTermId == 0 )          //terminal id 0 used internally
                              *pTermId = -1;
                    if ( *pTermId == -2 )         //login screen
                              *pTermId = 0;
          }
          else
                    *pTermId = 0;


          if ( (ptr = strstr(pECB->lpszQueryString, "SYNCID=")) )
                    *pSyncId = atoi((ptr+7));
          else
                    *pSyncId = 0;

          if ( !(ptr = strstr(pECB->lpszQueryString, "CMD=")) )
          {

                    ptr = szBuffer;
                    if ( !stricmp(szBuffer, "Default") )
                              strcpy(szBuffer, "CMD=Begin");
                    switch( *pFormId )
                    {
                              case WELCOME_FORM:
                                        strcpy(szBuffer, "CMD=Submit");
                                        break;
                              case MAIN_MENU_FORM:
                                        strcpy(szBuffer, "CMD=NewOrder");
                                        break;
                              case NEW_ORDER_FORM:
                              case PAYMENT_FORM:
                              case DELIVERY_FORM:
                              case ORDER_STATUS_FORM:
                              case STOCK_LEVEL_FORM:
                                        if ( !(*pTermId) )
                                                  return FALSE;
                                        if ( GetKeyValue(pECB->lpszQueryString,
"PI*", szTmp, sizeof(szTmp)) )
                                                  strcpy(szBuffer, "CMD=Process");
                                        else
                                        {
                                                  strcpy(szBuffer, "CMD=");
                                                  strcat(szBuffer, szCmds[*pFormId -
NEW_ORDER_FORM]);
                                        }
                                        break;
                              default:
                                        return FALSE;
                    }
          }

          ptr += 4;

          while( *ptr && *ptr != ’&’ )
                    *dest++ = *ptr++;
          *dest = 0;

          for(i=0; szCmds[i][0]; i++)
          {
                    if ( !strcmp(szCmds[i], szBuffer) )
                    {
                              *pCmd = i;
                              return TRUE;
                    }
          }
          return FALSE;
}
```

```
/* FUNCTION: void NewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId)
 *
 * PURPOSE:         This function wraps the functionality needed for the TPC-C New
Order Form.
 *
 * ARGUMENTS:       int
          iFormId             unused
 *                                      int
                    iTermId             id of calling browser, i.e. TERMID= from
http command line
 *                                      EXTENSION_CONTROL_BLOCK       *pECB
          structure pointer to passed in internet
 *
                                                  service information.
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

void NewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int
iSyncId)
{
          WriteZString(pECB, MakeNewOrderForm(iTermId, iSyncId, TRUE, FALSE));

          UNUSEDPARAM(iFormId);

          return;
}

/* FUNCTION: void PaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId)
 *
 * PURPOSE:         This function wraps the functionality needed for the TPC-C
Payment Form.
 *
 * ARGUMENTS:       int
          iFormId             unused
 *                                      int
                    iTermId             id of calling browser, i.e. TERMID= from
http command line
 *                                      int
                    iSyncId             sync id of calling browser
 *                                      EXTENSION_CONTROL_BLOCK       *pECB
          structure pointer to passed in internet
 *
                                                  service information.
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

void PaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int
iSyncId)
{
          WriteZString(pECB, MakePaymentForm(iTermId, iSyncId, TRUE) );

          UNUSEDPARAM(iFormId);

}
```

```
/* FUNCTION: void DeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId)
 *
 * PURPOSE:         This function wraps the functionality needed for the TPC-C
Delivery Form.
 *
 * ARGUMENTS:       int
          iFormId             unused
 *                                       int
                    iTermId             id of calling browser, i.e. TERMID= from
http command line
 *                                       int
                    iSyncId             sync id of calling browser
 *                                       EXTENSION_CONTROL_BLOCK       *pECB
          structure pointer to passed in internet
 *
                                                  service information.
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

void DeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int
iSyncId)
{
          WriteZString(pECB, MakeDeliveryForm(iTermId, iSyncId, TRUE) );

          UNUSEDPARAM(iFormId);
}

/* FUNCTION: void OrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId)
 *
 * PURPOSE:         This function wraps the functionality needed for the TPC-C Order
Status Form.
 *
 * ARGUMENTS:       int
          iFormId             unused
 *                                       int
                    iTermId             id of calling browser, i.e. TERMID= from
http command line
 *                                       int
                    iSyncId             sync id of calling borwser
 *                                       EXTENSION_CONTROL_BLOCK       *pECB
          structure pointer to passed in internet
 *
                                                  service information.
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

void OrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int
iSyncId)
{
          WriteZString(pECB, MakeOrderStatusForm(iTermId, iSyncId, TRUE) );

          UNUSEDPARAM(iFormId);

}
```

```
/* FUNCTION: void StockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId)
 *
 * PURPOSE:         This function wraps the functionality needed for the TPC-C Stock
Level Form.
 *
 * ARGUMENTS:       int
          iFormId             unused
 *                                       int
                    iTermId             id of calling browser, i.e. TERMID= from
http command line
 *                                       int
                    iSyncId             sync id of calling browser
 *                                       EXTENSION_CONTROL_BLOCK       *pECB
          structure pointer to passed in internet
 *
                                                  service information.
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

void StockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int
iSyncId)
{
          WriteZString(pECB, MakeStockLevelForm(iTermId, iSyncId, TRUE) );

          return;
}

/* FUNCTION: void Exitcmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId)
 *
 * PURPOSE:         This function removes a terminal id from use, the allocated
structure however remains
 *                            valid so the next request for a new client will not
require a new memory allocation.
 *
 * ARGUMENTS:       int
          iFormId             unused
 *                                       int
                    iTermId             id of calling browser, i.e. TERMID= from
http command line
 *                                       int
                    iSyncId             sync id of calling browser
 *                                       EXTENSION_CONTROL_BLOCK       *pECB
          structure pointer to passed in internet
 *
                                                  service information.
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

void Exitcmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int iSyncId)
{
          (*Term.Delete)(pECB, iTermId);

          WriteZString(pECB, MakeWelcomeForm() );
          UNUSEDPARAM(iFormId);
          UNUSEDPARAM(iSyncId);

          return;
}

/* FUNCTION: void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId)
 *
 * PURPOSE:         This function allocated a new terminal id in the Term structure
array.
 *
 * ARGUMENTS:       int
          iFormId             unused
 *                                      int
                    iTermId             id of calling browser, i.e. TERMID= from
http command line
 *                                      int
                    iSyncId             sync id of calling browser
 *                                      EXTENSION_CONTROL_BLOCK       *pECB
          structure pointer to passed in internet
 *
                                                  service information.
 * RETURNS:                   None
 *
 * COMMENTS:        A terminal id can be allocated but still be invalid if the
requested warehouse number
 *                                      is outside the range specified in the
registry. This then will force the client id
 *                                      to be invalid and an error message sent to
the users browser.
 */

void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int iSyncId)
{
          int       iCurrent;

          if ( (iCurrent = (*Term.Add)(pECB, pECB->lpszQueryString)) < 0 )
          {
                    ErrorMessage(pECB, ERR_CANNOT_INIT_TERMINAL, ERR_TYPE_WEBDLL,
NULL, iCurrent, iSyncId);
                    return;
          }

          if ( Term.pClientData[iCurrent].w_id > iMaxWareHouses ||
Term.pClientData[iCurrent].w_id < 1 )
          {
                    ErrorMessage(pECB, ERR_W_ID_INVALID, ERR_TYPE_WEBDLL, NULL,
iCurrent, iSyncId);
                    (*Term.Delete)(pECB, iCurrent);
                    return;
          }
          if ( Term.pClientData[iCurrent].d_id < 1 ||
Term.pClientData[iCurrent].d_id > 10 )
          {
                    ErrorMessage(pECB, ERR_D_ID_INVALID, ERR_TYPE_WEBDLL, NULL,
iCurrent, iSyncId);
                    (*Term.Delete)(pECB, iCurrent);
                    return;
          }

          WriteZString(pECB, MakeMainMenuForm(iCurrent,
Term.pClientData[iCurrent].iSyncId) );
```

```
          return;
}

/* FUNCTION: void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId)
 *
 * PURPOSE:         This function is the first command executed. It is executed with
the command
 *                            CMD=Begin?Server=xxx from the http command line.
 *
 * ARGUMENTS:       int
          iFormId             unused
 *                                      int
                    iTermId             id of calling browser, i.e. TERMID= from
http command line
 *                                      int
                    iSyncId             sync id of calling browser
 *                                      EXTENSION_CONTROL_BLOCK       *pECB
          structure pointer to passed in internet
 *
                                                  service information.
 * RETURNS:                   None
 *
 * COMMENTS:        SQL server must be specified, however the user and password
parameters are optional.
 *                                      The complete command line is
CMD=Begin&Server=server&User=sa&Psw=&. The & are used
 *                                      to separate parameters which is internet
browser standard.
 */

void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int iSyncId)
{
          LPSTR pQueryString;

          pQueryString = pECB->lpszQueryString;

/*        if ( !GetKeyValue(pQueryString, "Server", szServer, sizeof(szServer)) )
          {
                    ErrorMessage(pECB, ERR_NO_SERVER_SPECIFIED, ERR_TYPE_WEBDLL,
NULL, iTermId, iSyncId);
                    return;
                    strcpy(szServer, "TPCC");
          }

          if ( !GetKeyValue(pQueryString, "User", szUser, sizeof(szUser)) )
                    strcpy(szUser, "sa");

          if ( !GetKeyValue(pQueryString, "Psw", szPassword, sizeof(szPassword)) )
                    strcpy(szPassword, "");

          if ( !GetKeyValue(pQueryString, "Db", szDatabase, sizeof(szDatabase)) )
                    strcpy(szDatabase, "tpcc");
*/
          WriteZString(pECB, MakeWelcomeForm() );

          UNUSEDPARAM(iFormId);

          return;
}
```

```
/* FUNCTION: void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId)
 *
 * PURPOSE:         This function process the passed in http command
 *
 * ARGUMENTS:       int
          iFormId             unused
 *                                      int
                    iTermId             id of calling browser, i.e. TERMID= from
http command line
 *                                      int
                    iSyncId             sync id of calling browser
 *                                      EXTENSION_CONTROL_BLOCK       *pECB
          structure pointer to passed in internet
 *
                                                  service information.
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int
iSyncId)
{
          switch( iFormId )
          {
                    case WELCOME_FORM:
                              return;
                    case MAIN_MENU_FORM:
                              return;
                    case NEW_ORDER_FORM:
                              ProcessNewOrderForm(pECB, iTermId, iSyncId);
                              return;
                    case PAYMENT_FORM:
                              ProcessPaymentForm(pECB, iTermId, iSyncId);
                              return;
                    case DELIVERY_FORM:
                              ProcessDeliveryForm(pECB, iTermId, iSyncId);
                              return;
                    case ORDER_STATUS_FORM:
                              ProcessOrderStatusForm(pECB, iTermId, iSyncId);
                              return;
                    case STOCK_LEVEL_FORM:
                              ProcessStockLevelForm(pECB, iTermId, iSyncId);
                              return;

          }
}

/* FUNCTION: void ClearCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId)
 *
 * PURPOSE:         This function frees all currently logged in terminal ids.
 *
 * ARGUMENTS:       int
          iFormId             unused
 *                                      int
                    iTermId             id of calling browser, i.e. TERMID= from
http command line
 *                                      int
                    iSyncId             sync id of calling browser
 *                                      EXTENSION_CONTROL_BLOCK       *pECB
          structure pointer to passed in internet
 *
                                                  service information.
 * RETURNS:                   None
 *
 * COMMENTS:        Use this function with caution, it may cause unpredictable
results
 *                                      if existing browsers attempt to use the web
client with out
 *                                      beginning at the login screen for each
client.
 */

void ClearCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int iSyncId)
{
          int i;

          EnterCriticalSection(&CriticalSection);

          for(i=0; i<Term.iAvailable; i++)
          {
                    if ( Term.pClientData[i].inUse )
                              (*Term.Delete)(pECB, i);
          }

          Term.iNext                              = 0;
          Term.iAvailable               = 0;
          Term.iMasterSyncId  = 1;

          if ( Term.pClientData )
                    free(Term.pClientData);
          Term.pClientData    = NULL;
          Term.bInit                              = FALSE;

          (*Term.Init)();
          if ( !(*Term.Allocate)() )
          {
                    ErrorMessage(pECB, ERR_MAX_CONNECT_PARAM, ERR_TYPE_WEBDLL, NULL,
iTermId, iSyncId);
                    return;
          }
          for(i=Term.iNext; i<Term.iAvailable; i++)
                    Term.pClientData[i].inUse = 0;
          Term.pClientData[0].inUse = 1;

          LeaveCriticalSection(&CriticalSection);

          WriteZString(pECB, MakeWelcomeForm() );

          return;
}

/* FUNCTION: void MenuCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId)
 *
 * PURPOSE:         This function causes an exit to the main menu
 *
 * ARGUMENTS:       int
          iFormId             unused
 *                                      int
                    iTermId             id of calling browser, i.e. TERMID= from
http command line
 *                                      int
                    iSyncId             sync id of calling browser
 *                                       EXTENSION_CONTROL_BLOCK       *pECB
         structure pointer to passed in internet
 *
                                                  service information.
 * RETURNS:                    None
 *
 * COMMENTS:         None
 *
 */

void MenuCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int iSyncId)
{
          WriteZString(pECB, MakeMainMenuForm(iTermId, iSyncId) );

          return;
}


/* FUNCTION: void WriteZString(EXTENSION_CONTROL_BLOCK *pECB, char *szStr)
 *
 * PURPOSE:          This function is the low level output function. It writes a
string of text back to the
 *                             client browser.
 *
 * ARGUMENTS:        EXTENSION_CONTROL_BLOCK       *pECB     passed in structure
pointer from inetsrv.
 *                                       char                          *szStr
                    string to display in the client browser.
 *
 * RETURNS:                    None
 *
 * COMMENTS:         This function assumes that the string to written to the client
browser has
 *                                       been formatted in an HTML manner.
 */

static void WriteZString(EXTENSION_CONTROL_BLOCK *pECB, char *szStr)
{
          FILE      *fp;
          int       lpbSize;
#ifndef PERFORMIX
          int       iSize;
          char      szHeader[128];
          char      szHeader1[128];
#endif

          lpbSize = strlen(szStr)+1;

          if ( bLog )
          {
                    SYSTEMTIME          systemTime;

                    fp = fopen(szTpccLogPath, "ab");

                    GetLocalTime(&systemTime);

                    fprintf(fp, "* HTML PAGE * %2.2d/%2.2d/%2.2d
%2.2d:%2.2d:%2.2d\r\n\r\n%s\r\n\r\n",
                              systemTime.wYear, systemTime.wMonth, systemTime.wDay,
                              systemTime.wHour, systemTime.wMinute,
systemTime.wSecond,
                              szStr);
                    fclose(fp);
          }

#ifdef PERFORMIX

          (*pECB->ServerSupportFunction)(pECB->ConnID, HSE_REQ_DONE_WITH_SESSION,
NULL, 0, 0);

          (*pECB->WriteClient)(pECB->ConnID, szStr, &lpbSize, 0);


#else
          iSize = sprintf(szHeader, "200 Ok");
          sprintf(szHeader1, "Connection: keep-alive\r\nContent-type:
text/html\r\nContent-length: %d\r\n\r\n", lpbSize);
          (*pECB->ServerSupportFunction)(pECB->ConnID, HSE_REQ_SEND_RESPONSE_HEADER,
szHeader, &iSize, (LPDWORD)szHeader1);

          (*pECB->WriteClient)(pECB->ConnID, szStr, &lpbSize, 0);
#endif

          return;
}

/* FUNCTION: void h_printf(EXTENSION_CONTROL_BLOCK *pECB, char *format, ...)
 *
 * PURPOSE:          This function forms a high level printf for an HTML browser
 *
 * ARGUMENTS:        EXTENSION_CONTROL_BLOCK       *pECB     passed in structure
pointer from inetsrv.
 *                                       char                          *format
                    printf style format string
 *                                       ...
                                         other arguments as required by printf style
format string.
 *
 * RETURNS:                    None
 *
 * COMMENTS:         This function is mainly used for developmental support.
 */

static void h_printf(EXTENSION_CONTROL_BLOCK *pECB, char *format, ...)
{
          int lpbSize;
          char szBuff[512];
          char szTmp[512];

          va_list marker;
          va_start( marker, format );
          vsprintf(szTmp, format, marker);
          va_end( marker );

          lpbSize = wsprintf(szBuff, "<html>%s</html>", szTmp) + 1;

          (*pECB->WriteClient)(pECB->ConnID, szBuff, &lpbSize, 0);

          return;
}

/* FUNCTION: void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int iError, int
iErrorType, char *szMsg)
 *
 * PURPOSE:          This function displays an error message in the client browser.
 *
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB     passed in structure
pointer from inetsrv.
 *                                       int
          iError                         id of error message
 *                                       int
          iErrorType                     error type, ERR_TYPE_SQL, ERR_TYPE_DBLIB, or
ERR_TYPE_WEBDLL
 *                                       int
          iTermId                        terminal id from browser
 *                                       int
          iSyncid                        sync id from browser
 *                                       char *                        szMsg
                    optional error message string used with ERR_TYPE_SQL and
 *
                                         ERR_TYPE_DBLIB
 *
 * RETURNS:                   None
 *
 * COMMENTS:        If the error type is ERR_TYPE_WEBDLL the szmsg parameter may be
NULL because it
 *                                       is ignored. If the error type is
ERR_TYPE_SQL or ERR_TYPE_DBLIB then the szMsg
 *                                       parameter contains the text of the error
message, so the szMsg parameter cannot
 *                                       be NULL.
 *
 */

void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int iError, int iErrorType, char
*szMsg, int iTermId, int iSyncId)
{
          int i;

          static SERRORMSG errorMsgs[] =
          {
                    {         ERR_SUCCESS,
                              "Success, no error."

          },
                    {         ERR_COMMAND_UNDEFINED,
                    "Command undefined."
                                                                                },
                    {         ERR_NOT_IMPLEMENTED_YET,
          "Not Implemented Yet."
                                                                      },
                    {         ERR_CANNOT_INIT_TERMINAL,
          "Cannot initialize client connection."
                                        },
                    {         ERR_OUT_OF_MEMORY,
                    "insufficient memory."
                                                                                },
                    {         ERR_NEW_ORDER_NOT_PROCESSED,
          "Cannot process new Order form."
                                                  },
                    {         ERR_PAYMENT_NOT_PROCESSED,
          "Cannot process payment form."
                                                            },
                    {         ERR_NO_SERVER_SPECIFIED,
          "No Server name specified."
                                                            },
```

```
                    {         ERR_ORDER_STATUS_NOT_PROCESSED,
          "Cannot process order status form."
                                                  },
                    {         ERR_W_ID_INVALID,
                    "Invalid Warehouse ID."
                                                                                },
                    {         ERR_CAN_NOT_SET_MAX_CONNECTIONS,
          "Insufficient memory to allocate # connections."
                    },
                    {         ERR_NOSUCH_CUSTOMER,
                    "No such customer."
                                                                                },
                    {         ERR_D_ID_INVALID,
                    "Invalid District ID Must be 1 to 10."
                                                  },
                    {         ERR_MAX_CONNECT_PARAM,
                    "Max client connections exceeded, run install to increase."
                    },
                    {         ERR_INVALID_SYNC_CONNECTION,
          "Invalid Terminal Sync ID."
                                                            },
                    {         ERR_INVALID_TERMID,
                    "Invalid Terminal ID."
                                                                                },
                    {         ERR_PAYMENT_INVALID_CUSTOMER,
          "Payment Form, No such Customer."
                                                  },
                    {         ERR_SQL_OPEN_CONNECTION,
          "SQLOpenConnection API Failed."
                                                            },
                    {         ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,   "Stock Level
missing Threshold key \"TT*\"."
          },
                    {         ERR_STOCKLEVEL_THRESHOLD_INVALID,
          "Stock Level Threshold invalid data type range = 1 - 99."
          },
                    {         ERR_STOCKLEVEL_THRESHOLD_RANGE,
          "Stock Level Threshold out of range, range must be 1 - 99."
          },
                    {         ERR_STOCKLEVEL_NOT_PROCESSED,
          "Stock Level not processed."
                                                  },
                    {         ERR_NEWORDER_FORM_MISSING_DID,
          "New Order missing District key \"DID*\"."
                                        },
                    {         ERR_NEWORDER_DISTRICT_INVALID,
          "New Order District ID Invalid range 1 - 10."
                              },
                    {         ERR_NEWORDER_DISTRICT_RANGE,                      "New
Order District ID out of Range. Range = 1 - 10."                                },
                    {         ERR_NEWORDER_CUSTOMER_KEY,
          "New Order missing Customer key \"CID*\"."
                                        },
                    {         ERR_NEWORDER_CUSTOMER_INVALID,
          "New Order customer id invalid data type, range = 1 to 3000."
          },
                    {         ERR_NEWORDER_CUSTOMER_RANGE,                      "New
Order customer id out of range, range = 1 to 3000."                             },
                    {         ERR_NEWORDER_MISSING_IID_KEY,                     "New
Order missing Item Id key \"IID*\"."
                    },
                    {         ERR_NEWORDER_ITEM_BLANK_LINES,
          "New Order blank order lines all orders must be continuous."
          },
                    {         ERR_NEWORDER_ITEMID_INVALID,                    "New
Order Item Id is wrong data type, must be numeric."                         },
                    {         ERR_NEWORDER_MISSING_SUPPW_KEY,
          "New Order missing Supp_W key \"SP##*\"."
                                        },
                    {         ERR_NEWORDER_SUPPW_INVALID,
          "New Order Supp_W invalid data type must be numeric."
                    },
                    {         ERR_NEWORDER_MISSING_QTY_KEY,                   "New
Order Missing Qty key \"Qty##*\"."
                              },
                    {         ERR_NEWORDER_QTY_INVALID,
          "New Order Qty invalid must be numeric range 1 - 99."
                    },
                    {         ERR_NEWORDER_SUPPW_RANGE,
          "New Order Supp_W value out of range range = 1 - Max Warehouses."     },
                    {         ERR_NEWORDER_ITEMID_RANGE,
          "New Order Item Id is out of range. Range = 1 to 999999."
          },
                    {         ERR_NEWORDER_QTY_RANGE,
                    "New Order Qty is out of range. Range = 1 to 99."
                              },
                    {         ERR_PAYMENT_DISTRICT_INVALID,
          "Payment District ID is invalid must be 1 - 10."
                    },
                    {         ERR_NEWORDER_SUPPW_WITHOUT_ITEMID,              "New
Order Supp_W field entered without a corrisponding Item_Id."          },
                    {         ERR_NEWORDER_QTY_WITHOUT_ITEMID,                "New
Order Qty entered without a corrisponding Item_Id."                         },
                    {         ERR_NEWORDER_NOITEMS_ENTERED,                   "New
Order Blank Items between items, items must be continuous." },
                    {         ERR_PAYMENT_MISSING_DID_KEY,
          "Payment missing District Key \"DID*\"."
                                        },
                    {         ERR_PAYMENT_DISTRICT_RANGE,
          "Payment District Out of range, range = 1 - 10."
                    },
                    {         ERR_PAYMENT_MISSING_CID_KEY,
          "Payment missing Customer Key \"CID*\"."
                                        },
                    {         ERR_PAYMENT_CUSTOMER_INVALID,
          "Payment Customer data type invalid, must be numeric."
                    },
                    {         ERR_PAYMENT_MISSING_CLT,
          "Payment missing Customer Last Name Key \"CLT*\"."
                              },
                    {         ERR_PAYMENT_LAST_NAME_TO_LONG,
          "Payment Customer last name longer than 16 characters."
                    },
                    {         ERR_PAYMENT_CUSTOMER_RANGE,
          "Payment Customer ID out of range, must be 1 to 3000."
          },
                    {         ERR_PAYMENT_CID_AND_CLT,
          "Payment Customer ID and Last Name entered must be one or other."     },
                    {         ERR_PAYMENT_MISSING_CDI_KEY,
          "Payment missing Customer district key \"CDI*\"."
                    },
                    {         ERR_PAYMENT_CDI_INVALID,
          "Payment Customer district invalid must be numeric."
                    },
```

```
          {         ERR_PAYMENT_CDI_RANGE,
          "Payment Customer district out of range must be 1 - 10."
          },
          {         ERR_PAYMENT_MISSING_CWI_KEY,
"Payment missing Customer Warehouse key \"CWI*\"."
                    },
          {         ERR_PAYMENT_CWI_INVALID,
"Payment Customer Warehouse invalid must be numeric."
          },
          {         ERR_PAYMENT_CWI_RANGE,
          "Payment Customer Warehouse out of range, 1 to Max Warehouses."
          },
          {         ERR_PAYMENT_MISSING_HAM_KEY,
"Payment missing Amount key \"HAM*\"."
                              },
          {         ERR_PAYMENT_HAM_INVALID,
"Payment Amount invalid data type must be numeric."
                    },
          {         ERR_PAYMENT_HAM_RANGE,
          "Payment Amount out of range, 0 - 9999.99."
                                        },
          {         ERR_ORDERSTATUS_MISSING_DID_KEY,
"Order Status missing District key \"DID*\"."
                    },
          {         ERR_ORDERSTATUS_DID_INVALID,
"Order Status District invalid, value must be numeric 1 - 10."
},
          {         ERR_ORDERSTATUS_DID_RANGE,
"Order Status District out of range must be 1 - 10."
          },
          {         ERR_ORDERSTATUS_MISSING_CID_KEY,
"Order Status missing Customer key \"CID*\"."
                    },
          {         ERR_ORDERSTATUS_MISSING_CLT_KEY,
"Order Status missing Customer Last Name key \"CLT*\"."
          },
          {         ERR_ORDERSTATUS_CLT_RANGE,
"Order Status Customer last name longer than 16 characters."
},
          {         ERR_ORDERSTATUS_CID_INVALID,
"Order Status Customer ID invalid, range must be numeric 1 - 3000."   },
          {         ERR_ORDERSTATUS_CID_RANGE,
"Order Status Customer ID out of range must be 1 - 3000."
},
          {         ERR_ORDERSTATUS_CID_AND_CLT,
"Order Status Customer ID and LastName entered must be only one."     },
          {         ERR_DELIVERY_MISSING_OCD_KEY,
"Delivery missing Carrier ID key \"OCD*\"."
                              },
          {         ERR_DELIVERY_CARRIER_INVALID,
"Delivery Carrier ID invalid must be numeric 1 - 10."
          },
          {         ERR_DELIVERY_CARRIER_ID_RANGE,
"Delivery Carrier ID out of range must be 1 - 10."
                    },
          {         ERR_PAYMENT_MISSING_CLT_KEY,
"Payment missing Customer Last Name key \"CLT*\"."
                    },
          {         ERR_TPINIT_BAD,
                    "Tuxedo tpinit Failed."

},
                {       ERR_TPALLOC_BAD,
                "Tuxedo tpalloc Failed."
                                                                                },
                {       ERR_TPCALL_BAD,
                        "Tuxedo tpcall Failed."

        },
                {       0,
                                        ""

                                                        }
        };

        static char szNoMsg[] = "";
        char            *szForm;

        if ( !szMsg  )
                szMsg = szNoMsg;

        if ( iTermId > 0 && IsValidTermId(iTermId) )
                szForm = Term.pClientData[iTermId].szBuffer; //if termid valid
use common terminal static buffer.
        else
                szForm = Term.pClientData[0].szBuffer;  //else term id invalid
so use common terminal static buffer.
        switch(iErrorType)
        {
                case ERR_TYPE_WEBDLL:
                        for(i=0; errorMsgs[i].szMsg[0]; i++)
                        {
                                if ( iError == errorMsgs[i].iError )
                                        break;
                        }
                        if ( !errorMsgs[i].szMsg[0] )
                                i = 1;
                        strcpy(szForm,      "<HTML><HEAD><TITLE>Welcome To
TPC-C</TITLE></HEAD><BODY><FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">");
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">", iError);
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"%d\">", iErrVal);
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">", iTermId);
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">", iSyncId);
                        wsprintf(szForm+strlen(szForm), "Error: TPCCWEB(%d):
%s", iError, errorMsgs[i].szMsg);
                        strcat(szForm, "</FORM><BODY></HTML>");
                        WriteZString(pECB, szForm);
                        break;
                case ERR_TYPE_SQL:
                        strcpy(szForm,      "<HTML><HEAD><TITLE>Welcome To
TPC-C</TITLE></HEAD><BODY><FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">");
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">", iError);
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"%d\">", iErrVal);
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">", iTermId);
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">", iSyncId);
                        wsprintf(szForm+strlen(szForm), "Error: SQLSVR(%d):
%s", iError, szMsg);
                        strcat(szForm, "</FORM><BODY></HTML>");
                        WriteZString(pECB, szForm);
                        break;
                case ERR_TYPE_DBLIB:
                        strcpy(szForm,      "<HTML><HEAD><TITLE>Welcome To
TPC-C</TITLE></HEAD><BODY><FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">");
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">", iError);
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"%d\">", iErrVal);
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">", iTermId);
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">", iSyncId);
                        wsprintf(szForm+strlen(szForm), "Error: DBLIB(%d):
%s", iError, szMsg);
                        strcat(szForm, "</FORM><BODY></HTML>");
                        WriteZString(pECB, szForm);
                        break;
                case ERR_TYPE_TUXEDO:
                        strcpy(szForm,      "<HTML><HEAD><TITLE>Welcome To
TPC-C</TITLE></HEAD><BODY><FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">");
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">", iError);
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"%d\">", iErrVal);
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">", iTermId);
                        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">", iSyncId);
                        wsprintf(szForm+strlen(szForm), "Error: Tuxedo: %d
%s", iError, szMsg);
                        strcat(szForm, "</FORM><BODY></HTML>");
                        WriteZString(pECB, szForm);
                        break;
        }
        return;
}

/* FUNCTION: BOOL GetKeyValue(char *pQueryString, char *pKey, char *pValue, int
iMax)
 *
 * PURPOSE:         This function parses a http formatted string for specific key
values.
 *
 * ARGUMENTS:       char                    *pQueryString       http string
from client browser
 *                          char                        *pKey
                    key value to look for
 *                          char                        *pValue
                    character array into which to place key's value
 *                          int                                 iMax
                            maximum length of key value array.
 *
 * RETURNS:                 BOOL      FALSE     key value not found
 *                                                        TRUE      key valud
found
 *
 *
 * COMMENTS:        http keys are formatted either KEY=value& or KEY=value\0. This
DLL formats
 *                                  TPC-C input fields in such a manner that the
keys can be extracted in the
```

```
 *                                      above manner.
 */

static BOOL GetKeyValue(char *pQueryString, char *pKey, char *pValue, int iMax)
{
          char *ptr;


          if ( !(ptr=strstr(pQueryString, pKey)) )
                    return FALSE;
          if ( !(ptr=strchr(ptr, '=')) )
                    return FALSE;
          ptr++;
          iMax--;
          while( *ptr && *ptr != '&' && iMax)
          {
                    *pValue++ = *ptr++;
                    iMax--;
          }
          *pValue = 0;
          return TRUE;
}

/* FUNCTION: void TermInit(void)
 *
 * PURPOSE:         This function initializes the client ternimal structure it is
called when the TPCC.DLL
 *                            is first loaded by the inet service.
 *
 * ARGUMENTS:       none
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

static void TermInit(void)
{
          if ( Term.bInit )
                    return;
          Term.iNext                              = 0;
          Term.iMasterSyncId  = 1;

          Term.iAvailable               = 0;
          Term.pClientData    = NULL;
          Term.bInit                              = TRUE;

          return;
}


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

```
 *
 */

static void TermRestore(void)
{
          Term.iNext                              = 0;
          Term.iAvailable               = 0;
          Term.iMasterSyncId  = 0;
          if ( Term.pClientData )
                    free(Term.pClientData);
          Term.pClientData    = NULL;
          Term.bInit                              = FALSE;


          return;
}

/* FUNCTION: int TermAllocate(void)
 *
 * PURPOSE:         This function allocates more terminal array entries in the Term
structure.
 *
 * ARGUMENTS:   None
 *
 * RETURNS:                   int       TRUE or 1 if sucessfull
 *                                      int       FALSE or 0 if terminal id cannot
be allocated.
 *
 * COMMENTS:        None
 *
 */

static int TermAllocate(void)
{
          Term.iAvailable += 32;
          if ( !Term.pClientData )
                    Term.pClientData = (PCLIENTDATA)malloc(Term.iAvailable *
sizeof(CLIENTDATA));
          else
                    Term.pClientData = (PCLIENTDATA)realloc(Term.pClientData,
Term.iAvailable * sizeof(CLIENTDATA));
          return ( Term.pClientData ) ? 1 : 0;
}

/* FUNCTION: int TermAdd(EXTENSION_CONTROL_BLOCK *pECB, char *pQueryString)
 *
 * PURPOSE:         This function assigns a terminal id which is used to identify a
client browser.
 *
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB
          passed in structure pointer from inetsrv.
 *                                      char
          *pQueryString       http query string passed to this DLL.
 *
 * RETURNS:                   int                 assigned terminal id
 *                                                -1        cannot assign id error
occured.
 *
 *
 * COMMENTS:        if the terminal id cannot be assigned it is because of
insufficient memory or the
 *                                      SQL connection cannot be allocated.
 *
 */

static int TermAdd(EXTENSION_CONTROL_BLOCK *pECB, char *pQueryString)
{
          char      szTmp[32];
          int                 i, iCurrent, iTotalConnections, iTickCount;

          EnterCriticalSection(&CriticalSection);

          for(i=0, iTotalConnections = 0; i<Term.iAvailable; i++)
          {
                    if ( Term.pClientData[i].inUse )
                              iTotalConnections++;
          }

          if ( iTotalConnections >= iMaxConnections )
          {
                    for(iCurrent = 1, i=1, iTickCount = 0x7FFFFFFF;
i<iMaxConnections; i++)
                    {
                              if ( iTickCount > Term.pClientData[i].iTickCount )
                              {
                                        iTickCount = Term.pClientData[i].iTickCount;
                                        iCurrent = i;
                              }
                    }
          }
          else
          {
                    for(i=0; i<Term.iAvailable; i++)
                    {
                              if ( !Term.pClientData[i].inUse )
                                        break;
                    }
                    iCurrent = i;
          }

          if ( i == Term.iAvailable )
          {
                    Term.iNext = Term.iAvailable;
                    if ( !(*Term.Allocate)() )
                              goto TermAddErr1;
                    for(i=Term.iNext; i<Term.iAvailable; i++)
                              Term.pClientData[i].inUse = 0;
                    iCurrent = Term.iNext;
          }

          Term.pClientData[iCurrent].inUse = 1;

          if ( !GetKeyValue(pQueryString, "w_id", szTmp, sizeof(szTmp)) )
                    goto TermAddErr1;

          Term.pClientData[iCurrent].w_id = (short)atoi(szTmp);

          if ( !GetKeyValue(pQueryString, "d_id", szTmp, sizeof(szTmp)) )
                    goto TermAddErr1;

          Term.pClientData[iCurrent].d_id = atoi(szTmp);

          Term.pClientData[iCurrent].iTickCount = GetTickCount();
          Term.pClientData[iCurrent].iSyncId = Term.iMasterSyncId++;

          if ( Init(pECB, iCurrent, Term.pClientData[iCurrent].iSyncId, szServer,
szUser, szPassword, szDatabase) )
          {
                    (*Term.Delete)(pECB, iCurrent);
                    goto TermAddErr1;
          }

          LeaveCriticalSection(&CriticalSection);
          return iCurrent;

TermAddErr1:
          LeaveCriticalSection(&CriticalSection);
          return -1;          //terminal unsuccessfully added
}

/* FUNCTION: void TermDelete(EXTENSION_CONTROL_BLOCK *pECB, int id)
 *
 * PURPOSE:         This function makes a terminal entry in the Term array available
for reuse.
 *
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB     passed in structure
pointer from inetsrv.
 *                                      int
                    id                  Terminal id of client exiting
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

static void TermDelete(EXTENSION_CONTROL_BLOCK *pECB, int id)
{
          if ( id >= 0 && id < Term.iAvailable )
          {
                    Close(pECB, id, -1);
                    Term.pClientData[id].inUse = 0;
#ifndef LOCAL_ALLOC
                    tpfree((char *)Term.pClientData[id].NewOrderDataPtr);
                    tpfree((char *)Term.pClientData[id].PaymentDataPtr);
                    tpfree((char *)Term.pClientData[id].OrderStatusDataPtr);
                    tpfree((char *)Term.pClientData[id].DeliveryDataPtr);
                    tpfree((char *)Term.pClientData[id].StockLevelDataPtr);
#endif
          }

          return;
}

/* FUNCTION: BOOL Init(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int iSyncId, char
*szServer, char *szUser, char *szPassword, char *szDatabase)
 *
 * PURPOSE:         This function initializes the sql connection for use.
 *
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB               passed in
structure pointer from inetsrv.
 *                                      int
                    iTermId             id of browser client that this connection is
for.
 *                                      int
                    iSyncId             sync id for this client session
 *                                      char
          *szServer sql server name
 *                                      char
          *szUser             user name
 *                                      char
          *szPassword         user password
 *                                      char
          *szDatabase         database to use
 *
 * RETURNS:                   BOOL      FALSE     if successfull
 *                                                          TRUE      if an error
occurs and connection cannot be established.
 *
 * COMMENTS:          None
 *
 */

BOOL Init(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int iSyncId, char *szServer,
char *szUser, char *szPassword, char *szDatabase)
{
          char      szApp[32];
#ifndef LOCAL_ALLOC
          char      buf[40];
          int                 iRc;
#endif

          sprintf(szApp, "TPCC:%ld", (int)iTermId);

          Term.pClientData[iTermId].dbproc = NULL;

/*        if ( SQLOpenConnection(pECB, iTermId, iSyncId,
&Term.pClientData[iTermId].dbproc, szServer, szDatabase, szUser, szPassword, szApp,
&Term.pClientData[iTermId].spid) )
          {
                    ErrorMessage(pECB, ERR_SQL_OPEN_CONNECTION, ERR_TYPE_WEBDLL,
NULL, iTermId, iSyncId);
                    return TRUE;
          }
*/

#ifndef LOCAL_ALLOC
//        Add initialization of Tuxedo Structures

//                  NEWORDER

                    if ((Term.pClientData[iTermId].NewOrderDataPtr = (NEW_ORDER_DATA
*)tpalloc("CARRAY", NULL, sizeof(NEW_ORDER_DATA))) == NULL)
                    {
                              iRc = tperrno;
                              sprintf(buf, "Tpcalloc %d", iRc);
                              ErrorMessage(pECB, ERR_TPALLOC_BAD, ERR_TYPE_TUXEDO,
buf, 0, 0);
                              {
                                        FILE *fp;

                                        fp = fopen(szTpccLogPath, "ab");
                                        fprintf(fp, ">>>> Init Thread %d : NewOrder
tpalloc failed: iRc = %d \r\n",
                                                  GetCurrentThreadId(), iRc);
                                        fclose(fp);
                              }
                              sprintf(buf, "Tpcalloc %d", iRc);
                              ErrorMessage(pECB, ERR_TPALLOC_BAD, ERR_TYPE_TUXEDO,
buf, 0, 0);
                              return TRUE;
```

```
                    }

                    if ( dLog )
                    {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* Init Thread %d iTermId %d *
NewOrderDataPtr: %x \r\n",
                                        GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].NewOrderDataPtr);
                              fclose(fp);
                    }

//                  PAYMENT

                    if ((Term.pClientData[iTermId].PaymentDataPtr = (PAYMENT_DATA
*)tpalloc("CARRAY", NULL, sizeof(PAYMENT_DATA))) == NULL)
                    {
                              iRc = tperrno;
                              {
                                        FILE *fp;

                                        fp = fopen(szTpccLogPath, "ab");
                                        fprintf(fp, ">>>> Init Thread %d : Payment
tpalloc failed: iRc = %d \r\n",
                                                  GetCurrentThreadId(), iRc);
                                        fclose(fp);
                              }
                              sprintf(buf, "Tpcalloc %d", iRc);
                              ErrorMessage(pECB, ERR_TPALLOC_BAD, ERR_TYPE_TUXEDO,
buf, 0, 0);
                              return TRUE;
                    }

                    if ( dLog )
                    {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* Init Thread %d iTermId %d *
PaymentDataPtr: %x \r\n",
                                        GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].PaymentDataPtr);
                              fclose(fp);
                    }

//                  ORDERSTATUS

                    if ((Term.pClientData[iTermId].OrderStatusDataPtr =
(ORDER_STATUS_DATA *)tpalloc("CARRAY", NULL, sizeof(ORDER_STATUS_DATA))) == NULL)
                    {
                              iRc = tperrno;
                              {
                                        FILE *fp;

                                        fp = fopen(szTpccLogPath, "ab");
                                        fprintf(fp, ">>>> Init Thread %d :
OrderStatus tpalloc failed: iRc = %d \r\n",
                                                  GetCurrentThreadId(), iRc);
                                        fclose(fp);
                              }
                              sprintf(buf, "Tpcalloc %d", iRc);
                              ErrorMessage(pECB, ERR_TPALLOC_BAD, ERR_TYPE_TUXEDO, 
buf, 0, 0);
                              return TRUE;
                    }

                    if ( dLog )
                    {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* Init Thread %d iTermId %d * 
OrderStatusDataPtr: %x \r\n",
                                        GetCurrentThreadId(), iTermId, 
&Term.pClientData[iTermId].OrderStatusDataPtr);
                              fclose(fp);
                    }

//                  DELIVERY

                    if ((Term.pClientData[iTermId].DeliveryDataPtr = (DELIVERY_DATA 
*)tpalloc("CARRAY", NULL, sizeof(DELIVERY_DATA))) == NULL)
                    {
                              iRc = tperrno;
                              {
                                        FILE *fp;

                                        fp = fopen(szTpccLogPath, "ab");
                                        fprintf(fp, ">>>> Init Thread %d : Delivery 
tpalloc failed: iRc = %d \r\n",
                                                  GetCurrentThreadId(), iRc);
                                        fclose(fp);
                              }
                              sprintf(buf, "Tpcalloc %d", iRc);
                              ErrorMessage(pECB, ERR_TPALLOC_BAD, ERR_TYPE_TUXEDO, 
buf, 0, 0);
                              return TRUE;
                    }

                    if ( dLog )
                    {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");

                              fprintf(fp, "* Init Thread %d iTermId %d * 
DeliveryDataPtr: %x \r\n",
                                        GetCurrentThreadId(), iTermId, 
&Term.pClientData[iTermId].DeliveryDataPtr);
                              fclose(fp);
                    }

//                  STOCKLEVEL

                    if ((Term.pClientData[iTermId].StockLevelDataPtr = 
(STOCK_LEVEL_DATA *)tpalloc("CARRAY", NULL, sizeof(STOCK_LEVEL_DATA))) == NULL)
                    {
                              iRc = tperrno;
                              {
                                        FILE *fp;

                                        fp = fopen(szTpccLogPath, "ab");
                                        fprintf(fp, ">>>> Init Thread %d : 
StockLevel tpalloc failed: iRc = %d \r\n",
                                                  GetCurrentThreadId(), iRc);
                                        fclose(fp);
                              }
                              sprintf(buf, "Tpcalloc %d", iRc);
                              ErrorMessage(pECB, ERR_TPALLOC_BAD, ERR_TYPE_TUXEDO, 
buf, 0, 0);
                              return TRUE;
                    }

                    if ( dLog )
                    {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");

                              fprintf(fp, "* Init Thread %d iTermId %d * 
StockLevelDataPtr: %x \r\n",
                                        GetCurrentThreadId(), iTermId, 
&Term.pClientData[iTermId].StockLevelDataPtr);
                              fclose(fp);
                    }
#endif

          return FALSE;
}

/* FUNCTION: BOOL Close(EXTENSION_CONTROL_BLOCK   *pECB, int iTermId, int iSyncId)
 *
 * PURPOSE:         This function closes the sql connection for use.
 *
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB     passed in structure 
pointer from inetsrv.
 *                                      int 
                    iTermId   id of browser client that this connection is for.
 *                                      int 
                    iSyncId   sync id of client browser
 *
 * RETURNS:                   BOOL      FALSE     if successfull
 *                                                          TRUE      if an error 
occurs and connection cannot be terminated.
 *
 * COMMENTS:        None
 *
 */

static BOOL Close(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int iSyncId)
{
          PECBINFO  pEcbInfo;

          if (Term.pClientData[iTermId].dbproc != NULL)
          {
                    if ( (pEcbInfo = 
(PECBINFO)dbgetuserdata(Term.pClientData[iTermId].dbproc)) )
                    {
                              pEcbInfo->iTermId = -1;
                              pEcbInfo->iSyncId = -1;
                              free(pEcbInfo);     //free up user info
                    }
//                  return SQLCloseConnection(pECB, 
Term.pClientData[iTermId].dbproc);
                    return 1;
          }
          UNUSEDPARAM(iSyncId);
}


/* FUNCTION: void FormatString(char *szDest, char *szPic, char *szSrc)
 *
 * PURPOSE:         This function formats a character string for inclusion in the
 *                            HTML formatted page being constructed.
 *
 * ARGUMENTS:       char                *szDest   Destination buffer where formatted
string is to be placed
 *                                      char                *szPic    picture string
which describes how character value is to be
 *
          formatted.
 *                                      char                *szSrc    character
string value.
 *
 * RETURNS:                   None
 *
 * COMMENTS:        This functions is used to format TPC-C phone and zip value
strings.
 *
 */

static void FormatString(char *szDest, char *szPic, char *szSrc)
{
          while( *szPic )
          {
                    if ( *szPic == ’X’ )
                    {
                              if ( *szSrc )
                                        *szDest++ = *szSrc++;
                              else
                                        *szDest++ = ’ ’;
                    }
                    else
                              *szDest++ = *szPic;
                    szPic++;
          }
          *szDest = 0;

          return;
}

/* FUNCTION: char *MakeStockLevelForm(int iTermId, int iSyncId, BOOL bInput)
 *
 * PURPOSE:         This function constructs the Stock Level HTML page.
 *
 * ARGUMENTS:       int                                    iTermId   client browser
terminal id
 *                                      int
          iSyncId   client browser sync id
 *                                      BOOL                          bInput    TRUE
if form is being constructed for input else FALSE
 *
 * RETURNS:                   char *                        A pointer to buffer
inside client structure where HTML form is built.
 *
 * COMMENTS:        The internal client buffer is created when the terminal id is
assigned and should not
 *                                      be freed except when the client terminal id
is no longer needed.
```

```
 */

static char *MakeStockLevelForm(int iTermId, int iSyncId, BOOL bInput)
{
          char      *szForm;

          szForm = (char *)Term.pClientData[iTermId].szBuffer;

          Term.pClientData[iTermId].StockLevelData.w_id                          =
(short)Term.pClientData[iTermId].w_id;
          Term.pClientData[iTermId].StockLevelData.d_id                          =
(short)Term.pClientData[iTermId].d_id;
          Term.pClientData[iTermId].StockLevelData.num_deadlocks      = 0;

          if ( bInput)
          {
                    strcpy(szForm, "<HTML><HEAD><TITLE>TPC-C Stock
Level</TITLE></HEAD>\
<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"PI*\" VALUE=\"\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"7\">"); // STOCK_LEVEL_FORM = 7

                    wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">", iTermId, iSyncId);

                    strcat(szForm, "<PRE>                               Stock-
Level<BR>");

                    wsprintf(szForm+strlen(szForm), "Warehouse: %4.4d   District:
%2.2d<BR><BR>", Term.pClientData[iTermId].StockLevelData.w_id,
Term.pClientData[iTermId].StockLevelData.d_id);

                    strcat(szForm,      "Stock Level Threshold: <INPUT NAME=\"TT*\"
SIZE=2><BR><BR>\
low stock:    <BR><HR>\
<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\">\
<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">\
</FORM></HTML>");

          }
          else // Not bInput
          {

                    strcpy(szForm, "<HTML><HEAD><TITLE>TPC-C Stock
Level</TITLE></HEAD>\
<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"7\">"); // STOCK_LEVEL_FORM = 7

                    wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">", iTermId, iSyncId);

                    strcat(szForm, "<PRE>                               Stock-
Level<BR>");

                    wsprintf(szForm+strlen(szForm), "Warehouse: %4.4d   District:
%2.2d<BR><BR>\
Stock Level Threshold: %2.2d<BR><BR>low stock: %3.3d</PRE><BR><HR>",
          Term.pClientData[iTermId].StockLevelData.w_id,

          Term.pClientData[iTermId].StockLevelData.d_id,

          Term.pClientData[iTermId].StockLevelData.thresh_hold,

          Term.pClientData[iTermId].StockLevelData.low_stock);

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

          return szForm;
}

/* FUNCTION: char *MakeMainMenuForm(int iTermId, int iSyncId)
 *
 * PURPOSE:         This function
 *
 * ARGUMENTS:       int                                     iTermId   client browser
terminal id
 *                                      int
          iSyncId   client browser sync id
 *
 * RETURNS:                   char *                        A pointer to buffer
inside client structure where HTML form is built.
 *
 * COMMENTS:        The internal client buffer is created when the terminal id is
assigned and should not
 *                                      be freed except when the client terminal id
is no longer needed.
 */

static char *MakeMainMenuForm(int iTermId, int iSyncId)
{
          char      *szForm;

          szForm = (char *)Term.pClientData[iTermId].szBuffer;

          strcpy(szForm,      "<HTML><HEAD><TITLE>TPC-C Main
Menu</TITLE></HEAD><BODY>"
                                                  "Select Desired
Transaction.<BR><HR>"
                                                  "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">");
          strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">");
          strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">");
          wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\" NAME=\"TERMID\"
VALUE=\"%d\">", iTermId);
          wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\"
VALUE=\"%d\">", iSyncId);
          wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">", MAIN_MENU_FORM);
          strcat(szForm,      "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
```

```
                                                  "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
                                                  "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
                                                  "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                                                  "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                                                  "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"
                                                  "</FORM>"
                                                  "</HTML>" );

          return szForm;
}

/* FUNCTION: char *MakeWelcomeForm(void)
 *
 * PURPOSE:         This function
 *
 * ARGUMENTS:   None
 *
 * RETURNS:                   char *                        A pointer to the static
HTML welcome form.
 *
 * COMMENTS:        The welcome form is static.
 */

static char *MakeWelcomeForm(void)
{
          return szWelcomeForm;
}

/* FUNCTION: char *MakeNewOrderForm(int iTermId, BOOL bInput, BOOL bValid)
 *
 * PURPOSE:         This function
 *
 * ARGUMENTS:       int                                     iTermId   client browser
terminal id
 *                                      int
          iSyncId   client browser sync id
 *                                      BOOL                          bInput    TRUE
if form is being constructed for input else FALSE
 *                                      BOOL                          bValid    TRUE
if NewOrderData valid, ELSE FALSE effects output only
 *
 * RETURNS:                   char *                        A pointer to buffer
inside client structure where HTML form is built.
 *
 * COMMENTS:        The internal client buffer is created when the terminal id is
assigned and should not
 *                                      be freed except when the client terminal id
is no longer needed.
 */

static char *MakeNewOrderForm(int iTermId, int iSyncId, BOOL bInput, BOOL bValid)
{
          char      *szForm;
//        char      szName[146];
//        char      szCredit[14];
          int                 i;

          szForm = (char *)Term.pClientData[iTermId].szBuffer;
          Term.pClientData[iTermId].NewOrderData.w_id =
Term.pClientData[iTermId].w_id;

          if ( bInput)
          {
                    strcpy(szForm,"<HTML><HEAD><TITLE>TPC-C New Order</TITLE>\
</HEAD><BODY><FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"PI*\" VALUE=\"\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"3\">"); // NEW_ORDER_FORM = 3

                    sprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">\
<PRE>                                New Order<BR>\
Warehouse: %4.4d   District: <INPUT NAME=\"DID*\" SIZE=1>
Date:<BR>\
Customer:  <INPUT NAME=\"CID*\" SIZE=4>   Name:                    Credit:
%Disc:<BR>\
Order Number:           Number of Lines:           W_tax:         D_tax:<BR><BR>\
 Supp_W  Item_Id  Item Name                 Qty  Stock  B/G  Price    Amount<BR>",
                                                  iTermId,
                                                  iSyncId,

          Term.pClientData[iTermId].NewOrderData.w_id);

                    strcat(szForm,      " <INPUT NAME=\"SP00*\" SIZE=4>  <INPUT
NAME=\"IID00*\" SIZE=6>                           <INPUT NAME=\"Qty00*\"
SIZE=1><BR>\
 <INPUT NAME=\"SP01*\" SIZE=4>  <INPUT NAME=\"IID01*\" SIZE=6>
<INPUT NAME=\"Qty01*\" SIZE=1><BR>\
 <INPUT NAME=\"SP02*\" SIZE=4>  <INPUT NAME=\"IID02*\" SIZE=6>
<INPUT NAME=\"Qty02*\" SIZE=1><BR>\
 <INPUT NAME=\"SP03*\" SIZE=4>  <INPUT NAME=\"IID03*\" SIZE=6>
<INPUT NAME=\"Qty03*\" SIZE=1><BR>\
 <INPUT NAME=\"SP04*\" SIZE=4>  <INPUT NAME=\"IID04*\" SIZE=6>
<INPUT NAME=\"Qty04*\" SIZE=1><BR>\
 <INPUT NAME=\"SP05*\" SIZE=4>  <INPUT NAME=\"IID05*\" SIZE=6>
<INPUT NAME=\"Qty05*\" SIZE=1><BR>\
 <INPUT NAME=\"SP06*\" SIZE=4>  <INPUT NAME=\"IID06*\" SIZE=6>
<INPUT NAME=\"Qty06*\" SIZE=1><BR>\
 <INPUT NAME=\"SP07*\" SIZE=4>  <INPUT NAME=\"IID07*\" SIZE=6>
<INPUT NAME=\"Qty07*\" SIZE=1><BR>\
 <INPUT NAME=\"SP08*\" SIZE=4>  <INPUT NAME=\"IID08*\" SIZE=6>
<INPUT NAME=\"Qty08*\" SIZE=1><BR>\
 <INPUT NAME=\"SP09*\" SIZE=4>  <INPUT NAME=\"IID09*\" SIZE=6>
<INPUT NAME=\"Qty09*\" SIZE=1><BR>\
 <INPUT NAME=\"SP10*\" SIZE=4>  <INPUT NAME=\"IID10*\" SIZE=6>
<INPUT NAME=\"Qty10*\" SIZE=1><BR>\
 <INPUT NAME=\"SP11*\" SIZE=4>  <INPUT NAME=\"IID11*\" SIZE=6>
<INPUT NAME=\"Qty11*\" SIZE=1><BR>\
 <INPUT NAME=\"SP12*\" SIZE=4>  <INPUT NAME=\"IID12*\" SIZE=6>
<INPUT NAME=\"Qty12*\" SIZE=1><BR>\
 <INPUT NAME=\"SP13*\" SIZE=4>  <INPUT NAME=\"IID13*\" SIZE=6>
<INPUT NAME=\"Qty13*\" SIZE=1><BR>\
 <INPUT NAME=\"SP14*\" SIZE=4>  <INPUT NAME=\"IID14*\" SIZE=6>
<INPUT NAME=\"Qty14*\" SIZE=1><BR>\
Execution Status:                                            Total:<BR><HR>\
<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\">\
<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">\
</FORM></HTML>" );
```

```
          }
          else // not bInput
          {
                    if (bValid)
                    {

//                  FormatHTMLString(szName,
Term.pClientData[iTermId].NewOrderData.c_last, 16),
//                  FormatHTMLString(szCredit,
Term.pClientData[iTermId].NewOrderData.c_credit, 2);

                    strcpy(szForm,"<HTML><HEAD><TITLE>TPC-C New Order</TITLE>\
</HEAD><BODY><FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"3\">"); // NEW_ORDER_FORM = 3

sprintf(szForm+strlen(szForm),"<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">\
<PRE>                                New Order<BR>\
Warehouse: %4.4d   District: %2.2d                        Date: %2.2d-%2.2d-%4.4d
%2.2d:%2.2d:%2.2d <BR>\
Customer:  %4.4d   Name: %s   Credit: %s   %%Disc: %5.2f            <BR>\
Order Number: %8.8d  Number of Lines: %2.2d        W_tax: %5.2f   D_tax: %5.2f
<BR><BR>\
 Supp_W  Item_Id  Item Name                 Qty  Stock  B/G  Price    Amount<BR>",
                                        iTermId,
                                        iSyncId,
                                        Term.pClientData[iTermId].NewOrderData.w_id,
                                        Term.pClientData[iTermId].NewOrderData.d_id,

          Term.pClientData[iTermId].NewOrderData.o_entry_d.day,

          Term.pClientData[iTermId].NewOrderData.o_entry_d.month,

          Term.pClientData[iTermId].NewOrderData.o_entry_d.year,

          Term.pClientData[iTermId].NewOrderData.o_entry_d.hour,

          Term.pClientData[iTermId].NewOrderData.o_entry_d.minute,

          Term.pClientData[iTermId].NewOrderData.o_entry_d.second,
                                        Term.pClientData[iTermId].NewOrderData.c_id,

          Term.pClientData[iTermId].NewOrderData.c_last,

          Term.pClientData[iTermId].NewOrderData.c_credit,

          Term.pClientData[iTermId].NewOrderData.c_discount*100.0,
                                        Term.pClientData[iTermId].NewOrderData.o_id,

          Term.pClientData[iTermId].NewOrderData.o_ol_cnt,

          Term.pClientData[iTermId].NewOrderData.w_tax*100.0,

          Term.pClientData[iTermId].NewOrderData.d_tax*100.0);

// We could make 0-4 be hard coded in one sprintf, since we know that there is at
least 5
// order lines, or is this a benchmark special??
// etw
                    for(i=0; i<Term.pClientData[iTermId].NewOrderData.o_ol_cnt; i++)
```

```
                              {
//                                        FormatHTMLString(szName,
Term.pClientData[iTermId].NewOrderData.Ol[i].ol_i_name, 24);

                                        sprintf(szForm+strlen(szForm), "  %4.4d
%6.6d   %-24s  %2.2d    %3.3d    %1.1s   $%6.2f  $%7.2f  <BR>",

          Term.pClientData[iTermId].NewOrderData.Ol[i].ol_supply_w_id,

          Term.pClientData[iTermId].NewOrderData.Ol[i].ol_i_id,

          Term.pClientData[iTermId].NewOrderData.Ol[i].ol_i_name,

          Term.pClientData[iTermId].NewOrderData.Ol[i].ol_quantity,

          Term.pClientData[iTermId].NewOrderData.Ol[i].ol_stock,

          Term.pClientData[iTermId].NewOrderData.Ol[i].ol_brand_generic,

          Term.pClientData[iTermId].NewOrderData.Ol[i].ol_i_price,

          Term.pClientData[iTermId].NewOrderData.Ol[i].ol_amount );
                              }
                    for(; i<15; i++)
                              strcat(szForm, "<BR>");

                              sprintf(szForm+strlen(szForm), "Execution Status:
%24.24s                   Total:  $%8.2f  ",

          Term.pClientData[iTermId].NewOrderData.execution_status,

          Term.pClientData[iTermId].NewOrderData.total_amount);

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

                    }
                    else // not bValid
                    {

//                  FormatHTMLString(szName,
Term.pClientData[iTermId].NewOrderData.c_last, 16),
//                  FormatHTMLString(szCredit,
Term.pClientData[iTermId].NewOrderData.c_credit, 2);

                    strcpy(szForm,"<HTML><HEAD><TITLE>TPC-C New Order</TITLE>\
</HEAD><BODY><FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"1\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"1\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"3\">");  // NEW_ORDER_FORM = 3

                    sprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">\
<PRE>                                New Order<BR>\
Warehouse: %4.4d   District: %2.2d                     Date:<BR>\
Customer:  %4.4d   Name: %s   Credit: %s   %Disc:<BR>\
Order Number: %8.8d  Number of Lines:             W_tax:         D_tax:<BR><BR>",
                                        iTermId,
                                        iSyncId,
                                        Term.pClientData[iTermId].NewOrderData.w_id,
                                        Term.pClientData[iTermId].NewOrderData.d_id,
                                        Term.pClientData[iTermId].NewOrderData.c_id,

          Term.pClientData[iTermId].NewOrderData.c_last,

          Term.pClientData[iTermId].NewOrderData.c_credit,
                                        Term.pClientData[iTermId].NewOrderData.o_id
                                        );

                    strcat(szForm, " Supp_W  Item_Id  Item Name                 Qty
Stock  B/G  Price    Amount<BR>\
<BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR>");

                    sprintf(szForm+strlen(szForm), "Execution Status: %24.24s
Total:",

          Term.pClientData[iTermId].NewOrderData.execution_status);

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

                    }
          }
          return szForm;
}

/* FUNCTION: char *MakePaymentForm(int iTermId, int iSyncId, BOOL bInput)
 *
 * PURPOSE:         This function
 *
 * ARGUMENTS:       int                                     iTermId   client browser
terminal id
 *                                      int
          iSyncId   client browser sync id
 *                                      BOOL                          bInput    TRUE
if form is being constructed for input else FALSE
 *
 * RETURNS:                   char *                        A pointer to buffer
inside client structure where HTML form is built.
 *
 * COMMENTS:        The internal client buffer is created when the terminal id is
assigned and should not
 *                                      be freed except when the client terminal id
is no longer needed.
 */

static char *MakePaymentForm(int iTermId, int iSyncId, BOOL bInput)
{
          char      *szForm;
          char      *ptr;
          char      szTmp[64];
          char      szW_Zip[26];
          char      szD_Zip[26];
          char      szC_Zip[26];
          char      szC_Phone[26];
//        char      szTmpStr1[122];
//        char      szTmpStr2[122];
//        char      szTmpStr3[122];
//        char      szTmpStr4[122];
          int                 i;
          int                 l;
          char      *szZipPic = "XXXXX-XXXX";

          szForm = (char *)Term.pClientData[iTermId].szBuffer;

          Term.pClientData[iTermId].PaymentData.w_id =
Term.pClientData[iTermId].w_id;

          if ( bInput )
          {
                    strcpy(szForm,      "<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD><BODY>\
<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"PI*\" VALUE=\"\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"4\">"); // PAYMENT_FORM = 4

                    sprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">\
<PRE>                                 Payment<BR>\
Date:<BR><BR>\
Warehouse: %4.4d",
                                                            iTermId,
                                                            iSyncId,

          Term.pClientData[iTermId].PaymentData.w_id);

                              strcat(szForm,
"                          District: <INPUT NAME=\"DID*\"
SIZE=1><BR><BR><BR><BR><BR>\
Customer: <INPUT NAME=\"CID*\" SIZE=4>\
Cust-Warehouse: <INPUT NAME=\"CWI*\" SIZE=4>  \
Cust-District: <INPUT NAME=\"CDI*\" SIZE=1><BR>\
Name:                       <INPUT NAME=\"CLT*\" SIZE=16>
Since:<BR>\
                                                 Credit:<BR>\
                                                 Disc:<BR>\
                                                 Phone:<BR><BR>\
Amount Paid:          $<INPUT NAME=\"HAM*\" SIZE=7>      New Cust Balance:<BR>\
Credit Limit:<BR><BR>Cust-Data: <BR><BR><BR><BR></PRE><HR>\
<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\"><INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">\
</BODY></FORM></HTML>");

          }
          else // Not bInput
          {

//                  FormatHTMLString(szTmpStr1,
Term.pClientData[iTermId].PaymentData.w_street_1, 20);
//                  FormatHTMLString(szTmpStr2,
Term.pClientData[iTermId].PaymentData.d_street_1, 20);
```

```
//                  FormatHTMLString(szTmpStr3,
Term.pClientData[iTermId].PaymentData.w_street_2, 20);
//                  FormatHTMLString(szTmpStr4,
Term.pClientData[iTermId].PaymentData.d_street_2, 20);

                    FormatString(szW_Zip, szZipPic,
Term.pClientData[iTermId].PaymentData.w_zip);
                    FormatString(szD_Zip, szZipPic,
Term.pClientData[iTermId].PaymentData.d_zip);

//                  FormatHTMLString(szTmpStr5,
Term.pClientData[iTermId].PaymentData.w_city, 20);
//                  FormatHTMLString(szTmpStr6,
Term.pClientData[iTermId].PaymentData.w_state, 2);
//                  FormatHTMLString(szTmpStr7,
Term.pClientData[iTermId].PaymentData.d_city, 20);
//                  FormatHTMLString(szTmpStr8,
Term.pClientData[iTermId].PaymentData.d_state, 2);

//                  FormatHTMLString(szTmpStr9,
Term.pClientData[iTermId].PaymentData.c_first, 16);
//                  FormatHTMLString(szTmpStr10,
Term.pClientData[iTermId].PaymentData.c_middle, 2);
//                  FormatHTMLString(szTmpStr11,
Term.pClientData[iTermId].PaymentData.c_last, 16);

//                  FormatHTMLString(szTmpStr12,
Term.pClientData[iTermId].PaymentData.c_street_1, 20);
//                  FormatHTMLString(szTmpStr13,
Term.pClientData[iTermId].PaymentData.c_credit, 2);

//                  FormatHTMLString(szTmpStr14,
Term.pClientData[iTermId].PaymentData.d_street_2, 20);

                    FormatString(szC_Zip, szZipPic,
Term.pClientData[iTermId].PaymentData.c_zip);
                    FormatString(szC_Phone, "XXXXXX-XXX-XXX-XXXX",
Term.pClientData[iTermId].PaymentData.c_phone);

//                  FormatHTMLString(szTmpStr15,
Term.pClientData[iTermId].PaymentData.c_city, 20);
//                  FormatHTMLString(szTmpStr16,
Term.pClientData[iTermId].PaymentData.c_state, 2);

                              strcpy(szForm,      "<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD><BODY>\
<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"4\">"); // PAYMENT_FORM = 4

                              sprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">\
<PRE>                                 Payment<BR>\
Date: %2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d <BR><BR>\
Warehouse: %4.4d                          District: %2.2d<BR>\
%-20s                     %-20s<BR>\
%-20s                     %-20s<BR>\
%-20s %-2s %10.10s       %-20s %-2s %10.10s<BR><BR>\
Customer: %4.4d  Cust-Warehouse: %4.4d  Cust-District: %2.2d<BR>\
Name:   %-16s %-2s %-16s     Since:  %2.2d-%2.2d-%4.4d<BR>\
        %-20s                     Credit: %s<BR>\
        %-20s                     %%Disc:  %5.2f<BR>\
        %-20s %-2s %10.10s       Phone:  %-19.19s<BR><BR>\
Amount Paid:          $%7.2f      New Cust Balance: $%14.2f<BR>\
Credit Limit:   $%13.2f<BR><BR>",

                              iTermId,
                              iSyncId,
                              Term.pClientData[iTermId].PaymentData.h_date.day,
                              Term.pClientData[iTermId].PaymentData.h_date.month,
                              Term.pClientData[iTermId].PaymentData.h_date.year,
                              Term.pClientData[iTermId].PaymentData.h_date.hour,
                              Term.pClientData[iTermId].PaymentData.h_date.minute,
                              Term.pClientData[iTermId].PaymentData.h_date.second,
                              Term.pClientData[iTermId].PaymentData.w_id,
                              Term.pClientData[iTermId].PaymentData.d_id,
                              Term.pClientData[iTermId].PaymentData.w_street_1,
                              Term.pClientData[iTermId].PaymentData.d_street_1,
                              Term.pClientData[iTermId].PaymentData.w_street_2,
                              Term.pClientData[iTermId].PaymentData.d_street_2,
                              Term.pClientData[iTermId].PaymentData.w_city,
                              Term.pClientData[iTermId].PaymentData.w_state,
                              szW_Zip,
                              Term.pClientData[iTermId].PaymentData.d_city,
                              Term.pClientData[iTermId].PaymentData.d_state,
                              szD_Zip,
                              Term.pClientData[iTermId].PaymentData.c_id,
                              Term.pClientData[iTermId].PaymentData.c_w_id,
                              Term.pClientData[iTermId].PaymentData.c_d_id,
                              Term.pClientData[iTermId].PaymentData.c_first,
                              Term.pClientData[iTermId].PaymentData.c_middle,
                              Term.pClientData[iTermId].PaymentData.c_last,
                              Term.pClientData[iTermId].PaymentData.c_since.day,
                              Term.pClientData[iTermId].PaymentData.c_since.month,
                              Term.pClientData[iTermId].PaymentData.c_since.year,
                              Term.pClientData[iTermId].PaymentData.c_street_1,
                              Term.pClientData[iTermId].PaymentData.c_credit,
                              Term.pClientData[iTermId].PaymentData.d_street_2,

          Term.pClientData[iTermId].PaymentData.c_discount*100.0,
                              Term.pClientData[iTermId].PaymentData.c_city,
                              Term.pClientData[iTermId].PaymentData.c_state,
                              szC_Zip, szC_Phone,
                              Term.pClientData[iTermId].PaymentData.h_amount,
                              Term.pClientData[iTermId].PaymentData.c_balance,
                              Term.pClientData[iTermId].PaymentData.c_credit_lim);

// This part can probably be made a little more efficient
//etw
                    ptr = Term.pClientData[iTermId].PaymentData.c_credit;
                    if ( *ptr == 'B' && *(ptr+1) == 'C' )
                    {
                              ptr = Term.pClientData[iTermId].PaymentData.c_data;
                              l = strlen( ptr ) / 50;
                              for(i=0; i<4; i++, ptr += 50)
                              {
                                        if ( i <= l )
                                                  UtilStrCpy(szTmp, ptr, 50);
                                        else
                                                  szTmp[0] = 0;
                                        if ( !i )
                                        {
//                                                FormatHTMLString(szTmpStr1, szTmp,
50);
                                                  wsprintf(szForm+strlen(szForm),
"Cust-Data: %-50s<BR>", szTmp);
                                        }
                                        else
                                        {
//                                                FormatHTMLString(szTmpStr1, szTmp,
50);
                                                  wsprintf(szForm+strlen(szForm), "
%-50s<BR>", szTmp);
                                        }
                              }
                    }
                    else
                              strcat(szForm, "Cust-Data: <BR><BR><BR><BR>");

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

          }

          return szForm;
}

/* FUNCTION: char *MakeOrderStatusForm(int iTermId, int iSyncId, BOOL bInput)
 *
 * PURPOSE:         This function
 *
 * ARGUMENTS:       int                                    iTermId   client browser
terminal id
 *                                      int
          iSyncId   client browser sync id
 *                                      BOOL                          bInput    TRUE
if form is being constructed for input else FALSE
 *
 * RETURNS:                   char *                       A pointer to buffer
inside client structure where HTML form is built.
 *
 * COMMENTS:        The internal client buffer is created when the terminal id is
assigned and should not
 *                                      be freed except when the client terminal id
is no longer needed.
 */

static char *MakeOrderStatusForm(int iTermId, int iSyncId, BOOL bInput)
{
          char      *szForm;
//        char      c_first[98];
//        char      c_middle[14];
//        char      c_last[98];
          int                 i;

          szForm = (char *)Term.pClientData[iTermId].szBuffer;

          Term.pClientData[iTermId].OrderStatusData.w_id =
Term.pClientData[iTermId].w_id;

          if ( bInput )
          {
                    strcpy(szForm,      "<HTML><HEAD><TITLE>TPC-C Order-
Status</TITLE></HEAD><BODY>\
<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"PI*\" VALUE=\"\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"6\">"); // ORDER_STATUS_FORM = 6

                    wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">",
                                                            iTermId,
                                                            iSyncId);

                    strcat(szForm,      "<PRE>                              Order-
Status<BR>" );
                    wsprintf(szForm+strlen(szForm), "Warehouse: %4.4d   ",
Term.pClientData[iTermId].OrderStatusData.w_id);

                    strcat(szForm,      "District: <INPUT NAME=\"DID*\" SIZE=1><BR>\
Customer: <INPUT NAME=\"CID*\" SIZE=4>   Name:                     <INPUT
NAME=\"CLT*\" SIZE=23><BR>\
Cust-Balance:<BR><BR>\
Order-Number:            Entry-Date:                       Carrier-Number:<BR>\
Supply-W     Item-Id    Qty     Amount      Delivery-Date<BR></PRE>\
<HR><INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\"><INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">\
</BODY></FORM></HTML>" );

          }
          else // Not bInput
          {
                    strcpy(szForm,      "<HTML><HEAD><TITLE>TPC-C Order-
Status</TITLE></HEAD><BODY>\
<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"6\">"); // ORDER_STATUS_FORM = 6

                    wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">",
                                                            iTermId,
                                                            iSyncId);
          strcat(szForm,      "<PRE>                              Order-Status<BR>"
);

          sprintf(szForm+strlen(szForm), "Warehouse: %4.4d   District: %2.2d<BR>\
Customer: %4.4d   Name: %-16s %-2s %-16s<BR>\
Cust-Balance: $%9.2f<BR><BR>\
Order-Number: %8.8d   Entry-Date: %2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d   Carrier-
Number: %2.2d<BR>",

          Term.pClientData[iTermId].OrderStatusData.w_id,

          Term.pClientData[iTermId].OrderStatusData.d_id,

          Term.pClientData[iTermId].OrderStatusData.c_id,

          Term.pClientData[iTermId].OrderStatusData.c_first,

          Term.pClientData[iTermId].OrderStatusData.c_middle,
```

```
          Term.pClientData[iTermId].OrderStatusData.c_last,

          Term.pClientData[iTermId].OrderStatusData.c_balance,

          Term.pClientData[iTermId].OrderStatusData.o_id,

          Term.pClientData[iTermId].OrderStatusData.o_entry_d.day,

          Term.pClientData[iTermId].OrderStatusData.o_entry_d.month,

          Term.pClientData[iTermId].OrderStatusData.o_entry_d.year,

          Term.pClientData[iTermId].OrderStatusData.o_entry_d.hour,

          Term.pClientData[iTermId].OrderStatusData.o_entry_d.minute,

          Term.pClientData[iTermId].OrderStatusData.o_entry_d.second,

          Term.pClientData[iTermId].OrderStatusData.o_carrier_id);

                    strcat(szForm+strlen(szForm), "Supply-W     Item-Id    Qty
Amount      Delivery-Date<BR>");

                    for(i=0; i<Term.pClientData[iTermId].OrderStatusData.o_ol_cnt;
i++)
                    {
                              sprintf(szForm+strlen(szForm), "  %4.4d       %6.6d
%2.2d     $%8.2f      %2.2d-%2.2d-%4.4d<BR>",

          Term.pClientData[iTermId].OrderStatusData.OlOrderStatusData[i].ol_supply_w
_id,

          Term.pClientData[iTermId].OrderStatusData.OlOrderStatusData[i].ol_i_id,

          Term.pClientData[iTermId].OrderStatusData.OlOrderStatusData[i].ol_quantity
,

          Term.pClientData[iTermId].OrderStatusData.OlOrderStatusData[i].ol_amount,

          Term.pClientData[iTermId].OrderStatusData.OlOrderStatusData[i].ol_delivery
_d.day,

          Term.pClientData[iTermId].OrderStatusData.OlOrderStatusData[i].ol_delivery
_d.month,

          Term.pClientData[iTermId].OrderStatusData.OlOrderStatusData[i].ol_delivery
_d.year);
                    }

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

          return szForm;
}
```

```
/* FUNCTION: char *MakeDeliveryForm(int iTermId, int iSyncId, BOOL bInput)
 *
 * PURPOSE:         This function
 *
 * ARGUMENTS:       int                                    iTermId   client browser
terminal id
 *                                    int
          iSyncId   client browser sync id
 *                                    BOOL                          bInput    TRUE
if form is being constructed for input else FALSE
 *
 * RETURNS:                   char *                       A pointer to buffer
inside client structure where HTML form is built.
 *
 * COMMENTS:        The internal client buffer is created when the terminal id is
assigned and should not
 *                                    be freed except when the client terminal id
is no longer needed.
 */

static char *MakeDeliveryForm(int iTermId, int iSyncId, BOOL bInput)
{
          char      *szForm;

          szForm = (char *)Term.pClientData[iTermId].szBuffer;

          Term.pClientData[iTermId].DeliveryData.w_id =
Term.pClientData[iTermId].w_id;

          if ( bInput )
          {
                    strcpy( szForm,     "<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD><BODY>\
<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"PI*\" VALUE=\"\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"5\">"); // DELIVERY_FORM = 5

                    wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">",

          iTermId,

          iSyncId);

                    strcat(szForm,      "<PRE>
Delivery<BR>" );

                    wsprintf(szForm+strlen(szForm), "Warehouse: %4.4d<BR><BR>",
Term.pClientData[iTermId].DeliveryData.w_id);

                    strcat( szForm, "Carrier Number: <INPUT NAME=\"OCD*\"
SIZE=1><BR><BR>\
Execution Status:<BR></PRE>\
<HR><INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\">\
<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">\
</BODY></FORM></HTML>" );

          }
          else // Not bInput
          {
```

```
                    strcpy( szForm,     "<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD><BODY>\
<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">\
<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"5\">"); // DELIVERY_FORM = 5

                    wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">\
<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">",

          iTermId,

          iSyncId);

          strcat(szForm,      "<PRE>                              Delivery<BR>"
);

          wsprintf(szForm+strlen(szForm), "Warehouse: %4.4d<BR><BR>",
Term.pClientData[iTermId].DeliveryData.w_id);


                    wsprintf(szForm+strlen(szForm), "Carrier Number: %2.2d<BR><BR>\
Execution Status: %25.25s<BR></PRE>",

          Term.pClientData[iTermId].DeliveryData.o_carrier_id,

          Term.pClientData[iTermId].DeliveryData.execution_status);

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

          }

          return szForm;
}

/* 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.
 *
 */
```

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

          return;
}

/* FUNCTION: void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK         *pECB, int
iTermId, int iSyncId)
 *
 * PURPOSE:         This function gets and validates the input data from the new
order form
 *                            filling in the required input variables. it then calls
the SQLNewOrder
 *                            transaction, constructs the output form and writes it
back to client
 *                            browser.
 *
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB     passed in structure
pointer from inetsrv.
 *                                      int
                    iTermId   client browser terminal id
 *                                      int
                    iSyncId client browser sync id
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

static void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK     *pECB, int iTermId, int
iSyncId)
{
          int                           iRc, iError;
          long ilen, *olen;
//        PECBINFO  pEcbInfo;
          char buf[40];
#ifdef LOCAL_ALLOC
          NEW_ORDER_DATA *NewOrderDataPtr;
#endif

     if((iRc = ThrTpInit(pECB)) <0)
     {
          // This is bad
                     {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessNewOrder Thread %d iTermId %d
Failed ThrTpInit \r\n",
                                        GetCurrentThreadId(), iTermId );
                              fclose(fp);
                     }
                     sprintf(buf, "NewOrder ThrTpInit < 0 >> Thread %d",
GetCurrentThreadId());
         ErrorMessage(pECB, ERR_TPINIT_BAD, ERR_TYPE_TUXEDO, buf, 0, 0);
     }

          memset(&Term.pClientData[iTermId].NewOrderData, 0,
sizeof(NEW_ORDER_DATA));
```

```
          Term.pClientData[iTermId].NewOrderData.w_id =
Term.pClientData[iTermId].w_id;

          if ( (iError=GetNewOrderData(pECB->lpszQueryString,
&Term.pClientData[iTermId].NewOrderData)) != ERR_SUCCESS )
          {
                    ErrorMessage(pECB, iError, ERR_TYPE_WEBDLL, NULL, iTermId,
iSyncId);
                    return;
          }

#ifdef LOCAL_ALLOC
          if ((NewOrderDataPtr = (NEW_ORDER_DATA *) tpalloc("CARRAY", NULL,
sizeof(NEW_ORDER_DATA))) == NULL)
          {
                    iRc = tperrno;
                    {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessNewOrder Thread %d iTermId %d
tpalloc tperrno %d \r\n",
                                        GetCurrentThreadId(), iTermId, iRc);
                              fclose(fp);
                    }
                    sprintf(buf, "NewOrder Tpalloc %d", iRc);
                    ErrorMessage(pECB, ERR_TPALLOC_BAD, ERR_TYPE_TUXEDO, buf, 0, 0);
                    tpfree((char *)NewOrderDataPtr);
                    return;
          }

          *NewOrderDataPtr = Term.pClientData[iTermId].NewOrderData;
           ilen = sizeof(NEW_ORDER_DATA);
           olen = &ilen;

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "* ProcessNewOrder Thread %d iTermId %d
NewOrderDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId, &NewOrderDataPtr);
                    fclose(fp);
           }

#ifdef QQQ
     if (tpcall(qname[iTermId%9], (char *)NewOrderDataPtr, ilen, (char
**)&NewOrderDataPtr, (long *)olen, TPSIGRSTRT) == -1)
#else
     if (tpcall("NEWORDER", (char *)NewOrderDataPtr, ilen, (char
**)&NewOrderDataPtr, (long *)olen, TPSIGRSTRT) == -1)
#endif
           {
         iRc = tperrno;
                     {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessNewOrder Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                        GetCurrentThreadId(), iTermId, iRc);
                              fclose(fp);
                     }
                     sprintf(buf, "Neworder tpcall %d", iRc);
         ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf, iTermId, iSyncId);
                     tpfree((char *)NewOrderDataPtr);
         return;
     }

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "** ProcessNewOrder Thread %d iTermId %d
NewOrderDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId, &NewOrderDataPtr);
                    fclose(fp);
           }

     Term.pClientData[iTermId].NewOrderData = *NewOrderDataPtr;

     iRc = NewOrderDataPtr->retval;

           tpfree((char *)NewOrderDataPtr);

#else

          *Term.pClientData[iTermId].NewOrderDataPtr =
Term.pClientData[iTermId].NewOrderData;

     ilen = sizeof(NEW_ORDER_DATA);
     olen = &ilen;

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "* ProcessNewOrder Thread %d iTermId %d
NewOrderDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].NewOrderDataPtr);
                    fclose(fp);
           }

#ifdef QQQ
     if (tpcall(qname[iTermId%9], (char *)Term.pClientData[iTermId].NewOrderDataPtr,
ilen,
                     (char **)&Term.pClientData[iTermId].NewOrderDataPtr, (long
*)olen, TPSIGRSTRT) == -1)
#else
     if (tpcall("NEWORDER", (char *)Term.pClientData[iTermId].NewOrderDataPtr, ilen,
                     (char **)&Term.pClientData[iTermId].NewOrderDataPtr, (long
*)olen, TPSIGRSTRT) == -1)
#endif
           {
         iRc = tperrno;
                     {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessNewOrder Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                        GetCurrentThreadId(), iTermId, iRc);
```

```
                              fclose(fp);
                     }
                     sprintf(buf, "Neworder tpcall %d", iRc);
         ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf, iTermId, iSyncId);
//                   tpfree((char *)NewOrderDataPtr);
         return;
     }

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "** ProcessNewOrder Thread %d iTermId %d
NewOrderDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].NewOrderDataPtr);
                    fclose(fp);
           }

     Term.pClientData[iTermId].NewOrderData =
*Term.pClientData[iTermId].NewOrderDataPtr;

     iRc = Term.pClientData[iTermId].NewOrderDataPtr->retval;

#endif
#ifdef BOIL
          Sleep( 1100 );
#endif
          if ( iRc < 0 )
                    ErrorMessage(pECB, ERR_NEW_ORDER_NOT_PROCESSED, ERR_TYPE_WEBDLL,
NULL, iTermId, iSyncId);
          else
                    WriteZString(pECB, MakeNewOrderForm(iTermId, iSyncId, FALSE,
(BOOL)iRc) );
          return;
}

/* FUNCTION: void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId)
 *
 * PURPOSE:         This function gets and validates the input data from the payment
form
 *                            filling in the required input variables. It then calls
the SQLPayment
 *                            transaction, constructs the output form and writes it
back to client
 *                            browser.
 *
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB     passed in structure
pointer from inetsrv.
 *                                      int
                    iTermId   client browser terminal id
 *                                      int
                    iSyncId client browser sync id
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */
```

```
static void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId)
{
          int                            iRc, iError;
          long                ilen, *olen;
//        PECBINFO  pEcbInfo;
          char buf[40];
#ifdef LOCAL_ALLOC
          PAYMENT_DATA *PaymentDataPtr;
#endif

     if((iRc = ThrTpInit(pECB)) <0)
     {
          // This is bad
                     {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessPayment Thread %d iTermId %d
Failed ThrTpInit \r\n",
                                        GetCurrentThreadId(), iTermId );
                              fclose(fp);
                     }
                     sprintf(buf, "Payment ThrTpInit < 0 >> Thread %d",
GetCurrentThreadId());
         ErrorMessage(pECB, ERR_TPINIT_BAD, ERR_TYPE_TUXEDO, buf, 0, 0);
     }

           memset(&Term.pClientData[iTermId].PaymentData, 0, sizeof(PAYMENT_DATA));

     Term.pClientData[iTermId].PaymentData.w_id = Term.pClientData[iTermId].w_id;

     if ( (iError=GetPaymentData(pECB->lpszQueryString,
&Term.pClientData[iTermId].PaymentData)) != ERR_SUCCESS )
     {
          ErrorMessage(pECB, iError, ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
          return;
     }

#ifdef LOCAL_ALLOC
          if ((PaymentDataPtr = (PAYMENT_DATA *) tpalloc("CARRAY", NULL,
sizeof(PAYMENT_DATA))) == NULL)
          {
                    iRc = tperrno;
                    {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessPayment Thread %d iTermId %d
tpalloc tperrno %d \r\n",
                              GetCurrentThreadId(), iTermId, iRc);
                              fclose(fp);
                    }
                    sprintf(buf, "Payment Tpalloc %d", iRc);
                    ErrorMessage(pECB, ERR_TPALLOC_BAD, ERR_TYPE_TUXEDO, buf, 0, 0);
                    tpfree((char *)PaymentDataPtr);
                    return;
          }

    *PaymentDataPtr = Term.pClientData[iTermId].PaymentData;

           ilen = sizeof(PAYMENT_DATA);
     olen = &ilen;
```

```
           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "* ProcessPayment Thread %d iTermId %d
PaymentDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId, &PaymentDataPtr);
                    fclose(fp);
           }

#ifdef QQQ
     if (tpcall(qname[iTermId%9], (char *)PaymentDataPtr, ilen, (char
**)&PaymentDataPtr, (long *)olen, TPSIGRSTRT) == -1)
#else
     if (tpcall("PAYMENT", (char *)PaymentDataPtr, ilen, (char **)&PaymentDataPtr,
(long *)olen, TPSIGRSTRT) == -1)
#endif
           {
         iRc = tperrno;
                     {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessPayment Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                        GetCurrentThreadId(), iTermId, iRc);
                              fclose(fp);
                     }
                     sprintf(buf, "Payment tpcall %d", iRc);
         ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf, iTermId, iSyncId);
         tpfree((char *)PaymentDataPtr);
         return;
     }

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "** ProcessPayment Thread %d iTermId %d
PaymentDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId, &PaymentDataPtr);
                    fclose(fp);
           }

     Term.pClientData[iTermId].PaymentData = *PaymentDataPtr;

     iRc = PaymentDataPtr->retval;

           tpfree((char *)PaymentDataPtr);

#else

     *Term.pClientData[iTermId].PaymentDataPtr =
Term.pClientData[iTermId].PaymentData;

           ilen = sizeof(PAYMENT_DATA);
     olen = &ilen;

           if ( dLog )
           {
                   FILE *fp;

                   fp = fopen(szTpccLogPath, "ab");
                   fprintf(fp, "* ProcessPayment Thread %d iTermId %d
PaymentDataPtr: %x \r\n",
                            GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].PaymentDataPtr);
                   fclose(fp);
          }

#ifdef QQQ
     if (tpcall(qname[iTermId%9], (char *)Term.pClientData[iTermId].PaymentDataPtr,
ilen,
                    (char **)&Term.pClientData[iTermId].PaymentDataPtr, (long
*)olen, TPSIGRSTRT) == -1)
#else
     if (tpcall("PAYMENT", (char *)Term.pClientData[iTermId].PaymentDataPtr, ilen,
                    (char **)&Term.pClientData[iTermId].PaymentDataPtr, (long
*)olen, TPSIGRSTRT) == -1)
#endif
          {
        iRc = tperrno;
                    {
                            FILE *fp;

                            fp = fopen(szTpccLogPath, "ab");
                            fprintf(fp, "* ProcessPayment Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                     GetCurrentThreadId(), iTermId, iRc);
                            fclose(fp);
                    }
                    sprintf(buf, "Payment tpcall %d", iRc);
        ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf, iTermId, iSyncId);
//        tpfree((char *)PaymentDataPtr);
        return;
     }

          if ( dLog )
          {
                   FILE *fp;

                   fp = fopen(szTpccLogPath, "ab");
                   fprintf(fp, "** ProcessPayment Thread %d iTermId %d
PaymentDataPtr: %x \r\n",
                            GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].PaymentDataPtr);
                   fclose(fp);
          }

     Term.pClientData[iTermId].PaymentData =
*Term.pClientData[iTermId].PaymentDataPtr;

     iRc = Term.pClientData[iTermId].PaymentDataPtr->retval;

#endif
#ifdef BOIL
         Sleep( 2200 );
#endif

     if ( iRc == 0 )
         ErrorMessage(pECB, ERR_PAYMENT_INVALID_CUSTOMER, ERR_TYPE_WEBDLL, NULL,
iTermId, iSyncId);
     else if ( iRc < 0 ) {
```

```
          ErrorMessage(pECB, ERR_PAYMENT_NOT_PROCESSED, ERR_TYPE_WEBDLL, NULL,
iTermId, iSyncId);
     }
     else
          WriteZString(pECB, MakePaymentForm(iTermId, iSyncId, FALSE) );

     return;

}


/* FUNCTION: void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId,
int iSyncId)
 *
 * PURPOSE:          This function gets and validates the input data from the Order
Status
 *                            form filling in the required input variables. It then
calls the
 *                            SQLOrderStatus transaction, constructs the output form
and writes it
 *                            back to client browser.
 *
 * ARGUMENTS:        EXTENSION_CONTROL_BLOCK       *pECB     passed in structure
pointer from inetsrv.
 *                                      int
                   iTermId   client browser terminal id
 *                                      int
                   iSyncId client browser sync id
 *
 * RETURNS:                   None
 *
 * COMMENTS:         None
 *
 */

static void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId)
{
          int                            iRc, iError;
          long ilen, *olen;
//        PECBINFO  pEcbInfo;
          char buf[40];
#ifdef LOCAL_ALLOC
          ORDER_STATUS_DATA *OrderStatusDataPtr;
#endif

     if((iRc = ThrTpInit(pECB)) <0)
     {
          // This is bad
                     {
                            FILE *fp;

                            fp = fopen(szTpccLogPath, "ab");
                            fprintf(fp, "* ProcessOrderStatus Thread %d iTermId %d
Failed ThrTpInit \r\n",
                                     GetCurrentThreadId(), iTermId );
                            fclose(fp);
                     }
                     sprintf(buf, "OrderStatus ThrTpInit < 0 >> Thread %d",
GetCurrentThreadId());
         ErrorMessage(pECB, ERR_TPINIT_BAD, ERR_TYPE_TUXEDO, buf, 0, 0);
     }
     memset(&Term.pClientData[iTermId].OrderStatusData, 0,
sizeof(ORDER_STATUS_DATA));

     Term.pClientData[iTermId].OrderStatusData.w_id =
Term.pClientData[iTermId].w_id;

     if ( (iError=GetOrderStatusData(pECB->lpszQueryString,
&Term.pClientData[iTermId].OrderStatusData)) != ERR_SUCCESS )
     {
          ErrorMessage(pECB, iError, ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
          return;
     }

#ifdef LOCAL_ALLOC

          if ((OrderStatusDataPtr = (ORDER_STATUS_DATA *) tpalloc("CARRAY", NULL,
sizeof(ORDER_STATUS_DATA))) == NULL)
          {
                    iRc = tperrno;
                    {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessOrderStatus Thread %d iTermId %d
tpalloc tperrno %d \r\n",
                              GetCurrentThreadId(), iTermId, iRc);
                              fclose(fp);
                    }
                    sprintf(buf, "OrderStatus Tpcalloc %d", iRc);
                    ErrorMessage(NULL, ERR_TPALLOC_BAD, ERR_TYPE_TUXEDO, buf, 0, 0);
                    tpfree((char *)OrderStatusDataPtr);
                    return;
          }

     *OrderStatusDataPtr = Term.pClientData[iTermId].OrderStatusData;

           ilen = sizeof(ORDER_STATUS_DATA);
     olen = &ilen;

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "* ProcessOrderStatus Thread %d iTermId %d
OrderStatusDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId, &OrderStatusDataPtr);
                    fclose(fp);
           }

#ifdef QQQ
     if (tpcall(qname[iTermId%9], (char *)OrderStatusDataPtr, ilen, (char
**)&OrderStatusDataPtr, (long *)olen, TPSIGRSTRT) == -1)
#else
     if (tpcall("ORDERSTATUS", (char *)OrderStatusDataPtr, ilen, (char
**)&OrderStatusDataPtr, (long *)olen, TPSIGRSTRT) == -1)
#endif
           {
         iRc = tperrno;
                     {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
```

```
                              fprintf(fp, "* ProcessOrderStatus Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                        GetCurrentThreadId(), iTermId, iRc);
                              fclose(fp);
                     }
                     sprintf(buf, "OrderStatus tpcall %d", iRc);
         ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf, iTermId, iSyncId);
         tpfree((char *)OrderStatusDataPtr);
         return;
     }

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "** ProcessOrderStatus Thread %d iTermId %d
OrderStatusDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId, &OrderStatusDataPtr);
                    fclose(fp);
           }

     Term.pClientData[iTermId].OrderStatusData = *OrderStatusDataPtr;

     iRc = OrderStatusDataPtr->retval;

           tpfree((char *)OrderStatusDataPtr);

#else

     *Term.pClientData[iTermId].OrderStatusDataPtr =
Term.pClientData[iTermId].OrderStatusData;

           ilen = sizeof(ORDER_STATUS_DATA);
     olen = &ilen;

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "* ProcessOrderStatus Thread %d iTermId %d
OrderStatusDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].OrderStatusDataPtr);
                    fclose(fp);
           }

#ifdef QQQ
     if (tpcall(qname[iTermId%9], (char
*)Term.pClientData[iTermId].OrderStatusDataPtr, ilen,
                     (char **)&Term.pClientData[iTermId].OrderStatusDataPtr, (long
*)olen, TPSIGRSTRT) == -1)
#else
     if (tpcall("ORDERSTATUS", (char *)Term.pClientData[iTermId].OrderStatusDataPtr,
ilen,
                     (char **)&Term.pClientData[iTermId].OrderStatusDataPtr, (long
*)olen, TPSIGRSTRT) == -1)
#endif
           {
         iRc = tperrno;
                     {
                              FILE *fp;
                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessOrderStatus Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                        GetCurrentThreadId(), iTermId, iRc);
                              fclose(fp);
                     }
                     sprintf(buf, "OrderStatus tpcall %d", iRc);
         ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf, iTermId, iSyncId);
//         tpfree((char *)Term.pClientData[iTermId].OrderStatusDataPtr);
         return;
     }

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "** ProcessOrderStatus Thread %d iTermId %d
OrderStatusDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].OrderStatusDataPtr);
                    fclose(fp);
           }

     Term.pClientData[iTermId].OrderStatusData =
*Term.pClientData[iTermId].OrderStatusDataPtr;

     iRc = Term.pClientData[iTermId].OrderStatusDataPtr->retval;

#endif
#ifdef BOIL
          Sleep( 3300 );
#endif

           if ( iRc == 0 )
          ErrorMessage(pECB, ERR_NOSUCH_CUSTOMER, ERR_TYPE_WEBDLL, NULL, iTermId,
iSyncId);
     else if ( iRc < 0 ) {
          ErrorMessage(pECB, ERR_ORDER_STATUS_NOT_PROCESSED, ERR_TYPE_WEBDLL, NULL,
iTermId, iSyncId);
     }
     else
          WriteZString(pECB, MakeOrderStatusForm(iTermId, iSyncId, FALSE) );

     return;

}

/* FUNCTION: void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId,
int iSyncId)
 *
 * PURPOSE:          This function gets and validates the input data from the
delivery form
 *                            filling in the required input variables. It then calls
the PostDeliveryInfo
 *                            Api, The client is then informed that the transaction
has been posted.
 *
 * ARGUMENTS:        EXTENSION_CONTROL_BLOCK       *pECB     passed in structure
pointer from inetsrv.
 *                                      int
                    iTermId   client browser terminal id
 *                                      int
                    iSyncId   clinet browser sync id
 *
 * RETURNS:                   None
 *
 * COMMENTS:         None
 *
 */

static void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId)
{
          int                 iRc;
//        int                 iError;
          char      szTmp[26];
          long ilen;
          int                 tmp;
          char buf[40];
#ifdef LOCAL_ALLOC
          DELIVERY_DATA *DeliveryDataPtr;
#endif

     if((iRc = ThrTpInit(pECB)) <0)
     {
          // This is bad
                     {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessDelivery Thread %d iTermId %d
Failed ThrTpInit \r\n",
                                        GetCurrentThreadId(), iTermId );
                              fclose(fp);
                     }
                     sprintf(buf, "Delivery ThrTpInit < 0 >> Thread %d",
GetCurrentThreadId());
         ErrorMessage(pECB, ERR_TPINIT_BAD, ERR_TYPE_TUXEDO, buf, 0, 0);
     }

           memset(&Term.pClientData[iTermId].DeliveryData, 0,
sizeof(DELIVERY_DATA));

          if ( !GetKeyValue(pECB->lpszQueryString, "OCD*", szTmp, sizeof(szTmp)) )
          {
                    ErrorMessage(pECB, ERR_DELIVERY_MISSING_OCD_KEY,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
                    return;
          }

          if ( !IsNumeric(szTmp) )
          {
                    ErrorMessage(pECB, ERR_DELIVERY_CARRIER_INVALID,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
                    return;
          }

          Term.pClientData[iTermId].DeliveryData.o_carrier_id         = atoi(szTmp);
          tmp = Term.pClientData[iTermId].DeliveryData.o_carrier_id * 70;

          if ( Term.pClientData[iTermId].DeliveryData.o_carrier_id > 10 ||
Term.pClientData[iTermId].DeliveryData.o_carrier_id < 1 )
          {
```

```
                    ErrorMessage(pECB, ERR_DELIVERY_CARRIER_ID_RANGE,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
                    return;
          }

     Term.pClientData[iTermId].DeliveryData.w_id = Term.pClientData[iTermId].w_id;

           GetLocalTime(&Term.pClientData[iTermId].DeliveryData.queue_time);

//     if ( (iError=GetDeliveryData(pECB->lpszQueryString,
&Term.pClientData[iTermId].DeliveryData)) != ERR_SUCCESS )
//     {
//          ErrorMessage(pECB, iError, ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
//          return;
//     }


#ifdef LOCAL_ALLOC

          if ((DeliveryDataPtr = (DELIVERY_DATA *) tpalloc("CARRAY", NULL,
sizeof(DELIVERY_DATA))) == NULL)
          {
                    iRc = tperrno;
                    {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessDelivery Thread %d iTermId %d
tpalloc tperrno %d \r\n",
                              GetCurrentThreadId(), iTermId, iRc);
                              fclose(fp);
                    }
                    sprintf(buf, "Delivery Tpcalloc %d", iRc);
                    ErrorMessage(NULL, ERR_TPALLOC_BAD, ERR_TYPE_TUXEDO, buf, 0, 0);
                    tpfree((char *)DeliveryDataPtr);
                    return;
          }

          *DeliveryDataPtr = Term.pClientData[iTermId].DeliveryData;

     ilen = sizeof(DELIVERY_DATA);

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "* ProcessDelivery Thread %d iTermId %d
DeliveryDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId, &DeliveryDataPtr);
                    fclose(fp);
           }

     if (tpacall("DELIVERY", (char *)DeliveryDataPtr, ilen, TPNOREPLY|TPSIGRSTRT) !=
0) {
         iRc = tperrno;
                     {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessDelivery Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                        GetCurrentThreadId(), iTermId, iRc);
                              fclose(fp);
                     }
                     sprintf(buf, "Delivery tpcall %d", iRc);
         ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf, iTermId, iSyncId);
         tpfree((char *)DeliveryDataPtr);
         return;
     }

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "** ProcessDelivery Thread %d iTermId %d
DeliveryDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId, &DeliveryDataPtr);
                    fclose(fp);
           }

          tpfree((char *)DeliveryDataPtr);

#else

     *Term.pClientData[iTermId].DeliveryDataPtr =
Term.pClientData[iTermId].DeliveryData;

     ilen = sizeof(DELIVERY_DATA);
     olen = &ilen;

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "* ProcessDelivery Thread %d iTermId %d
DeliveryDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].DeliveryDataPtr);
                    fclose(fp);
           }

     if (tpcall("DELIVERY", (char *)Term.pClientData[iTermId].DeliveryDataPtr, ilen,
                     (char **)&Term.pClientData[iTermId].DeliveryDataPtr, (long
*)olen, TPSIGRSTRT) == -1) {
         iRc = tperrno;
                     {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessDelivery Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                        GetCurrentThreadId(), iTermId, iRc);
                              fclose(fp);
                     }
                     sprintf(buf, "Delivery tpcall %d", iRc);
         ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf, iTermId, iSyncId);
//         tpfree((char *)Term.pClientData[iTermId].DeliveryDataPtr);
         return;
     }

           if ( dLog )
           {
                    FILE *fp;
                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "** ProcessDelivery Thread %d iTermId %d
DeliveryDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].DeliveryDataPtr);
                    fclose(fp);
           }

    iRc = Term.pClientData[iTermId].DeliveryDataPtr->retval;

#endif

#ifdef TSLICE
          Sleep(tmp); // Relinquish thread timeslice
#endif
          strcpy(Term.pClientData[iTermId].DeliveryData.execution_status, "Delivery
has been queued.");

#ifdef BOIL
          Sleep( 4400 );
#endif

          WriteZString(pECB, MakeDeliveryForm(iTermId, iSyncId, FALSE) );

     return;

}

/* FUNCTION: void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId,
int iSyncId)
 *
 * PURPOSE:         This function gets and validates the input data from the Stock
Level
 *                            form filling in the required input variables. It then
calls the
 *                            SQLStockLevel transaction, constructs the output form
and writes it
 *                            back to client browser.
 *
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB     passed in structure
pointer from inetsrv.
 *                                      int
                    iTermId   client browser terminal id
 *                                      int
                    iSyncId   client browser sync id
 *
 * RETURNS:                   None
 *
 * COMMENTS:        None
 *
 */

static void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId)
{
          int                 iRc;
//        int                 iError;
          char                szTmp[26];
//        BOOL                bRc;
          long ilen, *olen;
//        PECBINFO  pEcbInfo;
          char buf[40];
```

```
#ifdef LOCAL_ALLOC
          STOCK_LEVEL_DATA *StockLevelDataPtr;
#endif

     if((iRc = ThrTpInit(pECB)) <0)
     {
          // This is bad
                     {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessStockLevel Thread %d iTermId %d
Failed ThrTpInit \r\n",
                                        GetCurrentThreadId(), iTermId );
                              fclose(fp);
                     }
                     sprintf(buf, "StockLevel ThrTpInit < 0 >> Thread %d",
GetCurrentThreadId());
         ErrorMessage(pECB, ERR_TPINIT_BAD, ERR_TYPE_TUXEDO, buf, 0, 0);
     }

     memset(&Term.pClientData[iTermId].StockLevelData, 0, sizeof(STOCK_LEVEL_DATA));

     Term.pClientData[iTermId].StockLevelData.w_id = Term.pClientData[iTermId].w_id;
     Term.pClientData[iTermId].StockLevelData.d_id = Term.pClientData[iTermId].d_id;


     if ( !GetKeyValue(pECB->lpszQueryString, "TT*", szTmp, sizeof(szTmp)) )
     {
          ErrorMessage(pECB, ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY, ERR_TYPE_WEBDLL,
NULL, iTermId, iSyncId);
          return;
     }

     if ( !IsNumeric(szTmp) )
     {
          ErrorMessage(pECB, ERR_STOCKLEVEL_THRESHOLD_INVALID, ERR_TYPE_WEBDLL,
NULL, iTermId, iSyncId);
          return;
     }

     Term.pClientData[iTermId].StockLevelData.thresh_hold = atoi(szTmp);

     if ( Term.pClientData[iTermId].StockLevelData.thresh_hold >= 100 ||
Term.pClientData[iTermId].StockLevelData.thresh_hold < 0 )
     {
          ErrorMessage(pECB, ERR_STOCKLEVEL_THRESHOLD_RANGE, ERR_TYPE_WEBDLL, NULL,
iTermId, iSyncId);
          return;
     }

#ifdef LOCAL_ALLOC
          if ((StockLevelDataPtr = (STOCK_LEVEL_DATA *) tpalloc("CARRAY", NULL,
sizeof(STOCK_LEVEL_DATA))) == NULL)
          {
                    iRc = tperrno;
                    {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessStockLevel Thread %d iTermId %d
tpalloc tperrno %d \r\n",
                              GetCurrentThreadId(), iTermId, iRc);
                              fclose(fp);
                    }
                    sprintf(buf, "Tpcalloc %d", iRc);
                    ErrorMessage(NULL, ERR_TPALLOC_BAD, ERR_TYPE_TUXEDO, buf, 0, 0);
                    tpfree((char *)StockLevelDataPtr);
                    return;
          }

     *StockLevelDataPtr = Term.pClientData[iTermId].StockLevelData;

     ilen = sizeof(STOCK_LEVEL_DATA);
     olen = &ilen;

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "* ProcessStockLevel Thread %d iTermId %d
StockLevelDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId, &StockLevelDataPtr);
                    fclose(fp);
           }

#ifdef QQQ
     if (tpcall(qname[iTermId%9], (char *) StockLevelDataPtr, ilen, (char **)
&StockLevelDataPtr, (long *) olen, TPSIGRSTRT) == -1) {
#else
     if (tpcall("STOCKLEVEL", (char *) StockLevelDataPtr, ilen, (char **)
&StockLevelDataPtr, (long *) olen, TPSIGRSTRT) == -1) {
#endif
         iRc = tperrno;
                     {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessStockLevel Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                        GetCurrentThreadId(), iTermId, iRc);
                              fclose(fp);
                     }
                     sprintf(buf, "StockLevel tpcall %d", iRc);
         ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf, iTermId, iSyncId);
                     tpfree((char *)StockLevelDataPtr);
         return;
     }

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "** ProcessStockLevel Thread %d iTermId %d
StockLevelDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId, &StockLevelDataPtr);
                    fclose(fp);
           }

     Term.pClientData[iTermId].StockLevelData = *StockLevelDataPtr;

     iRc = StockLevelDataPtr->retval;

           tpfree((char *)StockLevelDataPtr);
```

```
#else

     *Term.pClientData[iTermId].StockLevelDataPtr =
Term.pClientData[iTermId].StockLevelData;

     ilen = sizeof(STOCK_LEVEL_DATA);
     olen = &ilen;

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "* ProcessStockLevel Thread %d iTermId %d
StockLevelDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].StockLevelDataPtr);
                    fclose(fp);
           }

#ifdef QQQ
     if (tpcall(qname[iTermId%9], (char
*)Term.pClientData[iTermId].StockLevelDataPtr, ilen,
                     (char **)&Term.pClientData[iTermId].StockLevelDataPtr, (long *)
olen, TPSIGRSTRT) == -1) {
#else
     if (tpcall("STOCKLEVEL", (char *)Term.pClientData[iTermId].StockLevelDataPtr,
ilen,
                     (char **)&Term.pClientData[iTermId].StockLevelDataPtr, (long *)
olen, TPSIGRSTRT) == -1) {
#endif
         iRc = tperrno;
                     {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ProcessStockLevel Thread %d iTermId %d
tpcall tperrno %d \r\n",
                                        GetCurrentThreadId(), iTermId, iRc);
                              fclose(fp);
                     }
                     sprintf(buf, "StockLevel tpcall %d", iRc);
         ErrorMessage(pECB, ERR_TPCALL_BAD, ERR_TYPE_TUXEDO, buf, iTermId, iSyncId);
//                   tpfree((char *)StockLevelDataPtr);
         return;
     }

           if ( dLog )
           {
                    FILE *fp;

                    fp = fopen(szTpccLogPath, "ab");
                    fprintf(fp, "** ProcessStockLevel Thread %d iTermId %d
StockLevelDataPtr: %x \r\n",
                              GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].StockLevelDataPtr);
                    fclose(fp);
           }

     Term.pClientData[iTermId].StockLevelData =
*Term.pClientData[iTermId].StockLevelDataPtr;
```

```
     iRc = Term.pClientData[iTermId].StockLevelDataPtr->retval;

#endif

#ifdef BOIL
          Sleep( 5500 );
#endif

           if ( iRc == 0 ) {
          ErrorMessage(pECB, ERR_STOCKLEVEL_NOT_PROCESSED, ERR_TYPE_WEBDLL, NULL,
iTermId, iSyncId);
     }
     else
          WriteZString(pECB, MakeStockLevelForm(iTermId, iSyncId, FALSE) );

     return;

}

/* FUNCTION: int GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA
*pNewOrderData)
 *
 * PURPOSE:         This function extracts and validates the new order form data
from an http command string.
 *
 * ARGUMENTS:       LPSTR                         lpszQueryString
          client browser http command string
 *                                      NEW_ORDER_DATA      *pNewOrderData
          pointer to new order data structure
 *
 * RETURNS:                   int
                                        error code indicating reason for failure
 *                                      ERR_SUCCESS
                                        new order input data successfully parsed
 *
 *
 * COMMENTS:        None
 *
 */

static int GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA *pNewOrderData)
{
          char      szTmp[26];
          int                 i;
          short               spw;
          short     items;
          BOOL      bCheck;
          char *  s;

          s=lpszQueryString;

          GetVal(s, "DID*", szTmp, sizeof(szTmp))
          if(! *s)
                    return ERR_NEWORDER_FORM_MISSING_DID;

          if ( !IsNumeric(szTmp) )
                    return ERR_NEWORDER_DISTRICT_INVALID;

          pNewOrderData->d_id = atoi(szTmp);

          GetVal(s, "CID*", szTmp, sizeof(szTmp))
          if(! *s)
                    return ERR_NEWORDER_CUSTOMER_KEY;
```

```
          if ( !IsNumeric(szTmp) )
                    return ERR_NEWORDER_CUSTOMER_INVALID;
          pNewOrderData->c_id = atoi(szTmp);
          pNewOrderData->o_all_local=1;

          bCheck = FALSE;
          for(i=0, items=0; i<15; i++)
          {
                    GetVal(s, szIID[i], szTmp, sizeof(szTmp))
                    if(! *s)
                              return ERR_NEWORDER_MISSING_IID_KEY;

                    s=s-24;
                    if ( szTmp[0] )
                    {
                              //if blank lines between item ids
                              if ( bCheck )
                                        return ERR_NEWORDER_ITEM_BLANK_LINES;
                              if ( !IsNumeric(szTmp) )
                                        return ERR_NEWORDER_ITEMID_INVALID;
                              pNewOrderData->Ol[i].ol_i_id = atoi(szTmp);

                              GetVal(s, szSP[i], szTmp, sizeof(szTmp))
                              if(! *s)
                                        return ERR_NEWORDER_MISSING_SUPPW_KEY;
                              if ( !IsNumeric(szTmp) )
                                        return ERR_NEWORDER_SUPPW_INVALID;

                              if ( (spw = (short)atoi(szTmp)) <=0 || spw >
iMaxWareHouses )
                                        return ERR_NEWORDER_SUPPW_INVALID;

                              if ((pNewOrderData->Ol[i].ol_supply_w_id = spw ) !=
pNewOrderData->w_id)
                                        pNewOrderData->o_all_local=0;


                              GetVal(s, szQty[i], szTmp, sizeof(szTmp))
                              if(! *s)
                                        return ERR_NEWORDER_MISSING_QTY_KEY;

                              if ( !IsNumeric(szTmp) )
                                        return ERR_NEWORDER_QTY_INVALID;

                              pNewOrderData->Ol[i].ol_quantity = atoi(szTmp);
                              items++;

                              if ( pNewOrderData->Ol[i].ol_i_id >= 1000000 ||
pNewOrderData->Ol[i].ol_i_id < 1 )
                                        return ERR_NEWORDER_ITEMID_RANGE;
                              if ( pNewOrderData->Ol[i].ol_quantity >= 100 ||
pNewOrderData->Ol[i].ol_quantity < 1 )
                                        return ERR_NEWORDER_QTY_RANGE;
                    }
                    else
                    {
                              GetVal(s, szSP[i], szTmp, sizeof(szTmp))
                              if(! *s)
                                        return ERR_NEWORDER_MISSING_QTY_KEY;

                              if ( szTmp[0] )
                                        return ERR_NEWORDER_SUPPW_WITHOUT_ITEMID;
                              GetVal(s, szQty[i], szTmp, sizeof(szTmp))
                              if(! *s)
                                        return ERR_NEWORDER_MISSING_QTY_KEY;

                              if ( szTmp[0] )
                                        return ERR_NEWORDER_QTY_WITHOUT_ITEMID;

                              bCheck = TRUE;
                    }
          }
          if ( items == 0 )
                    return ERR_NEWORDER_NOITEMS_ENTERED;

          pNewOrderData->o_ol_cnt = items;

          return ERR_SUCCESS;
}

/* FUNCTION: int GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA *pPaymentData)
 *
 * PURPOSE:         This function extracts and validates the payment form data from
an http command string.
 *
 * ARGUMENTS:       LPSTR                         lpszQueryString
          client browser http command string
 *                                      PAYMENT_DATA        *pPaymentData
          pointer to payment data structure
 *
 * RETURNS:                             int
                                        error code indicating reason for failure
 *                                                ERR_SUCCESS
                                        all input data successfully parsed
 *
 * COMMENTS:        None
 *
 */

static int GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA *pPaymentData)
{
          char      szTmp[26];
          char      *ptr;
          char      *s;

          s=lpszQueryString;

          GetVal(s, "DID*", szTmp, sizeof(szTmp))
          if(! *s)
                    return ERR_PAYMENT_MISSING_DID_KEY;

          if ( !IsNumeric(szTmp) )
                    return ERR_PAYMENT_DISTRICT_INVALID;
          pPaymentData->d_id = atoi(szTmp);

          GetVal(s, "CID*", szTmp, sizeof(szTmp))
          if(! *s)
                    return ERR_PAYMENT_MISSING_CID_KEY;

          if ( szTmp[0] && !IsNumeric(szTmp) )
                    return ERR_PAYMENT_CUSTOMER_INVALID;


          pPaymentData->c_id = atoi(szTmp);
```

```
          if ( szTmp[0] == 0 )
          {
                    GetVal(s, "CLT*", szTmp, sizeof(szTmp))
                    if(! *s)
                              return ERR_PAYMENT_MISSING_CLT;
                    _strupr( szTmp );

                    strcpy(pPaymentData->c_last, szTmp);
                    if ( strlen(pPaymentData->c_last) > 16 )
                              return ERR_PAYMENT_LAST_NAME_TO_LONG;
          }
          else
          {
                    GetVal(s, "CLT*", szTmp, sizeof(szTmp))
                    if(! *s)
                              return ERR_PAYMENT_MISSING_CLT_KEY;
                    if ( szTmp[0] )
                              return ERR_PAYMENT_CID_AND_CLT;
          }

          s=lpszQueryString;

          GetVal(s, "CWI*", szTmp, sizeof(szTmp))
          if(! *s)
                    return ERR_PAYMENT_MISSING_CWI_KEY;

          if ( !IsNumeric(szTmp) )
                    return ERR_PAYMENT_CWI_INVALID;

          pPaymentData->c_w_id = atoi(szTmp);

          GetVal(s, "CDI*", szTmp, sizeof(szTmp))
          if(! *s)
                    return ERR_PAYMENT_MISSING_CDI_KEY;
          if ( !IsNumeric(szTmp) )
                    return ERR_PAYMENT_CDI_INVALID;
          pPaymentData->c_d_id = atoi(szTmp);


          GetVal(s, "HAM*", szTmp, sizeof(szTmp))
          if(! *s)
                    return ERR_PAYMENT_MISSING_HAM_KEY;

          ptr = szTmp;

          while( *ptr )
          {
                    if ( *ptr == '.' )
                    {
                              ptr++;
                              if ( !*ptr )
                                        break;
                              if ( *ptr < '0' || *ptr > '9' )
                                        return ERR_PAYMENT_HAM_INVALID;
                              ptr++;
                              if ( !*ptr )
                                        break;
                              if ( *ptr < '0' || *ptr > '9' )
                                        return ERR_PAYMENT_HAM_INVALID;
                              if ( !*ptr )
                                        return ERR_PAYMENT_HAM_INVALID;
                    }
                    else if ( *ptr < ’0’ || *ptr > ’9’ )
                                        return ERR_PAYMENT_HAM_INVALID;
                    ptr++;
          }

          pPaymentData->h_amount = atof(szTmp);
          if ( pPaymentData->h_amount >= 10000.00 || pPaymentData->h_amount < 0 )
                    return ERR_PAYMENT_HAM_RANGE;

          return ERR_SUCCESS;
}

/* FUNCTION: int GetOrderStatusData(LPSTR lpszQueryString, ORDER_STATUS_DATA
*pOrderStatusData)
 *
 * PURPOSE:         This function extracts and validates the payment form data from
an http command string.
 *
 * ARGUMENTS:       LPSTR                                   lpszQueryString
          client browser http command string
 *                                      ORDER_STATUS_DATA   *pOrderStatusData
          pointer to order status data structure
 *
 * RETURNS:                   int
                                        error code indicating reason for failure
 *                                      ERR_SUCCESS
                                        successfully parsed all required input data
 *
 * COMMENTS:        None
 *
 */
static int GetOrderStatusData(LPSTR lpszQueryString, ORDER_STATUS_DATA
*pOrderStatusData)
{
          char      szTmp[26];
          char      *s;

          s=lpszQueryString;

          GetVal(s, "DID*", szTmp, sizeof(szTmp))
          if(! *s)
                    return ERR_ORDERSTATUS_MISSING_DID_KEY;
          if ( !IsNumeric(szTmp) )
                    return ERR_ORDERSTATUS_DID_INVALID;
          pOrderStatusData->d_id = atoi(szTmp);

          GetVal(s, "CID*", szTmp, sizeof(szTmp))
          if(! *s)
                    return ERR_ORDERSTATUS_MISSING_CID_KEY;

          if ( szTmp[0] == 0 )
          {
                    pOrderStatusData->c_id = 0;
                    GetVal(s, "CLT*", szTmp, sizeof(szTmp))
                    if(! *s)
                              return ERR_ORDERSTATUS_MISSING_CLT_KEY;
                    _strupr( szTmp );
                    strcpy(pOrderStatusData->c_last, szTmp);
                    if ( strlen(pOrderStatusData->c_last) > 16 )
                              return ERR_ORDERSTATUS_CLT_RANGE;
          }
          else
          {
```

```
                    if ( !IsNumeric(szTmp) )
                              return ERR_ORDERSTATUS_CID_INVALID;
                    pOrderStatusData->c_id = atoi(szTmp);
                    GetVal(s, "CLT*", szTmp, sizeof(szTmp))
                    if(! *s)
                              return ERR_ORDERSTATUS_MISSING_CLT_KEY;
                    if ( szTmp[0] )
                              return ERR_ORDERSTATUS_CID_AND_CLT;
          }

          return ERR_SUCCESS;
}

/* FUNCTION: BOOL ReadRegistrySettings(void)
 *
 * PURPOSE:         This function reads the NT registry for startup parameters.
There parameters are
 *                            under the TPCC key.
 *
 * ARGUMENTS:   None
 *
 * RETURNS:                   None
 *
 * COMMENTS:        This function also sets up required operation variables to their
default value
 *                                      so if registry is not setup the default
values will be used.
 *
 */

static BOOL ReadRegistrySettings(void)
{
          HKEY      hKey;
          DWORD     size;
          DWORD     type;
          char      szTmp[256];

          bLog                          = FALSE;
          dLog                          = FALSE;
          iMaxWareHouses      = 0;
          iThreads            = 5;
          iDelayMs            = 100;
          iDeadlockRetry      = (short)3;
          strcpy(szTpccLogPath, "tpcclog.");


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

          if ( RegQueryValueEx(hKey, "PATH", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
          {
                    strcpy(szTpccLogPath, szTmp);
                    strcat(szTpccLogPath, "tpcclog.");
                    strcpy(szErrorLogPath, szTmp);
                    strcat(szErrorLogPath, "tpccerr.");
          }

          size = sizeof(szTmp);
          if ( RegQueryValueEx(hKey, "LOG", 0, &type, szTmp, &size) == ERROR_SUCCESS
)
```

```
          {
                    if ( !stricmp(szTmp, "ON") )
                              bLog = TRUE;
          }

          size = sizeof(szTmp);
          if ( RegQueryValueEx(hKey, "DEBUG", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
          {
                    if ( !stricmp(szTmp, "ON") )
                              dLog = TRUE;
          }

          size = sizeof(szTmp);
          if ( RegQueryValueEx(hKey, "MaximumWarehouses", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
          {
                    iMaxWareHouses = atoi(szTmp);
                    if ( iMaxWareHouses == 0 )
                              iMaxWareHouses = 0;
          }

          size = sizeof(szTmp);
          if ( RegQueryValueEx(hKey, "NumberOfDeliveryThreads", 0, &type, szTmp,
&size) == ERROR_SUCCESS )
                    iThreads = atoi(szTmp);
          if ( !iThreads )
                    iThreads = 5;

          size = sizeof(szTmp);
          if ( RegQueryValueEx(hKey, "BackoffDelay", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
                    iDelayMs = atoi(szTmp);
                    if ( !iDelayMs )
                              iDelayMs = 100;

          size = sizeof(szTmp);
          if ( RegQueryValueEx(hKey, "DeadlockRetry", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
                    iDeadlockRetry = (short)atoi(szTmp);
                    if ( !iDeadlockRetry )
                              iDeadlockRetry = (short)3;

          size = sizeof(szTmp);
          if ( RegQueryValueEx(hKey, "MaxConnections", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
                    iMaxConnections = (short)atoi(szTmp);
                    if ( !iMaxConnections )
                              iMaxConnections = (short)25;

          RegCloseKey(hKey);

          return FALSE;

}



/* FUNCTION: BOOL IsNumeric(char *ptr)
 *
 * PURPOSE:         This function determines if a string is numeric. It fails if any
characters other
 *                            than numeric and null terminator are present.
```

```
 *
 * ARGUMENTS:       char                          *ptr      pointer to string to
check.
 *
 * RETURNS:                   BOOL      FALSE     if string is not all numeric
 *                                                          TRUE      if string
contains only numeric characters i.e. ’0’ - ’9’
 *
 * COMMENTS:        None
 *
 */

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

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

#ifdef USE_ODBC
          /* FUNCTION: static int ODBCError(DBPROCESS *dbproc)
           *
           * PURPOSE:         This function Handles the processing of errors from
ODBC APIs
           *
           * ARGUMENTS:       PDBPROCESS
           *
           * RETURNS:                   BOOL      FALSE     if string is not all
numeric
           *                                                          TRUE      if
string contains only numeric characters i.e. ’0’ - ’9’
           *
           * COMMENTS:        None
           *
           */

          static int ODBCError(DBPROCESS *dbproc)
          {
                    RETCODE     rc;
                    SDWORD      lNativeError;
                    BOOL                bError;
                    char        szState[6];
                    char        szMsg[SQL_MAX_MESSAGE_LENGTH];
                    char                timebuf[128];
                    char                datebuf[128];

                    pdbproc->deadlock_detected = TRUE;
                    bError = FALSE;

                    while( SQLError(dbproc->henv,
                                                        dbproc->hdbc,
                                                        dbproc->hstmt,
                                                        szState,
                                                        &lNativeError,
                                                        szMsg,
                                                        sizeof(szMsg),
                                                        NULL) !=
SQL_NO_DATA_FOUND )
                    {
                              if (lNativeError == 1205)
```

```
                                        dbproc->deadlock_detected = TRUE;
                              else
                              {
                                        _strtime(timebuf);
                                        _strdate(datebuf);

                                        h_printf(dbproc->pECB, "%s %s : ODBC Error:
State=%s, Error=%ld, %s\n", datebuf, timebuf, szState, lNativeError, szMsg);
                                        bError = TRUE;
                              }
                    }
                    if ( bError )
                              return -1;
                    return dbproc->deadlock_detected;
          }
#endif

/* FUNCTION: void FormatHTMLString(char *szBuff, int iLen, char *szStr)
 *
 * PURPOSE:         This function Handles translation of HTML specific character
field data
 *                            when an HTML output form is generated.
 *
 * ARGUMENTS:       char      *szBuff   Returned string information
 *                                      char      *szStr    input string to be
formatted.
 *                                      int                 iLen      Length of
returned string
 *
 * RETURNS:                   none
 *
 * COMMENTS:        The length paramter is the absolute length of the returned
string in
 *                                      HTML characters. For example the input
string > would be returned as
 *                                      &gt; which would be counted as 1
character.If the number of input
 *                                      characters is less than the iLen parameter
spaces are appended to
 *                                      the end of the string to ensure that at
least iLen characters are
 *                                      returned in the szBuff parameter.
 *
 */

static void FormatHTMLString(char *szBuff, char *szStr, int iLen)
{
          while( iLen && *szStr )
          {
                    switch( *szStr )
                    {
                              case '>':
                                        *szBuff++ = '&';
                                        *szBuff++ = 'g';
                                        *szBuff++ = 't';
                                        *szBuff++ = ';';
                                        szStr++;
                                        break;
                              case '<':
                                        *szBuff++ = '&';
                                        *szBuff++ = 'l';
                                        *szBuff++ = 't';
                                        *szBuff++ = ';';
                                        szStr++;
                                        break;
                              case '&':
                                        *szBuff++ = '&';
                                        *szBuff++ = 'a';
                                        *szBuff++ = 'm';
                                        *szBuff++ = 'p';
                                        *szBuff++ = ';';
                                        szStr++;
                                        break;
                              case '\"':
                                        *szBuff++ = '&';
                                        *szBuff++ = 'q';
                                        *szBuff++ = 'u';
                                        *szBuff++ = 'o';
                                        *szBuff++ = 't';
                                        *szBuff++ = ';';
                                        szStr++;
                                        break;
                              default:
                                        *szBuff++ = *szStr++;
                                        break;
                    }
                    iLen--;
          }
          while( iLen-- )
                    *szBuff++ = ' ';

          *szBuff = 0;

          return;
}


static int ThrTpInit(EXTENSION_CONTROL_BLOCK *pECB)
{
     // use CS for tptpalloc ?
     static int num_tpinits=0;
     static int x=1;
     static int once=0;
     static CRITICAL_SECTION     TpCriticalSection;
           int iRc, TpRc, lasterr;
           char buf[40];
           int retry = 0;
           BOOL Success = FALSE;

     if(!TlsGetValue(TLSIsTpInitedKey))
     {
                    if (!once)
                    {
                              InitializeCriticalSection(&TpCriticalSection);
                              once=1;
                    }


                    if ( dLog )
                    {
                              FILE *fp;


                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* In ThrTpInit Thread %d * \r\n",
GetCurrentThreadId());
                              fclose(fp);
                    }

                    while ( retry <= TP_MAX_RETRIES )
                    {
                              EnterCriticalSection(&TpCriticalSection);

                              if(tpinf == NULL)
                              {
                                 if ((tpinf = ( TPINIT *)tpalloc("TPINIT", NULL,
sizeof(TPINIT))) == NULL)
                                 {
                                        TpRc = tperrno;
                                        {
                                                  FILE *fp;

                                                  fp = fopen(szTpccLogPath, "ab");
                                                  fprintf(fp, ">>>> ThrTpInit %d :
tpalloc of tpinit failed: %d \r\n",
                                                            GetCurrentThreadId(),
TpRc);
                                                  fclose(fp);
                                        }
                                        tpinf = NULL;
                                        retry++;
                              LeaveCriticalSection(&TpCriticalSection);
                                        Sleep(50); // Relinquish thread timeslice
                                        continue;
                                 }

                                 tpinf->flags|=TPMULTICONTEXTS;
                              }


// Do the TPINIT
                              itoa(++num_tpinits, tpinf->cltname, 10);
                              iRc = tpinit(tpinf);

                              // check tpinit() ?
                              if (iRc < 0 )
                              {
                                        lasterr = GetLastError();
                                        retry++ ;

                                        if ( dLog )
                                        {
                                                  FILE *fp;

                                                  fp = fopen(szTpccLogPath, "ab");
                                                  fprintf(fp, "* ThrTpInit Thread %d
Retry : count %d * \r\n",
                                                            GetCurrentThreadId(),
retry);
                                                  fclose(fp);
                                        }
                              }
                              else
                              {
                                        Success = TRUE;
                                        tpfree( ( char * ) tpinf);
                                        tpinf=NULL;
                                        break;
                              }
                              --num_tpinits;
```

```
                    LeaveCriticalSection(&TpCriticalSection);
                              Sleep(50);                    // Relinquish thread
timeslice

                    } // retry the tpinit if it failed the first time

          LeaveCriticalSection(&TpCriticalSection);

                    if ( Success == FALSE )
                    {
                              TpRc = tperrno;
                              {
                                        FILE *fp;

                                        fp = fopen(szTpccLogPath, "ab");
                                        fprintf(fp, ">>>> ThrTpInit Thread %d Failed
tperrno= %d Last Err= %d \r\n",
                                                  GetCurrentThreadId(), TpRc,
lasterr);
                                        fclose(fp);
                              }
                                        sprintf(buf,"Thread %d: In ThrTpInit iRc=%d
Tperrno=%d",
                                                  GetCurrentThreadId(),iRc, TpRc);
                                        ErrorMessage(pECB, ERR_TPINIT_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);

                              return -1;
                    }
                    if ( Success == TRUE )
                    {
                              if ( dLog )
                              {
                                        FILE *fp;

                                        fp = fopen(szTpccLogPath, "ab");
                                        fprintf(fp, "* ThrTpInit Thread %d Success
retry count %d * \r\n",
                                                  GetCurrentThreadId(), retry);
                                        fclose(fp);
                              }

                              if ( ( iRc=TlsSetValue(TLSIsTpInitedKey,&x)) == 0)
                              {
                                        {
                                                  FILE *fp;

                                                  fp = fopen(szTpccLogPath, "ab");
                                                  fprintf(fp, ">>>> ThrTpInit %d :
TlsSetValue Failed iRc: %d \r\n",
                                                  GetCurrentThreadId(), iRc);
                                                  fclose(fp);
                                        }
                                        sprintf(buf,"TlsSetValue %d",iRc);
                                        ErrorMessage(pECB, ERR_TPINIT_BAD,
ERR_TYPE_TUXEDO, buf, 0, 0);
                                        return -1;
                              }
                    }
          }
          else
          {
                    if ( dLog )
                    {
                              FILE *fp;

                              fp = fopen(szTpccLogPath, "ab");
                              fprintf(fp, "* ThrTpInit Thread %d already tpinited *
\r\n", GetCurrentThreadId());
                              fclose(fp);
                    }
          }

     return 0;
}
```

TPCC.DEF

```
LIBRARY TPCC.DLL

EXPORTS

          GetExtensionVersion @1
          HttpExtensionProc   @2
```

TPCC.H

```
/*        FILE:               TPCC.H
 *                                      Microsoft TPC-C Kit Ver. 3.00.000
 *                                      Audited 08/23/96, By Francois Raab
 *
 *                                      Copyright Microsoft, 1996
 *
 *        PURPOSE:  Header file for ISAPI TPCC.DLL, defines structures and functions
used in the isapi tpcc.dll.
 *        Author:             Philip Durr
 *                                      philipdu@Microsoft.com
 */

//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
          5

#define   ERR_TYPE_WEBDLL
          1
#define   ERR_TYPE_SQL
          2
#define   ERR_TYPE_DBLIB
          3
#define ERR_TYPE_TUXEDO
          4

#define   ERR_SUCCESS
                    1000      //"Success, no error.
#define   ERR_COMMAND_UNDEFINED                                                   1001
          //"Command undefined.
#define   ERR_NOT_IMPLEMENTED_YET                                                 1002
          //"Not Implemented Yet.
#define   ERR_CANNOT_INIT_TERMINAL                                      1003
          //"Cannot initialize client connection.
#define   ERR_OUT_OF_MEMORY                                                       1004
          //"insufficient memory.
#define   ERR_NEW_ORDER_NOT_PROCESSED                                   1005
          //"Cannot process new Order form.
#define   ERR_PAYMENT_NOT_PROCESSED                                     1006
          //"Cannot process payment form.
#define   ERR_NO_SERVER_SPECIFIED                                                 1007
          //"No Server name specified.
#define   ERR_ORDER_STATUS_NOT_PROCESSED                                1008
          //"Cannot process order status form.
#define   ERR_W_ID_INVALID                                                        1009
          //"Invalid Warehouse ID.
#define   ERR_CAN_NOT_SET_MAX_CONNECTIONS                               1010
          //"Insufficient memory to allocate # connections.
#define   ERR_NOSUCH_CUSTOMER                                                     1011
          //"No such customer.
#define   ERR_D_ID_INVALID                                                        1012
          //"Invalid District ID Must be 1 to 10.
#define   ERR_MAX_CONNECT_PARAM                                                   1013
          //"Max client connections exceeded, run install to increase.
#define   ERR_INVALID_SYNC_CONNECTION                                   1014
          //"Invalid Terminal Sync ID.
#define   ERR_INVALID_TERMID                                                      1015
          //"Invalid Terminal ID.
#define   ERR_PAYMENT_INVALID_CUSTOMER                        1016      //"Payment
Form, No such Customer.
#define   ERR_SQL_OPEN_CONNECTION                                                 1017
          //"SQLOpenConnection API Failed.
#define   ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY    1018      //"Stock Level missing
Threshold key "TT*".
#define   ERR_STOCKLEVEL_THRESHOLD_INVALID                    1019      //"Stock Level
Threshold invalid data type range = 1 - 99.
#define   ERR_STOCKLEVEL_THRESHOLD_RANGE                                1020
          //"Stock Level Threshold out of range, range must be 1 - 99.
#define   ERR_STOCKLEVEL_NOT_PROCESSED                        1021      //"Stock Level
not processed.
#define   ERR_NEWORDER_FORM_MISSING_DID                       1022      //"New Order
missing District key "DID*".
#define   ERR_NEWORDER_DISTRICT_INVALID                       1023      //"New Order
District ID Invalid range 1 - 10.
#define   ERR_NEWORDER_DISTRICT_RANGE                                   1024
          //"New Order District ID out of Range. Range = 1 - 10.
#define   ERR_NEWORDER_CUSTOMER_KEY                                     1025
          //"New Order missing Customer key "CID*".
#define   ERR_NEWORDER_CUSTOMER_INVALID                       1026      //"New Order
customer id invalid data type, range = 1 to 3000.
#define   ERR_NEWORDER_CUSTOMER_RANGE                                   1027
          //"New Order customer id out of range, range = 1 to 3000.
#define   ERR_NEWORDER_MISSING_IID_KEY                        1028      //"New Order
missing Item Id key "IID*".
#define   ERR_NEWORDER_ITEM_BLANK_LINES                       1029      //"New Order
blank order lines all orders must be continuous.
#define   ERR_NEWORDER_ITEMID_INVALID                                   1030
          //"New Order Item Id is wrong data type, must be numeric.
#define   ERR_NEWORDER_MISSING_SUPPW_KEY                                1031
          //"New Order missing Supp_W key "SP##*".
#define   ERR_NEWORDER_SUPPW_INVALID                                    1032
          //"New Order Supp_W invalid data type must be numeric.
```

```
#define   ERR_NEWORDER_MISSING_QTY_KEY                      1033      //"New Order
Missing Qty key "Qty##*".
#define   ERR_NEWORDER_QTY_INVALID                                    1034
          //"New Order Qty invalid must be numeric range 1 - 99.
#define   ERR_NEWORDER_SUPPW_RANGE                                    1035
          //"New Order Supp_W value out of range range = 1 - Max Warehouses.
#define   ERR_NEWORDER_ITEMID_RANGE                                   1036
          //"New Order Item Id is out of range. Range = 1 to 999999.
#define   ERR_NEWORDER_QTY_RANGE                                                1037
          //"New Order Qty is out of range. Range = 1 to 99.
#define   ERR_PAYMENT_DISTRICT_INVALID                      1038      //"Payment
District ID is invalid must be 1 - 10.
#define   ERR_NEWORDER_SUPPW_WITHOUT_ITEMID                 1039      //"New Order
Supp_W field entered without a corrisponding Item_Id.
#define   ERR_NEWORDER_QTY_WITHOUT_ITEMID                             1040
          //"New Order Qty entered without a corrisponding Item_Id.
#define   ERR_NEWORDER_NOITEMS_ENTERED                      1041      //"New Order
Blank Items between items, items must be continuous.
#define   ERR_PAYMENT_MISSING_DID_KEY                                 1042
          //"Payment missing District Key "DID*".
#define   ERR_PAYMENT_DISTRICT_RANGE                                  1043
          //"Payment District Out of range, range = 1 - 10.
#define   ERR_PAYMENT_MISSING_CID_KEY                                 1044
          //"Payment missing Customer Key "CID*".
#define   ERR_PAYMENT_CUSTOMER_INVALID                      1045      //"Payment
Customer data type invalid, must be numeric.
#define   ERR_PAYMENT_MISSING_CLT                                               1046
          //"Payment missing Customer Last Name Key "CLT*".
#define   ERR_PAYMENT_LAST_NAME_TO_LONG                     1047      //"Payment
Customer last name longer than 16 characters.
#define   ERR_PAYMENT_CUSTOMER_RANGE                                  1048
          //"Payment Customer ID out of range, must be 1 to 3000.
#define   ERR_PAYMENT_CID_AND_CLT                                               1049
          //"Payment Customer ID and Last Name entered must be one or other.
#define   ERR_PAYMENT_MISSING_CDI_KEY                                 1050
          //"Payment missing Customer district key "CDI*".
#define   ERR_PAYMENT_CDI_INVALID                                               1051
          //"Payment Customer district invalid must be numeric.
#define   ERR_PAYMENT_CDI_RANGE                                                 1052
          //"Payment Customer district out of range must be 1 - 10.
#define   ERR_PAYMENT_MISSING_CWI_KEY                                 1053
          //"Payment missing Customer Warehouse key "CWI*".
#define   ERR_PAYMENT_CWI_INVALID                                               1054
          //"Payment Customer Warehouse invalid must be numeric.
#define   ERR_PAYMENT_CWI_RANGE                                                 1055
          //"Payment Customer Warehouse out of range, 1 to Max Warehouses.
#define   ERR_PAYMENT_MISSING_HAM_KEY                                 1056
          //"Payment missing Amount key "HAM*".
#define   ERR_PAYMENT_HAM_INVALID                                               1057
          //"Payment Amount invalid data type must be numeric.
#define   ERR_PAYMENT_HAM_RANGE                                                 1058
          //"Payment Amount out of range, 0 - 9999.99.
#define   ERR_ORDERSTATUS_MISSING_DID_KEY                             1059
          //"Order Status missing District key "DID*".
#define   ERR_ORDERSTATUS_DID_INVALID                                 1060
          //"Order Status District invalid, value must be numeric 1 - 10.
#define   ERR_ORDERSTATUS_DID_RANGE                                   1061
          //"Order Status District out of range must be 1 - 10.
#define   ERR_ORDERSTATUS_MISSING_CID_KEY                             1062
          //"Order Status missing Customer key "CID*".
#define   ERR_ORDERSTATUS_MISSING_CLT_KEY                             1063
          //"Order Status missing Customer Last Name key "CLT*".
```

```
#define   ERR_ORDERSTATUS_CLT_RANGE                                             1064
          //"Order Status Customer last name longer than 16 characters.
#define   ERR_ORDERSTATUS_CID_INVALID                                           1065
          //"Order Status Customer ID invalid, range must be numeric 1 - 3000.
#define   ERR_ORDERSTATUS_CID_RANGE                                             1066
          //"Order Status Customer ID out of range must be 1 - 3000.
#define   ERR_ORDERSTATUS_CID_AND_CLT                                           1067
          //"Order Status Customer ID and LastName entered must be only one."
#define   ERR_DELIVERY_MISSING_OCD_KEY                      1068      //"Delivery
missing Carrier ID key \"OCD*\".
#define   ERR_DELIVERY_CARRIER_INVALID                      1069      //"Delivery
Carrier ID invalid must be numeric 1 - 10.
#define   ERR_DELIVERY_CARRIER_ID_RANGE                     1070      //"Delivery
Carrier ID out of range must be 1 - 10.
#define ERR_PAYMENT_MISSING_CLT_KEY                                             1071
          //"Payment missing Customer Last Name key "CLT*".
#define ERR_TPINIT_BAD
          5001      //"Bad TPINIT"
#define ERR_TPALLOC_BAD
          5002      //"Bad TPALLOC"
#define ERR_TPCALL_BAD
          5003      //"Bad TPCALL"

//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 is used for posting delivery transactions
typedef struct _DELIVERY_TRANSACTION
{
          SYSTEMTIME          queue;                        //time delivery
transaction queued
          short               w_id;                         //delivery warehouse
          short               o_carrier_id;       //carrier id
} DELIVERY_TRANSACTION;

//error message structure used in ErrorMessage API
typedef struct _SERRORMSG
{
          int                 iError;                                 //error id of
message
          char      szMsg[80];                              //message to sent to
browser
} SERRORMSG;
```

```
//This structure defines the data necessary to keep distinct for each terminal or
client connection.
typedef struct _CLIENTDATA
{
          int                           inUse;
          //in use flag allows client entries to be reused
          int                           w_id;
          //warehouse id assigned at welcome form
          int                           d_id;
          //district id assigned at welcome form

          int                           bDeadlock;
          int                           bFailed;
          short               DeadlockRetry;
          int                           Error;
          int                           retval;

          DBPROCESS *dbproc;                                //dblib connection
pointer
          int                           spid;
          //spid assigned from dblib
          int                           iSyncId;
          //syncronization id
          int                           iTickCount;
          //time of last access;
          int                           iTermId;
          //terminal id of http stream connection

          char                                    szBuffer[4096];     //form buffer
each HTML form is built for a client in here

          NEW_ORDER_DATA                NewOrderData;                 //new order
form data
          PAYMENT_DATA                  PaymentData;                  //payment form
data
          ORDER_STATUS_DATA   OrderStatusData;    //order status form data
          DELIVERY_DATA                 DeliveryData;                 //delivery
form data
          STOCK_LEVEL_DATA    StockLevelData;               //stock level form data

#ifndef LOCAL_ALLOC
          NEW_ORDER_DATA                *NewOrderDataPtr;   //New Order Tuxedo
Buffer
          PAYMENT_DATA                  *PaymentDataPtr;    //Payment Tuxedo Buffer
          ORDER_STATUS_DATA   *OrderStatusDataPtr;          //Order Status Tuxedo
Buffer
          DELIVERY_DATA                 *DeliveryDataPtr;   //Delivery Tuxedo Buffer
          STOCK_LEVEL_DATA    *StockLevelDataPtr; //Stock Level Tuxedo Buffer
#endif

} CLIENTDATA;

typedef CLIENTDATA *PCLIENTDATA;                            //pointer to client
structure

//This structure is used to define the operational interface for terminal id support
typedef struct _TERM
{
          int                           iAvailable;
                              //total allocated terminal array entries
          int                           iNext;
                              //next available terminal array element
```

```
          int                           iMasterSyncId;
                    //syncronization id
          BOOL                bInit;
                    //structure has been initialized flag
          CLIENTDATA          *pClientData;
          //pointer to allocated client data
          void                (*Init)(void);
          //API to initialize this structure
          int                           (*Allocate)(void);
          //API to allocate a new terminal entry array id returned
          void                (*Restore)(void);
          //API to free terminal data
          int                           (*Add)(EXTENSION_CONTROL_BLOCK *pECB, char
*pQueryString);     //API to add a terminal id to array, this context will

                                        //be passed from the browser to the tpcc.dll
in the

                                        //TERMID= key in the HTTP string.
          void                (*Delete)(EXTENSION_CONTROL_BLOCK *pECB, int id);
          //API to free resources used by a terminal array entry
} TERM;

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

#ifdef USE_ODBC
          typedef struct _DBPROCESS
          {
                    HDBC                                              hdbc;
                    HSTMT                                             hstmt;
                    int
          spid;
                    int
          iTermId;
                    int
          iSyncId;
                    BOOL                                              bDeadlock;
                    EXTENSION_CONTROL_BLOCK       *pECB;
          } DBPROCESS, *PDBPROCESS;
#else
          //this structure allows the EXTENSION CONTROL BLOCK to be passed to the
msg and error handlers.
          typedef struct _ECBINFO
          {
                    int
          iTermId;  //terminal id
                    int
          iSyncId;  //browser sync id
                    BOOL                                              bDeadlock;
          //deadlock condition flag
                    BOOL                                              bFailed;
          //cleared before sql transaction, set in err handlers if an error occurs
                    EXTENSION_CONTROL_BLOCK *pECB;                    //inetsrv
current connection structure information
          } ECBINFO, *PECBINFO;
#endif

//function prototypes
BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call, LPVOID lpReserved);
static void DeliveryDisconnect(void *ptr);
static BOOL IsValidTermId(int TermId);
```

```
BOOL ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd, int *pFormId, int
*pTermId, int *pSyncId);
void NewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int
iSyncId);
void PaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int
iSyncId);
void DeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int
iSyncId);
void OrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int
iSyncId);
void StockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int
iSyncId);
void Exitcmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int iSyncId);
void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int
iSyncId);
void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int iSyncId);
void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int
iSyncId);
void ClearCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int iSyncId);
void MenuCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int iSyncId);
static void WriteZString(EXTENSION_CONTROL_BLOCK *pECB, char *szStr);
static void h_printf(EXTENSION_CONTROL_BLOCK *pECB, char *format, ...);
void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int iError, int iErrorType, char
*szMsg, int iTermId, int iSyncId);
static BOOL GetKeyValue(char *pQueryString, char *pKey, char *pValue, int iMax);
static void TermInit(void);
int err_handler(DBPROCESS *dbproc, int severity, int dberr, int oserr, char
*dberrstr, char *oserrstr);
int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int severity, char
*msgtext);
static void TermRestore(void);
static int TermAllocate(void);
static int TermAdd(EXTENSION_CONTROL_BLOCK *pECB, char *pQueryString);
static void TermDelete(EXTENSION_CONTROL_BLOCK *pECB, int id);
BOOL Init(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int iSyncId, char *szServer,
char *szUser, char *szPassword, char *szDatabase);
static BOOL Close(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int iSyncId);
static BOOL SQLOpenConnection(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId, DBPROCESS **dbproc, char *server, char *database, char *user, char
*password, char *app, int *spid);
static BOOL SQLCloseConnection(EXTENSION_CONTROL_BLOCK *pECB, DBPROCESS *dbproc);
static BOOL SQLStockLevel(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int iSyncId,
DBPROCESS *dbproc, STOCK_LEVEL_DATA *pStockLevel, short deadlock_retry);
static int SQLNewOrder(EXTENSION_CONTROL_BLOCK    *pECB, int iTermId, int iSyncId,
DBPROCESS *dbproc, NEW_ORDER_DATA  *pNewOrder, short deadlock_retry);
static int SQLPayment(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int iSyncId,
DBPROCESS *dbproc, PAYMENT_DATA *pPayment, short deadlock_retry);
static int SQLOrderStatus(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int iSyncId,
DBPROCESS *dbproc, ORDER_STATUS_DATA *pOrderStatus, short deadlock_retry);
static int SQLDelivery(DBPROCESS *dbproc, DELIVERY_DATA *pDelivery, short
deadlock_retry);
static void WriteLog(DELIVERY_DATA *pDelivery, SYSTEMTIME *trans_end );
static void CalculateElapsedTime(int *pElapsed, LPSYSTEMTIME lpBegin, LPSYSTEMTIME
lpEnd);
BOOL SQLDetectDeadlock(DBPROCESS *dbproc);
static void FormatString(char *szDest, char *szPic, char *szSrc);
static char *MakeStockLevelForm(int iTermId, int iSyncId, BOOL bInput);
static char *MakeMainMenuForm(int iTermId, int iSyncId);
static char *MakeWelcomeForm(void);
static char *MakeNewOrderForm(int iTermId, int iSyncId, BOOL bInput, BOOL bValid);
static char *MakeNewOrderInputForm(int iTermId, int iSyncId, BOOL bInput, BOOL
bValid);
static char *MakeNewOrderOutputForm(int iTermId, int iSyncId, BOOL bInput, BOOL
bValid);
static char *MakePaymentForm(int iTermId, int iSyncId, BOOL bInput);
static char *MakeOrderStatusForm(int iTermId, int iSyncId, BOOL bInput);
static char *MakeDeliveryForm(int iTermId, int iSyncId, BOOL bInput);
static void UtilStrCpy(char * pDest, char * pSrc, int n);
static void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK     *pECB, int iTermId, int
iSyncId);
static void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId);
static void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId);
static void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId);
static void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, int
iSyncId);
static int GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA *pNewOrderData);
static int GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA *pPaymentData);
static int GetOrderStatusData(LPSTR lpszQueryString, ORDER_STATUS_DATA
*pOrderStatusData);
static BOOL ReadRegistrySettings(void);
static BOOL PostDeliveryInfo(short w_id, short o_carrier_id);
static BOOL IsNumeric(char *ptr);
static int ODBCError(DBPROCESS *dbproc);
static void FormatHTMLString(char *szBuff, char *szStr, int iLen);
static void         PrintParameters(void);
```

TPCC.RC

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

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

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

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

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

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

VS_VERSION_INFO VERSIONINFO
 FILEVERSION 0,3,0,2
 PRODUCTVERSION 0,3,0,2
 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\0"
            VALUE "CompanyName", "Microsoft\0"
            VALUE "FileDescription", "tpcc\0"
            VALUE "FileVersion", "0, 3, 0, 2\0"
            VALUE "InternalName", "tpcc\0"
            VALUE "LegalCopyright", "Copyright © 1996\0"
            VALUE "OriginalFilename", "tpcc.dll\0"
            VALUE "ProductName", "Microsoft tpcc\0"
            VALUE "ProductVersion", "0, 3, 0, 2\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

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

```

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


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

TRANS.H

```
/*        FILE:               TRANS.H
 *                                      Microsoft TPC-C Kit Ver. 3.00.000
 *                                      Audited 08/23/96    By Francois Raab
 *        PURPOSE:  Header file for ISAPI TPCC.DLL, defines structures and functions
used in the isapi tpcc.dll.
 *
 *                                      Copyright Microsoft inc. 1996, All Rights
Reserved
 *
 *        Author:             PhilipDu, from tpcc.h by DamienL
 *                                      DamienL@Microsoft.com
 *                                      philipdu@Microsoft.com
 */

#ifndef _INC_TRANS

          #define _INC_TRANS

          #ifdef USE_ODBC
                    #ifndef TIMESTAMP_STRUCT
                              #include <sqltypes.h>
                    #endif
          #else
                    #ifndef _INC_SQLFRONT
                              #include <sqlfront.h>
                    #endif
          #endif

          #ifndef DBINT
                    typedef long DBINT;
          #endif

          #define   DEFCLPACKSIZE                                 4096
          #define   DEADLOCKWAIT                                  10

          // 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

// transaction structures

typedef struct
{
          short                                   ol_supply_w_id;
          long                                    ol_i_id;
          char                                    ol_i_name[I_NAME_LEN+1];
          short                                   ol_quantity;
          char
ol_brand_generic[BRAND_LEN+1];
          double                                  ol_i_price;
          double                                  ol_amount;
          short                                   ol_stock;
          short                                   num_warehouses;
} OL_NEW_ORDER_DATA;

typedef struct
{
          short               w_id;
          short               d_id;
          long                c_id;
          short               o_ol_cnt;
          char                c_last[LAST_NAME_LEN+1];
          char                c_credit[CREDIT_LEN+1];
          double              c_discount;
          double              w_tax;
          double              d_tax;
          long                          o_id;
          short                                   o_commit_flag;
#ifdef USE_ODBC
          TIMESTAMP_STRUCT    o_entry_d;
#else
          DBDATEREC           o_entry_d;
#endif
          short               o_all_local;
          double              total_amount;
          long                num_deadlocks;
          int                                               retval;
          char
execution_status[STATUS_LEN];
          OL_NEW_ORDER_DATA   Ol[MAX_OL_NEW_ORDER_ITEMS];
} NEW_ORDER_DATA;

typedef struct
{
```

```
                    short                                   w_id;
                    short                                   d_id;
                    long                                    c_id;
                    short                                   c_d_id;
                    short                                   c_w_id;
                    double                                  h_amount;
          #ifdef USE_ODBC
                    TIMESTAMP_STRUCT     h_date;
          #else
                    DBDATEREC                    h_date;
          #endif
                    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_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];
          #ifdef USE_ODBC
                    TIMESTAMP_STRUCT     c_since;
          #else
                    DBDATEREC                         c_since;
          #endif
                    char                                    c_credit[CREDIT_LEN+1];
                    double                                  c_credit_lim;
                    double                                  c_discount;
                    double                                  c_balance;
                    char                                    c_data[200+1];

                    long                                    num_deadlocks;
                    int                                               retval;
                    char
execution_status[STATUS_LEN];
          } PAYMENT_DATA;

          typedef struct
          {
                    long                                   ol_i_id;
                    short                                  ol_supply_w_id;
                    short                                  ol_quantity;
                    double                                 ol_amount;
          #ifdef USE_ODBC
                    TIMESTAMP_STRUCT    ol_delivery_d;
          #else
                    DBDATEREC                                ol_delivery_d;
          #endif
          } OL_ORDER_STATUS_DATA;

          typedef struct
          {
                    short                 w_id;
                    short                 d_id;
                    long                  c_id;
                    char                  c_first[FIRST_NAME_LEN+1];
                    char                  c_middle[MIDDLE_NAME_LEN+1];
                    char                  c_last[LAST_NAME_LEN+1];
                    double                c_balance;
                    long                  o_id;
          #ifdef USE_ODBC
                    TIMESTAMP_STRUCT      o_entry_d;
          #else
                    DBDATEREC             o_entry_d;
          #endif
                    short                 o_carrier_id;
                    OL_ORDER_STATUS_DATA
OlOrderStatusData[MAX_OL_ORDER_STATUS_ITEMS];
                    short                 o_ol_cnt;
                    long                  num_deadlocks;
                    int                                                  retval;
                    char                                execution_status[STATUS_LEN];
          } ORDER_STATUS_DATA;

          typedef struct
          {
                    long                                     o_id;
          } DEL_ITEM;

          typedef struct
          {
                    short                          w_id;
                    short                          o_carrier_id;
                    SYSTEMTIME                     queue_time;
                    long                           num_deadlocks;
                    long                                     o_id[10];
                    int                                                 retval;
                    char                           execution_status[STATUS_LEN];
          } DELIVERY_DATA;

          typedef struct
          {
                    short                                    w_id;
                    short                                    d_id;
                    short                                    thresh_hold;
                    long                                     low_stock;
                    long                                     num_deadlocks;
                    int                                                  retval;
                    char
          execution_status[STATUS_LEN];
          } STOCK_LEVEL_DATA;

#endif
```

Appendix B: Database Design

The TPC-C database was created with the following Transact-SQL scripts:

DBOPT1.SQL

```
--  File:     DBOPT1.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.00
--            Copyright Microsoft, 1996
--  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. 4.01
--            Copyright Microsoft, 1996
--  Purpose:  Resets database options after data load


use master
go

sp_dboption tpcc,’select ’,false
go

sp_dboption tpcc,’trunc. ’,false
go

use tpcc
go

checkpoint
go
```

```
sp_configure allow,1
go

reconfigure with override
go

/*                                            */
/* Set option values for user-defined indexes */
/*                                            */

sp_indexoption ’customer’,’AllowPageLocks’,FALSE
go
sp_indexoption ’customer’,’AllowRowLocks’,TRUE
go
sp_indexoption ’district’,’AllowPageLocks’,FALSE
go
sp_indexoption ’warehouse’,’AllowPageLocks’,FALSE
go
sp_indexoption ’stock’,’AllowPageLocks’,FALSE
go
sp_indexoption ’stock’,’AllowRowLocks’,TRUE
go
sp_indexoption ’order_line’,’AllowPageLocks’,TRUE
go
sp_indexoption ’order_line’,’AllowRowLocks’,FALSE
go
sp_indexoption ’orders’,’AllowPageLocks’,TRUE
go
sp_indexoption ’orders’,’AllowRowLocks’,FALSE
go
sp_indexoption ’new_order’,’AllowRowLocks’,FALSE
go
sp_indexoption ’item’,’AllowRowLocks’,FALSE
go
sp_indexoption ’item’,’AllowPageLocks’,FALSE
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,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. 4.00
--            Copyright Microsoft, 1996
--  Purpose:  Creates delivery transaction stored procedure


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 updlock)
        where no_w_id = @w_id and
                          no_d_id = @d_id
                    order by no_o_id asc

        if (@@rowcount <> 0)
        begin

--  claim the order for this district

            delete new_order
            where no_w_id = @w_id and
                              no_d_id = @d_id and
                                    no_o_id = @o_id

--  set carrier_id on this order (and get customer id)

            update orders
                        set o_carrier_id = @o_carrier_id,
                @c_id        = o_c_id
            where o_w_id = @w_id and
                              o_d_id = @d_id and
                                    o_id   = @o_id

--  set date in all lineitems for this order (and sum amounts)

            update order_line
                        set ol_delivery_d = getdate(),
                      @total        = @total + ol_amount
            where ol_w_id = @w_id and
                  ol_d_id = @d_id and
                              ol_o_id = @o_id

--  accummulate lineitem amounts for this order into customer

                        update customer
                              set c_balance      = c_balance + @total,
                                  c_delivery_cnt = c_delivery_cnt + 1
            where c_w_id = @w_id and
                              c_d_id = @d_id and
                                    c_id   = @c_id

        end

        select @oid1 = case @d_id when  1  then @o_id else @oid1 end,
               @oid2 = case @d_id when  2  then @o_id else @oid2 end,
               @oid3 = case @d_id when  3  then @o_id else @oid3 end,
               @oid4 = case @d_id when  4  then @o_id else @oid4 end,
               @oid5 = case @d_id when  5  then @o_id else @oid5 end,
               @oid6 = case @d_id when  6  then @o_id else @oid6 end,
               @oid7 = case @d_id when  7  then @o_id else @oid7 end,
               @oid8 = case @d_id when  8  then @o_id else @oid8 end,
               @oid9 = case @d_id when  9  then @o_id else @oid9 end,
               @oid10 = case @d_id when 10 then @o_id else @oid10 end

    end

commit tran d

-- return delivery data to client

select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

go
```

IDXCUSCL.SQL

```
--  File:     IDXCUSCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.00
--            Copyright Microsoft, 1996
--  Purpose:  Creates clustered index on customer table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name = ’customer_c1’ )
          drop index customer.customer_c1

create unique clustered index customer_c1 on customer(c_w_id, c_d_id, c_id)
          on big_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

IDXCUSNC.SQL

```
--  File:     IDXCUSNC.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.00
--            Copyright Microsoft, 1996
--  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 big_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

IDXDISCL.SQL

```
--  File:     IDXDISCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.00
--            Copyright Microsoft, 1996
--  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 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. 4.00
--            Copyright Microsoft, 1996
--  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 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. 4.00
--            Copyright Microsoft, 1996
--  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 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. 4.00
--            Copyright Microsoft, 1996
--  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 misc_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. 4.00
--            Copyright Microsoft, 1996
--  Purpose:  Creates clustered index on warehouse table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where name =
’warehouse_c1’ )
                                drop index warehouse.warehouse_c1

create unique clustered index warehouse_c1 on warehouse(w_id)
                                with fillfactor=100 on 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. 4.00
--            Copyright Microsoft, 1996
--  Purpose:  Creates clustered index on orders table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)
```

```
if exists ( select name from sysindexes where  name = 'orders_c1' )
          drop index orders.orders_c1

create unique clustered index orders_c1 on orders(o_w_id, o_d_id, o_id)
          on 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. 4.00
--            Copyright Microsoft, 1996
--  Purpose:  Creates non-clustered index on orders table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'orders_nc1' )
          drop index orders.orders_nc1

create index orders_nc1 on orders(o_w_id, o_d_id, o_c_id, o_id)
          on 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. 4.00
--            Copyright Microsoft, 1996
--  Purpose:  Creates clustered index on stock table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'stock_c1' )
          drop index stock.stock_c1

create unique clustered index stock_c1 on stock(s_i_id, s_w_id)
          on big_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. 4.00
--            Copyright Microsoft, 1996
--  Purpose:  Creates new order transaction stored procedure
--
--  Modified 6/24/98 - Jamie Reding - Microsoft Corporation
--        Replaced Select of @s_quantity with Select of @li_qty to insure
--        correct data when sending line-item data to client.
--

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 availible order id and update
--  plus initialize local variables

    update district
          set @d_tax      = d_tax,
              @o_id       = d_next_o_id,
                    d_next_o_id = d_next_o_id + 1,
                    @o_entry_d = getdate(),
                    @li_no = 0,
                    @commit_flag = 1
              where d_w_id = @w_id and
                    d_id   = @d_id

--  process orderlines

    while (@li_no < @o_ol_cnt)
          begin

        select @li_no = @li_no + 1

--  set i_id, s_w_id, and qty for this lineitem
```

```
              select @li_id = case @li_no
                      when 1 then @i_id1
                                  when 2 then @i_id2
                                  when 3 then @i_id3
                                  when 4 then @i_id4
                                  when 5 then @i_id5
                                  when 6 then @i_id6
                                  when 7 then @i_id7
                                  when 8 then @i_id8
                                  when 9 then @i_id9
                                  when 10 then @i_id10
                                  when 11 then @i_id11
                                  when 12 then @i_id12
                                  when 13 then @i_id13
                                  when 14 then @i_id14
                                  when 15 then @i_id15
                                  end,

                                  @li_s_w_id = case @li_no
                                  when 1 then @s_w_id1
                                  when 2 then @s_w_id2
                                  when 3 then @s_w_id3
                                  when 4 then @s_w_id4
                                  when 5 then @s_w_id5
                                  when 6 then @s_w_id6
                                  when 7 then @s_w_id7
                                  when 8 then @s_w_id8
                                  when 9 then @s_w_id9
                                  when 10 then @s_w_id10
                                  when 11 then @s_w_id11
                                  when 12 then @s_w_id12
                                  when 13 then @s_w_id13
                                  when 14 then @s_w_id14
                                  when 15 then @s_w_id15
                                  end,

                                  @li_qty = case  @li_no
                                  when 1 then @ol_qty1
                                  when 2 then @ol_qty2
                                  when 3 then @ol_qty3
                                  when 4 then @ol_qty4
                                  when 5 then @ol_qty5
                                  when 6 then @ol_qty6
                                  when 7 then @ol_qty7
                                  when 8 then @ol_qty8
                                  when 9 then @ol_qty9
                                  when 10 then @ol_qty10
                                  when 11 then @ol_qty11
                                  when 12 then @ol_qty12
                                  when 13 then @ol_qty13
                                  when 14 then @ol_qty14
                                  when 15 then @ol_qty15
                                  end

--  get item data (no one updates item)

        select @i_price = i_price,
               @i_name  = i_name,
               @i_data  = i_data
                    from item (tablock repeatableread)
                    where i_id = @li_id

--  if there actually is an item with this id, go to work
```

```
                    if (@@rowcount > 0)
                    begin
            update stock set s_ytd        = s_ytd + @li_qty,
                             @s_quantity  = s_quantity,
                             s_quantity   = s_quantity - @li_qty +
                                 case when (s_quantity - @li_qty < 10) then 91 else
0 end,
                             s_order_cnt  = s_order_cnt + 1,
                             s_remote_cnt = s_remote_cnt + case
                                 when (@li_s_w_id = @w_id) then 0 else 1 end,
                             @s_data      = s_data,
                             @s_dist      = case @d_id
                                                       when 1  then s_dist_01
                                                       when 2  then s_dist_02
                                                       when 3  then s_dist_03
                                                       when 4  then s_dist_04
                                                       when 5  then s_dist_05
                                                       when 6  then s_dist_06
                                                       when 7  then s_dist_07
                                                       when 8  then s_dist_08
                                                       when 9  then s_dist_09
                                                       when 10 then s_dist_10
                                           end
            where s_i_id = @li_id and
                                    s_w_id = @li_s_w_id

--  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,
                   @li_qty,
                   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 found - triggers rollback condition

                              select "",0,"",0,0
                              select @commit_flag = 0

                    end
          end
```

```
--  get customer last name, discount, and credit rating

    select @c_last     = c_last,
           @c_discount = c_discount,
           @c_credit   = c_credit,
                       @c_id_local = c_id
    from customer (repeatableread)
    where c_id   = @c_id and
          c_w_id = @w_id and
          c_d_id = @d_id

--  insert fresh row into orders table

    insert into orders values (@o_id,
                                       @d_id,
                                               @w_id,
                                                         @c_id_local,
                                                         @o_entry_d,
                                                             0,
                                                             @o_ol_cnt,
                                                             @o_all_local)

--  insert corresponding row into new-order table

          insert into new_order values (@o_id,

@d_id,

@w_id)

--  select warehouse tax

          select @w_tax = w_tax
          from warehouse (repeatableread)
          where w_id = @w_id

          if (@commit_flag = 1)
                    commit transaction n
          else

--  all that work for nuthin!!!

                    rollback transaction n

--  return order data to client

          select @w_tax,
                 @d_tax,
                       @o_id,
                       @c_last,
                       @c_discount,
                       @c_credit,
                       @o_entry_d,
                       @commit_flag

end

go
```

ORDSTAT.SQL

```
--  File:     ORDSTAT.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.00
--            Copyright Microsoft, 1996
--  Purpose:  Creates order status transaction stored procedure


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. 4.00
--            Copyright Microsoft, 1996
--  Purpose:  Creates payment transaction stored procedure


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
```

STOCKLEV.SQL

```
--  File:     STOCKLEV.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.00
--            Copyright Microsoft, 1996
--  Purpose:  Creates stock level transaction stored procedure


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,
                                                            @threshhold
          smallint
as

          declare @o_id_low int,
                        @o_id_high int

    select @o_id_low  = (d_next_o_id - 20),
                 @o_id_high = (d_next_o_id - 1)
    from district
    where d_w_id = @w_id and
                      d_id   = @d_id

    select count(distinct(s_i_id))
          from  stock, order_line
    where ol_w_id    =  @w_id and
                ol_d_id    =  @d_id and
                ol_o_id between @o_id_low and @o_id_high and
          s_w_id     =  ol_w_id and
                s_i_id     =  ol_i_id and
                s_quantity <  @threshhold

go



TABLES.SQL

--  File:     TABLES.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.00
--            Copyright Microsoft, 1996
--  Purpose:  Creates TPC-C tables


use tpcc
go

if exists ( select name from sysobjects where name = 'warehouse' )
          drop table warehouse
go
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 misc_fg
go

if exists ( select name from sysobjects where name = 'district' )
          drop table district
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 misc_fg
go

if exists ( select name from sysobjects where name = 'customer' )
          drop table customer
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 big_fg
go

if exists ( select name from sysobjects where name = 'history' )
          drop table history
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 misc_fg
go

if exists ( select name from sysobjects where name = 'new_order' )
          drop table new_order
```

```
go
create table new_order
(
          no_o_id                                 int,
          no_d_id                                 tinyint,
          no_w_id                                 smallint
) on misc_fg
go

if exists ( select name from sysobjects where name = 'orders' )
          drop table orders
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 misc_fg
go

if exists ( select name from sysobjects where name = 'order_line' )
          drop table order_line
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 misc_fg
go

if exists ( select name from sysobjects where name = 'item' )
          drop table item
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 misc_fg
go

if exists ( select name from sysobjects where name = 'stock' )
          drop table stock
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 big_fg
go
```

CREATEDB.SQL

```
--  File:     CREATEDB.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.00
--            Copyright Microsoft, 1996
--  Purpose:  Creates tpcc database and backup files


use master
go

--  remove any existing database and backup files

exec sp_dbremove tpcc, dropdev
exec sp_dropdevice 'tpccback1'
exec sp_dropdevice 'tpccback2'
exec sp_dropdevice 'tpccback3'
exec sp_dropdevice 'tpccback4'
go


declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)


--  create main database files

create database tpcc on
          (name="MSSQL70_tpcc_root",filename="C:\MSSQL70_tpcc_root.mdf",size=8MB,
FILEGROWTH=0)
log on
          (name="MSSQL70_tpcc_log",filename="F:",size=52000MB, FILEGROWTH=0)


--  create filegroups
alter database tpcc add filegroup misc_fg
alter database tpcc add filegroup big_fg
```

```
--  add files to filegroups
--  10028 * 7
--  8775 * 8
alter database tpcc add file
          (name="misc1",filename="N:",size=7875MB, FILEGROWTH=0),
          (name="misc2",filename="O:",size=7875MB, FILEGROWTH=0),
          (name="misc3",filename="P:",size=7875MB, FILEGROWTH=0),
          (name="misc4",filename="Q:",size=7875MB, FILEGROWTH=0),
          (name="misc5",filename="R:",size=7875MB, FILEGROWTH=0),
          (name="misc6",filename="S:",size=7875MB, FILEGROWTH=0),
          (name="misc7",filename="T:",size=7875MB, FILEGROWTH=0)
to filegroup misc_fg

-- 15425 * 7
alter database tpcc add file
          (name="stock1",filename="G:",size=16800MB, FILEGROWTH=0),
          (name="stock2",filename="H:",size=16800MB, FILEGROWTH=0),
          (name="stock3",filename="I:",size=16800MB, FILEGROWTH=0),
          (name="stock4",filename="J:",size=16800MB, FILEGROWTH=0),
          (name="stock5",filename="K:",size=16800MB, FILEGROWTH=0),
          (name="stock6",filename="L:",size=16800MB, FILEGROWTH=0),
          (name="stock7",filename="M:",size=16800MB, FILEGROWTH=0)
to filegroup big_fg



select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)
go

--  create backup devices

exec sp_addumpdevice 'disk','tpccback1','y:\tpccback1.dmp'
exec sp_addumpdevice 'disk','tpccback2','y:\tpccback2.dmp'
exec sp_addumpdevice 'disk','tpccback3','z:\tpccback3.dmp'
exec sp_addumpdevice 'disk','tpccback4','z:\tpccback4.dmp'
go
```

Database Loader Source

TPCCLDR.MAK

```
# Microsoft Developer Studio Generated NMAKE File, Format Version 4.10
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Console Application" 0x0103

!IF "$(CFG)" == ""
CFG=tpccldr - Win32 Debug
!MESSAGE No configuration specified.  Defaulting to tpccldr - Win32 Debug.
!ENDIF

!IF "$(CFG)" != "tpccldr - Win32 Release" && "$(CFG)" !=\
 "tpccldr - Win32 Debug"
!MESSAGE Invalid configuration "$(CFG)" specified.
!MESSAGE You can specify a configuration when running NMAKE on this makefile
!MESSAGE by defining the macro CFG on the command line.  For example:
!MESSAGE
!MESSAGE NMAKE /f "tpccldr.mak" CFG="tpccldr - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpccldr - Win32 Release" (based on "Win32 (x86) Console Application")
!MESSAGE "tpccldr - Win32 Debug" (based on "Win32 (x86) Console Application")
!MESSAGE
!ERROR An invalid configuration is specified.
!ENDIF

!IF "$(OS)" == "Windows_NT"
NULL=
!ELSE
NULL=nul
!ENDIF
################################################################################
# Begin Project
# PROP Target_Last_Scanned "tpccldr - Win32 Debug"
RSC=rc.exe
CPP=cl.exe

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir "bin"
# PROP Intermediate_Dir "objects"
# PROP Target_Dir ""
OUTDIR=.\bin
INTDIR=.\objects
```

```
ALL : "$(OUTDIR)\tpccldr.exe"

CLEAN :
          -@erase "$(INTDIR)\getargs.obj"
          -@erase "$(INTDIR)\random.obj"
          -@erase "$(INTDIR)\strings.obj"
          -@erase "$(INTDIR)\time.obj"
          -@erase "$(INTDIR)\tpccldr.obj"
          -@erase "$(OUTDIR)\tpccldr.exe"

"$(OUTDIR)" :
    if not exist "$(OUTDIR)/$(NULL)" mkdir "$(OUTDIR)"

"$(INTDIR)" :
    if not exist "$(INTDIR)/$(NULL)" mkdir "$(INTDIR)"

# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_CONSOLE" /YX /c
# ADD CPP /nologo /MT /W3 /GX /O2 /I "c:\mssql\dblib\include" /D "NDEBUG" /D "WIN32"
/D "_CONSOLE" /D "DBNTWIN32" /c
# SUBTRACT CPP /YX
CPP_PROJ=/nologo /MT /W3 /GX /O2 /I "c:\mssql\dblib\include" /D "NDEBUG" /D\
 "WIN32" /D "_CONSOLE" /D "DBNTWIN32" /Fo"$(INTDIR)/" /c
CPP_OBJS=.\objects/
CPP_SBRS=.\.
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
BSC32_FLAGS=/nologo /o"$(OUTDIR)/tpccldr.bsc"
BSC32_SBRS= \

LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:console /machine:I386
# ADD LINK32 c:\mssql\dblib\lib\ntwdblib.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib
odbc32.lib odbccp32.lib /nologo /subsystem:console /pdb:none /machine:I386
LINK32_FLAGS=c:\mssql\dblib\lib\ntwdblib.lib kernel32.lib user32.lib gdi32.lib\
 winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib\
 uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:console /pdb:none\
 /machine:I386 /out:"$(OUTDIR)/tpccldr.exe"
LINK32_OBJS= \
          "$(INTDIR)\getargs.obj" \
          "$(INTDIR)\random.obj" \
          "$(INTDIR)\strings.obj" \
          "$(INTDIR)\time.obj" \
          "$(INTDIR)\tpccldr.obj"

"$(OUTDIR)\tpccldr.exe" : "$(OUTDIR)" $(DEF_FILE) $(LINK32_OBJS)
    $(LINK32) @<<
  $(LINK32_FLAGS) $(LINK32_OBJS)
<<

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

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir "bin"
# PROP Intermediate_Dir "objects"
# PROP Target_Dir ""
OUTDIR=.\bin
INTDIR=.\objects

ALL : "$(OUTDIR)\tpccldr.exe"

CLEAN :
          -@erase "$(INTDIR)\getargs.obj"
          -@erase "$(INTDIR)\random.obj"
          -@erase "$(INTDIR)\strings.obj"
          -@erase "$(INTDIR)\time.obj"
          -@erase "$(INTDIR)\tpccldr.obj"
          -@erase "$(INTDIR)\vc40.idb"
          -@erase "$(INTDIR)\vc40.pdb"
          -@erase "$(OUTDIR)\tpccldr.exe"

"$(OUTDIR)" :
    if not exist "$(OUTDIR)/$(NULL)" mkdir "$(OUTDIR)"

"$(INTDIR)" :
    if not exist "$(INTDIR)/$(NULL)" mkdir "$(INTDIR)"

# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_CONSOLE" /YX
/c
# ADD CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /I "c:\mssql\dblib\include" /D "_DEBUG"
/D "WIN32" /D "_CONSOLE" /D "DBNTWIN32" /c
# SUBTRACT CPP /YX
CPP_PROJ=/nologo /MTd /W3 /Gm /GX /Zi /Od /I "c:\mssql\dblib\include" /D\
 "_DEBUG" /D "WIN32" /D "_CONSOLE" /D "DBNTWIN32" /Fo"$(INTDIR)/"\
 /Fd"$(INTDIR)/" /c
CPP_OBJS=.\objects/
CPP_SBRS=.\.
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
BSC32_FLAGS=/nologo /o"$(OUTDIR)/tpccldr.bsc"
BSC32_SBRS= \

LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:console /debug /machine:I386
# ADD LINK32 c:\mssql\dblib\lib\ntwdblib.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib
odbc32.lib odbccp32.lib /nologo /subsystem:console /pdb:none /debug /machine:I386
LINK32_FLAGS=c:\mssql\dblib\lib\ntwdblib.lib kernel32.lib user32.lib gdi32.lib\
 winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib\
 uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:console /pdb:none /debug\
 /machine:I386 /out:"$(OUTDIR)/tpccldr.exe"
LINK32_OBJS= \
          "$(INTDIR)\getargs.obj" \
          "$(INTDIR)\random.obj" \
          "$(INTDIR)\strings.obj" \
          "$(INTDIR)\time.obj" \
          "$(INTDIR)\tpccldr.obj"

"$(OUTDIR)\tpccldr.exe" : "$(OUTDIR)" $(DEF_FILE) $(LINK32_OBJS)
    $(LINK32) @<<
  $(LINK32_FLAGS) $(LINK32_OBJS)
<<

!ENDIF

.c{$(CPP_OBJS)}.obj:
   $(CPP) $(CPP_PROJ) $<

.cpp{$(CPP_OBJS)}.obj:
   $(CPP) $(CPP_PROJ) $<

.cxx{$(CPP_OBJS)}.obj:
   $(CPP) $(CPP_PROJ) $<

.c{$(CPP_SBRS)}.sbr:
   $(CPP) $(CPP_PROJ) $<

.cpp{$(CPP_SBRS)}.sbr:
   $(CPP) $(CPP_PROJ) $<

.cxx{$(CPP_SBRS)}.sbr:
   $(CPP) $(CPP_PROJ) $<

################################################################################
# Begin Target

# Name "tpccldr - Win32 Release"
# Name "tpccldr - Win32 Debug"

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

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

!ENDIF

################################################################################
# Begin Source File

SOURCE=.\src\random.c
DEP_CPP_RANDO=\
          ".\src\tpcc.h"\
          "\mssql\dblib\include\sqldb.h"\
          "\mssql\dblib\include\sqlfront.h"\


"$(INTDIR)\random.obj" : $(SOURCE) $(DEP_CPP_RANDO) "$(INTDIR)"
   $(CPP) $(CPP_PROJ) $(SOURCE)


# End Source File
################################################################################
# Begin Source File

SOURCE=.\src\strings.c
DEP_CPP_STRIN=\
          ".\src\tpcc.h"\
          "\mssql\dblib\include\sqldb.h"\
          "\mssql\dblib\include\sqlfront.h"\


"$(INTDIR)\strings.obj" : $(SOURCE) $(DEP_CPP_STRIN) "$(INTDIR)"
   $(CPP) $(CPP_PROJ) $(SOURCE)
```

```
# End Source File
################################################################################
# Begin Source File

SOURCE=.\src\time.c
DEP_CPP_TIME_=\
          ".\src\tpcc.h"\
          "\mssql\dblib\include\sqldb.h"\
          "\mssql\dblib\include\sqlfront.h"\


"$(INTDIR)\time.obj" : $(SOURCE) $(DEP_CPP_TIME_) "$(INTDIR)"
   $(CPP) $(CPP_PROJ) $(SOURCE)


# End Source File
################################################################################
# Begin Source File

SOURCE=.\src\tpccldr.c
DEP_CPP_TPCCL=\
          ".\src\tpcc.h"\
          "\mssql\dblib\include\sqldb.h"\
          "\mssql\dblib\include\sqlfront.h"\


"$(INTDIR)\tpccldr.obj" : $(SOURCE) $(DEP_CPP_TPCCL) "$(INTDIR)"
   $(CPP) $(CPP_PROJ) $(SOURCE)


# End Source File
################################################################################
# Begin Source File

SOURCE=.\src\getargs.c
DEP_CPP_GETAR=\
          ".\src\tpcc.h"\
          "\mssql\dblib\include\sqldb.h"\
          "\mssql\dblib\include\sqlfront.h"\


"$(INTDIR)\getargs.obj" : $(SOURCE) $(DEP_CPP_GETAR) "$(INTDIR)"
   $(CPP) $(CPP_PROJ) $(SOURCE)


# End Source File
# End Target
# End Project
################################################################################
```

TPCC.H

```
//        File:               TPCC.H
//                                      Microsoft TPC-C Kit Ver. 4.00
//                                      Copyright Microsoft, 1996, 1997, 1998
//        Purpose:  Header file for TPC-C database loader


// Build number of TPC Benchmark Kit
#define TPCKIT_VER  "4.00"

// 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 ORDERS_PER_DIST                  3000
#define LOADER_RES_FILE                   "logs\\load.out"
#define LOADER_NURAND_C                             123
#define DEF_STARTING_WAREHOUSE                    1
#define BUILD_INDEX                                         1         // build both
data and indexes
#define INDEX_ORDER                                         1         // build
indexes before load
#define SCALE_DOWN                                          0   // build a normal
scale database
#define INDEX_SCRIPT_PATH     "scripts"

typedef struct
{
    char                                          *server;
    char                                          *database;
    char                                          *user;
    char                                          *password;
          BOOL                                    tables_all;
          // set if loading all tables
          BOOL                                    table_item;
          // set if loading ITEM table specifically
          BOOL                                    table_warehouse;    // set if
loading WAREHOUSE, DISTRICT, and STOCK
          BOOL                                    table_customer;               //
set if loading CUSTOMER and HISTORY
```

```
          BOOL                                    table_orders;                 //
set if loading NEW-ORDER, ORDERS, ORDER-LINE
    long                                          num_warehouses;
    long                                          batch;
    long                                          verbose;
          long                                    pack_size;
          char                                    *loader_res_file;
          char                                    *synch_servername;
          long                                    case_sensitivity;
          long                                    starting_warehouse;
          long                                    build_index;
          long                                    index_order;
          long                                    scale_down;
          char                                    *index_script_path;
} TPCCLDR_ARGS;

// String length constants
          #define SERVER_NAME_LEN            20
          #define DATABASE_NAME_LEN          20
          #define USER_NAME_LEN              20
          #define PASSWORD_LEN               20
          #define TABLE_NAME_LEN             20
          #define I_DATA_LEN                 50
          #define I_NAME_LEN                 24
          #define BRAND_LEN                   1
          #define LAST_NAME_LEN              16
          #define W_NAME_LEN                 10
          #define ADDRESS_LEN                20
          #define STATE_LEN                   2
          #define ZIP_LEN                     9
          #define S_DIST_LEN                 24
          #define S_DATA_LEN                 50
          #define D_NAME_LEN                 10
          #define FIRST_NAME_LEN             16
          #define MIDDLE_NAME_LEN             2
          #define PHONE_LEN                  16
          #define CREDIT_LEN                  2
          #define C_DATA_LEN                500
          #define H_DATA_LEN                 24
          #define DIST_INFO_LEN              24
          #define MAX_OL_NEW_ORDER_ITEMS     15
          #define MAX_OL_ORDER_STATUS_ITEMS  15
          #define STATUS_LEN                 25
          #define OL_DIST_INFO_LEN                   24
          #define C_SINCE_LEN                                  23
          #define H_DATE_LEN                                   23
          #define OL_DELIVERY_D_LEN                  23
          #define O_ENTRY_D_LEN              23


// Functions in random.c
void      seed();
long      irand();
double    drand();
void      WUCreate();
short     WURand();
long      RandomNumber(long lower, long upper);

// Functions in getargs.c;
void      GetArgsLoader();
void      GetArgsLoaderUsage();

// Functions in time.c
long      TimeNow();

// Functions in strings.c
void      MakeAddress();
void      LastName();
int       MakeAlphaString();
int       MakeOriginalAlphaString();
int       MakeNumberString();
int       MakeZipNumberString();
void      InitString();
void      InitAddress();
void      PaddString();
```

TPCCLDR.C

```
//        File:               TPCCLDR.C
//                                      Microsoft TPC-C Kit Ver. 4.00
//                                      Copyright Microsoft, 1996, 1997, 1998
//        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);

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      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     i_hstmt1;
HSTMT     w_hstmt1;
HSTMT     c_hstmt1, c_hstmt2;
HSTMT     o_hstmt1, o_hstmt2, o_hstmt3;

ORDERS_STRUCT   orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT customer_buf[CUSTOMERS_PER_DISTRICT];
long            orders_rows_loaded;
long            new_order_rows_loaded;
long            order_line_rows_loaded;
long            history_rows_loaded;
long            customer_rows_loaded;
long            stock_rows_loaded;
long            district_rows_loaded;
long            item_rows_loaded;
long            warehouse_rows_loaded;
long            main_time_start;
long            main_time_end;
long                          max_items;
long                          customers_per_district;
long                          orders_per_district;
long                          first_new_order;
long                          last_new_order;


TPCCLDR_ARGS    *aptr, args;


//=======================================================================
//
// Function name: main
//
//=======================================================================

int main(int  argc, char **argv)
```

```
{
        DWORD           dwThreadID[MAX_MAIN_THREADS];
        HANDLE          hThread[MAX_MAIN_THREADS];
        FILE            *fLoader;
        char            buffer[255];
        int                                     i;

        for (i=0; i<MAX_MAIN_THREADS; i++)
                hThread[i] = NULL;

    printf("\n**************************************************");
        printf("\n*                                                *");
        printf("\n*  Microsoft SQL Server                          *");
        printf("\n*                                                *");
        printf("\n*  TPC-C BENCHMARK KIT:  Database loader         *");
        printf("\n*  Version %s                                   *", TPCKIT_VER);
        printf("\n*                                                *");
        printf("\n**************************************************\n\n");

    // process command line arguments

    aptr = &args;
    GetArgsLoader(argc, argv, aptr);

        printf("Build interface is ODBC.\n");

        if (aptr->build_index == 0)
                printf("Data load only - no index creation.\n");
        else
                printf("Data load and index creation.\n");

        if (aptr->index_order == 0)
                printf("Clustered indexes will be created after bulk load.\n");
        else
                printf("Clustered indexes will be created before bulk load.\n");

        // set database scale values
        if (aptr->scale_down == 1)
        {
                printf("*** Scaled Down Database ***\n");
                max_items = MAXITEMS_SCALE_DOWN;
                customers_per_district = CUSTOMERS_SCALE_DOWN;
                orders_per_district = ORDERS_SCALE_DOWN;
                first_new_order = 0;
                last_new_order  = 30;
        }
        else
        {
                max_items = MAXITEMS;
                customers_per_district = CUSTOMERS_PER_DISTRICT;
                orders_per_district = ORDERS_PER_DISTRICT;
                first_new_order = 2100;
                last_new_order  = 3000;
        }

        // open connections to SQL Server

        OpenConnections();

        // open file for loader results
    fLoader = fopen(aptr->loader_res_file, "w");
```

```
    if (fLoader == NULL)
        {
                printf("Error, loader result file open failed.");
                exit(-1);
        }

    // start loading data

    sprintf(buffer,"TPC-C load started for %ld warehouses.\n",aptr->num_warehouses);

        printf("%s",buffer);
        fprintf(fLoader,"%s",buffer);

        main_time_start = (TimeNow() / MILLI);

        // start parallel load threads


        if (aptr->tables_all || aptr->table_item)
        {
                fprintf(fLoader, "\nStarting loader threads for: item\n");

                hThread[0] = CreateThread(NULL,
                                                                                  0,

(LPTHREAD_START_ROUTINE) LoadItem,

NULL,
                                                                                  0,

&dwThreadID[0]);

                if (hThread[0] == NULL)
                {
                        printf("Error, failed in creating creating thread =
0.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_warehouse)
        {
                fprintf(fLoader, "Starting loader threads for: warehouse\n");

                hThread[1] = CreateThread(NULL,
                                                                                  0,

(LPTHREAD_START_ROUTINE) LoadWarehouse,

NULL,
                                                                                  0,

&dwThreadID[1]);

                if (hThread[1] == NULL)
                {
                        printf("Error, failed in creating creating thread =
1.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_customer)
```

```
          {
                    fprintf(fLoader, "Starting loader threads for: customer\n");

                    hThread[2] = CreateThread(NULL,
                                                                                  0,

(LPTHREAD_START_ROUTINE) LoadCustomer,

NULL,
                                                                                  0,

&dwThreadID[2]);

                    if (hThread[2] == NULL)
                    {
                              printf("Error, failed in creating creating main thread
= 2.\n");
                              exit(-1);
                    }
          }

          if (aptr->tables_all || aptr->table_orders)
          {
                    fprintf(fLoader, "Starting loader threads for: orders\n");

                    hThread[3] = CreateThread(NULL,
                                                                                  0,

(LPTHREAD_START_ROUTINE) LoadOrders,

NULL,
                                                                                  0,

&dwThreadID[3]);

                    if (hThread[3] == NULL)
                    {
                              printf("Error, failed in creating creating main thread
= 3.\n");
                              exit(-1);
                    }
          }

          // Wait for threads to finish...
          for (i=0; i<MAX_MAIN_THREADS; i++)
          {
                    if (hThread[i] != NULL)
                    {
                              WaitForSingleObject( hThread[i], INFINITE );
                              CloseHandle(hThread[i]);
                              hThread[i] = NULL;
                    }
          }

    main_time_end = (TimeNow() / MILLI);

    sprintf(buffer,"\nTPC-C load completed successfully in %ld minutes.\n",
                              (main_time_end - main_time_start)/60);

          printf("%s",buffer);
          fprintf(fLoader, "%s", buffer);

          fclose(fLoader);
```

```
          SQLFreeEnv(henv);

    exit(0);

          return 0;
}


//=======================================================================
//
// Function name: LoadItem
//
//=======================================================================

void LoadItem()
{
    long            i_id;
          long                i_im_id;
    char            i_name[I_NAME_LEN+1];
    double                    i_price;
    char            i_data[I_DATA_LEN+1];
          char                name[20];
          long                time_start;
          RETCODE             rc;
          DBINT               rcint;
          char                bcphint[128];

          // 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 != SUCCEED)
                    HandleErrorDBC(i_hdbc1);

          if ((aptr->build_index == 1) && (aptr->index_order == 1))
          {
                    sprintf(bcphint, "tablock, order (i_id), ROWS_PER_BATCH =
100000");
                    rc = bcp_control(i_hdbc1, BCPHINTS, (int) bcphint);
                    if (rc != SUCCEED)
                              HandleErrorDBC(i_hdbc1);
          }

          rc = bcp_bind(i_hdbc1, (BYTE *) &i_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
          if (rc != SUCCEED)
                    HandleErrorDBC(i_hdbc1);

          rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 2);
          if (rc != SUCCEED)
```

```
                    HandleErrorDBC(i_hdbc1);

          rc = bcp_bind(i_hdbc1, (BYTE *) i_name, 0, I_NAME_LEN, NULL, 0, 0, 3);
          if (rc != SUCCEED)
                    HandleErrorDBC(i_hdbc1);

          rc = bcp_bind(i_hdbc1, (BYTE *) &i_price, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 4);
          if (rc != SUCCEED)
                    HandleErrorDBC(i_hdbc1);

          rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0, I_DATA_LEN, NULL, 0, 0, 5);
          if (rc != SUCCEED)
                    HandleErrorDBC(i_hdbc1);

          time_start = (TimeNow() / MILLI);

          item_rows_loaded = 0;

          for (i_id = 1; i_id <= max_items; i_id++)
          {
                    i_im_id = RandomNumber(1L, 10000L);

                    MakeAlphaString(14, 24, I_NAME_LEN, i_name);

                    i_price = ((float) RandomNumber(100L, 10000L))/100.0;

                    MakeOriginalAlphaString(26, 50, I_DATA_LEN, i_data, 10);

                    rc = bcp_sendrow(i_hdbc1);
                    if (rc != SUCCEED)
                              HandleErrorDBC(i_hdbc1);

                    item_rows_loaded++;
                    CheckForCommit(i_hdbc1, i_hstmt1, item_rows_loaded, "item",
&time_start);
          }

          rcint = bcp_done(i_hdbc1);
          if (rcint < 0)
                    HandleErrorDBC(i_hdbc1);

          printf("Finished loading item table.\n");

          SQLFreeStmt(i_hstmt1, SQL_DROP);
          SQLDisconnect(i_hdbc1);
          SQLFreeConnect(i_hdbc1);

          // if build index after load
          if ((aptr->build_index == 1) && (aptr->index_order == 0))
                    BuildIndex("idxitmcl");

}


//=======================================================================
//
// Function   : LoadWarehouse
//
// Loads WAREHOUSE table and loads Stock and District as Warehouses are created
//
//=======================================================================
```

```
void LoadWarehouse()
{
    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      bcphint[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("idxwarcl");

          InitString(w_name, W_NAME_LEN+1);
          InitAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

          sprintf(name, "%s..%s", aptr->database, "warehouse");

          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(bcphint, "tablock, order (w_id), ROWS_PER_BATCH = %d",
aptr->num_warehouses);
                    rc = bcp_control(w_hdbc1, BCPHINTS, (int) bcphint);
                    if (rc != SUCCEED)
                              HandleErrorDBC(w_hdbc1);
          }

          rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 1);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0, W_NAME_LEN, NULL, 0, 0, 2);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          rc = bcp_bind(w_hdbc1, (BYTE *) w_street_1, 0, ADDRESS_LEN, NULL, 0, 0,
3);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          rc = bcp_bind(w_hdbc1, (BYTE *) w_street_2, 0, ADDRESS_LEN, NULL, 0, 0,
4);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) w_city, 0, ADDRESS_LEN, NULL, 0, 0, 5);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          rc = bcp_bind(w_hdbc1, (BYTE *) w_state, 0, STATE_LEN, NULL, 0, 0, 6);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          rc = bcp_bind(w_hdbc1, (BYTE *) w_zip, 0, ZIP_LEN, NULL, 0, 0, 7);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 8);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 9);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          time_start = (TimeNow() / MILLI);

          warehouse_rows_loaded = 0;

          for (w_id = (short)aptr->starting_warehouse; w_id <= aptr->num_warehouses;
w_id++)
          {
                    MakeAlphaString(6,10, W_NAME_LEN, w_name);

                    MakeAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

                    w_tax = ((float) RandomNumber(0L,2000L))/10000.00;

                    w_ytd = 300000.00;

                    rc = bcp_sendrow(w_hdbc1);
                    if (rc != SUCCEED)
                              HandleErrorDBC(w_hdbc1);

                    warehouse_rows_loaded++;
                    CheckForCommit(w_hdbc1, i_hstmt1, warehouse_rows_loaded,
"warehouse", &time_start);
          }

          rcint = bcp_done(w_hdbc1);
          if (rcint < 0)
                    HandleErrorDBC(w_hdbc1);

          printf("Finished loading warehouse table.\n");

          // if build index after load...
          if ((aptr->build_index == 1) && (aptr->index_order == 0))
                    BuildIndex("idxwarcl");

          stock_rows_loaded = 0;
          district_rows_loaded = 0;

          District();
          Stock();
```

```
}


//======================================================================
//
// Function   : District
//
//======================================================================

void District()
{
    short d_id;
    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, (int) 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 != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          rc = bcp_bind(w_hdbc1, (BYTE *) d_zip, 0, ZIP_LEN, NULL, 0, 0, 8);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          rc = bcp_bind(w_hdbc1, (BYTE *) &d_tax, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 9);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          rc = bcp_bind(w_hdbc1, (BYTE *) &d_ytd, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 10);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          rc = bcp_bind(w_hdbc1, (BYTE *) &d_next_o_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 11);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          d_ytd  = 30000.0;

          d_next_o_id = orders_per_district+1;

          time_start = (TimeNow() / MILLI);

          for (w_id = aptr->starting_warehouse; w_id <= aptr->num_warehouses;
w_id++)
          {
                    d_w_id = w_id;

                    for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
                    {
                              MakeAlphaString(6,10,D_NAME_LEN, d_name);

                              MakeAddress(d_street_1, d_street_2, d_city, d_state,
d_zip);

                              d_tax = ((float) RandomNumber(0L,2000L))/10000.00;
```

```
                              rc = bcp_sendrow(w_hdbc1);
                              if (rc != SUCCEED)
                                        HandleErrorDBC(w_hdbc1);

                              district_rows_loaded++;
                              CheckForCommit(w_hdbc1, w_hstmt1,
district_rows_loaded, "district", &time_start);
                    }
    }

          rcint = bcp_done(w_hdbc1);
          if (rcint < 0)
                    HandleErrorDBC(w_hdbc1);

          printf("Finished loading district table.\n");

          // if build index after load...
          if ((aptr->build_index == 1) && (aptr->index_order == 0))
                    BuildIndex("idxdiscl");

    return;
}


//=======================================================================
//
// Function   : Stock
//
//=======================================================================

void Stock()
{
    long  s_i_id;
    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 != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (s_i_id, s_w_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 100000));
                rc = bcp_control(w_hdbc1, BCPHINTS, (int) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);
        }

        rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &s_quantity, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 3);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01, 0, S_DIST_LEN, NULL, 0, 0, 4);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02, 0, S_DIST_LEN, NULL, 0, 0, 5);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03, 0, S_DIST_LEN, NULL, 0, 0, 6);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04, 0, S_DIST_LEN, NULL, 0, 0, 7);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05, 0, S_DIST_LEN, NULL, 0, 0, 8);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06, 0, S_DIST_LEN, NULL, 0, 0, 9);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07, 0, S_DIST_LEN, NULL, 0, 0, 10);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0, S_DIST_LEN, NULL, 0, 0, 11);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09, 0, S_DIST_LEN, NULL, 0, 0, 12);
```

```
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0, S_DIST_LEN, NULL, 0, 0, 13);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 14);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &s_order_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 15);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) &s_remote_cnt, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 16);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0, S_DATA_LEN, NULL, 0, 0, 17);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        s_ytd = s_order_cnt = s_remote_cnt = 0;

        time_start = (TimeNow() / MILLI);

        printf("...Loading stock table\n");

        for (s_i_id=1; s_i_id <= max_items; s_i_id++)
        {

                for (s_w_id = (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];

          // 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 != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

          if ((aptr->build_index == 1) && (aptr->index_order == 1))
          {
                    sprintf(bcphint, "tablock, order (c_w_id, c_d_id, c_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
                    rc = bcp_control(c_hdbc1, BCPHINTS, (int) bcphint);
                    if (rc != SUCCEED)
                              HandleErrorDBC(c_hdbc1);
          }
```

```
          sprintf(name, "%s..%s", aptr->database, "history");

          rc = bcp_init(c_hdbc2, name, NULL, "logs\\history.err", DB_IN);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc2);

          sprintf(bcphint, "tablock");
          rc = bcp_control(c_hdbc2, BCPHINTS, (int) bcphint);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc2);

          customer_rows_loaded    = 0;
          history_rows_loaded     = 0;

          CustomerBufInit();

          customer_time_start.time_start = (TimeNow() / MILLI);
          history_time_start.time_start = (TimeNow() / MILLI);

          for (w_id = (short)aptr->starting_warehouse; w_id <= aptr->num_warehouses;
w_id++)
          {
                    for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
                    {

                              CustomerBufLoad(d_id, w_id);

                              // Start parallel loading threads here...

                              // Start customer table thread

                              printf("...Loading customer table for: d_id = %d, w_id
= %d\n", d_id, w_id);

                              hThread[0] = CreateThread(NULL,

            0,

            (LPTHREAD_START_ROUTINE) LoadCustomerTable,

            &customer_time_start,

            0,

            &dwThreadID[0]);

                              if (hThread[0] == NULL)
                              {
                                        printf("Error, failed in creating creating
thread = 0.\n");
                                        exit(-1);
                              }

                              // Start History table thread

                              printf("...Loading history table for: d_id = %d, w_id
= %d\n", d_id, w_id);

                              hThread[1] = CreateThread(NULL,

            0,
            (LPTHREAD_START_ROUTINE) LoadHistoryTable,

            &history_time_start,

            0,

            &dwThreadID[1]);

                              if (hThread[1] == NULL)
                              {
                                        printf("Error, failed in creating creating
thread = 1.\n");
                                        exit(-1);
                              }

                              WaitForSingleObject( hThread[0], INFINITE );
                              WaitForSingleObject( hThread[1], INFINITE );

                              if (CloseHandle(hThread[0]) == FALSE)
                              {
                                        printf("Error, failed in closing customer
thread handle with errno: %d\n", GetLastError());
                              }

                              if (CloseHandle(hThread[1]) == FALSE)
                              {
                                        printf("Error, failed in closing history
thread handle with errno: %d\n", GetLastError());
                              }


                    }

          }

          // flush the bulk connection
          rcint = bcp_done(c_hdbc1);
          if (rcint < 0)
                    HandleErrorDBC(c_hdbc1);

          rcint = bcp_done(c_hdbc2);
          if (rcint < 0)
                    HandleErrorDBC(c_hdbc2);

          printf("Finished loading customer table.\n");

          // if build index after load...
          if ((aptr->build_index == 1) && (aptr->index_order == 0))
                    BuildIndex("idxcuscl");

          // build non-clustered index
          if (aptr->build_index == 1)
                    BuildIndex("idxcusnc");

          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(500, 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 != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

          rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 3);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0, FIRST_NAME_LEN, NULL, 0, 0, 4);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0, MIDDLE_NAME_LEN,NULL, 0, 0, 5);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN, NULL, 0, 0, 6);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN, NULL, 0, 0, 7);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN, NULL,0,0, 8);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL, 0, 0, 9);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0, 0, 10);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0, 11);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL, 0, 0, 12);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

          rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0, C_SINCE_LEN, NULL, 0,
SQLCHARACTER, 13);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL, 0, 0, 14);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 15);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 16);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);


          //  fix to avoid ODBC float to numeric conversion problem.

    //  rc = bcp_bind(c_hdbc1, (BYTE *) &c_balance, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 17);
          //  if (rc != SUCCEED)
          //        HandleErrorDBC(c_hdbc1);
```

```
    rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0, SQLCHARACTER, 17);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);


    rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, 18);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 19);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt,0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 20);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0, 500, NULL, 0, 0, 21);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

    for (i = 0; i < customers_per_district; i++)
    {
                    c_id = customer_buf[i].c_id;
                    c_d_id = customer_buf[i].c_d_id;
                    c_w_id = customer_buf[i].c_w_id;

                    strcpy(c_first, customer_buf[i].c_first);
                    strcpy(c_middle, customer_buf[i].c_middle);
                    strcpy(c_last, customer_buf[i].c_last);
                    strcpy(c_street_1, customer_buf[i].c_street_1);
                    strcpy(c_street_2, customer_buf[i].c_street_2);
                    strcpy(c_city, customer_buf[i].c_city);
                    strcpy(c_state, customer_buf[i].c_state);
                    strcpy(c_zip, customer_buf[i].c_zip);
                    strcpy(c_phone, customer_buf[i].c_phone);
                    strcpy(c_credit, customer_buf[i].c_credit);

                    FormatDate(&c_since);


                    c_credit_lim = customer_buf[i].c_credit_lim;
                    c_discount = customer_buf[i].c_discount;

                    //  fix to avoid ODBC float to numeric conversion problem.

                    //  c_balance = customer_buf[i].c_balance;
                    strcpy(c_balance, customer_buf[i].c_balance);

                    c_ytd_payment = customer_buf[i].c_ytd_payment;
                    c_payment_cnt = customer_buf[i].c_payment_cnt;
                    c_delivery_cnt = customer_buf[i].c_delivery_cnt;

                    strcpy(c_data, customer_buf[i].c_data);

                    // Send data to server
                    rc = bcp_sendrow(c_hdbc1);
                    if (rc != SUCCEED)
                              HandleErrorDBC(c_hdbc1);
                    customer_rows_loaded++;
                    CheckForCommit(c_hdbc1, c_hstmt1, customer_rows_loaded,
"customer", &customer_time_start->time_start);
    }

}


//=======================================================================
//
// Function   : LoadHistoryTable
//
//=======================================================================

void LoadHistoryTable(LOADER_TIME_STRUCT *history_time_start)
{
          int         i;
    long        c_id;
    short       c_d_id;
    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("idxordcl");
                              BuildIndex("idxnodcl");
                              BuildIndex("idxodlcl");
          }

          // initialize bulk copy
          sprintf(name, "%s..%s", aptr->database, "orders");

          rc = bcp_init(o_hdbc1, name, NULL, "logs\\orders.err", DB_IN);
          if (rc != 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, (int) 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, (int) 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, (int) 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_DIST+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_DIST);

          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 12:00:00.000");

                              }
                    }
          }
}


//=======================================================================
//
// 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("idxordcl");

                    // build non-clustered index
                    if (aptr->build_index == 1)
                              BuildIndex("idxordnc");
          }

}


//=========================================================================
//
// Function   : LoadNewOrderTable
//
//=========================================================================

void LoadNewOrderTable(LOADER_TIME_STRUCT *new_order_time_start)
{
    int         i;
    long        o_id;
    short       o_d_id;
    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("idxnodcl");

          }

}


//=========================================================================
//
// 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++)
    {
                    o_id    = orders_buf[i].o_id;
                    o_d_id  = orders_buf[i].o_d_id;
                    o_w_id  = orders_buf[i].o_w_id;

                    for (j=0; j < orders_buf[i].o_ol_cnt; j++)
                    {
                              ol             = orders_buf[i].o_ol[j].ol;
                              ol_i_id        = orders_buf[i].o_ol[j].ol_i_id;
                              ol_supply_w_id = orders_buf[i].o_ol[j].ol_supply_w_id;
                              ol_quantity    = orders_buf[i].o_ol[j].ol_quantity;
                              ol_amount      = orders_buf[i].o_ol[j].ol_amount;


          strcpy(ol_delivery_d,orders_buf[i].o_ol[j].ol_delivery_d);


          strcpy(ol_dist_info,orders_buf[i].o_ol[j].ol_dist_info);

                              rc = bcp_sendrow(o_hdbc3);
                              if (rc != SUCCEED)
                                        HandleErrorDBC(o_hdbc3);

                              order_line_rows_loaded++;
                              CheckForCommit(o_hdbc3, o_hstmt3,
order_line_rows_loaded, "order_line", &order_line_time_start->time_start);
                    }

          }

          // rcint = bcp_batch(o_hdbc3);
          // if (rcint < 0)
          //        HandleErrorDBC(o_hdbc3);

          if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
          {
```

```
                    rcint = bcp_done(o_hdbc3);
                    if (rcint < 0)
                              HandleErrorDBC(o_hdbc3);

                    SQLFreeStmt(o_hstmt3, SQL_DROP);
                    SQLDisconnect(o_hdbc3);
                    SQLFreeConnect(o_hdbc3);

                    // if build index after load...
                    if ((aptr->build_index == 1) && (aptr->index_order == 0))
                              BuildIndex("idxodlcl");

          }

}


//======================================================================
//
// Function   : GetPermutation
//
//======================================================================

void GetPermutation(int perm[], int n)
{
          int i, r, t;

          for (i=1;i<=n;i++)
                    perm[i] = i;

          for (i=1;i<=n;i++)
          {
                    r = RandomNumber(i,n);
                    t = perm[i];
                    perm[i] = perm[r];
                    perm[r] = t;
          }

}


//======================================================================
//
// Function   : CheckForCommit
//
//======================================================================

void CheckForCommit(HDBC hdbc,
                                                  HSTMT hstmt,
                                        int rows_loaded,
                                                  char *table_name,
                                        long *time_start)
{

    long  time_end, time_diff;
          // DBINT  rcint;

    if ( !(rows_loaded % aptr->batch) )
    {

                    // rcint = bcp_batch(hdbc);
                    // if (rcint < 0)
                    //        HandleErrorDBC(hdbc);
                    time_end = (TimeNow() / MILLI);
                    time_diff = time_end - *time_start;

                    printf("-> Loaded %ld rows into %s in %ld sec - Total = %d (%.2f
rps)\n",
                              aptr->batch,
                              table_name,
                                        time_diff,
                              rows_loaded,
                                        (float) aptr->batch / (time_diff ? time_diff
: 1L));

                    *time_start = time_end;
    }

    return;
}


//=======================================================================
//
// Function   : OpenConnections
//
//=======================================================================

void OpenConnections()
{

          RETCODE         rc;

          char                          szDriverString[300];
          char                          szDriverStringOut[1024];
          SQLSMALLINT                   cbDriverStringOut;


          SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv );

          SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0 );

          SQLAllocHandle(SQL_HANDLE_DBC, henv , &i_hdbc1);
          SQLAllocHandle(SQL_HANDLE_DBC, henv , &w_hdbc1);
          SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc1);
          SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc2);
          SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc1);
          SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc2);
          SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc3);

          SQLSetConnectAttr(i_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
          SQLSetConnectAttr(w_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
          SQLSetConnectAttr(c_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
          SQLSetConnectAttr(c_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
          SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
          SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
          SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
```

```
          // Open connections to SQL Server

          // Connection 1

          sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                              aptr->server,
                                                              aptr->user,
                                                              aptr->password,
                                                              aptr->database );

          rc = SQLSetConnectOption (i_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
          if (rc != SUCCEED)
                    HandleErrorDBC(i_hdbc1);

          rc = SQLDriverConnect ( i_hdbc1,
                                                      NULL,

          (SQLCHAR*)&szDriverString[0] ,
                                                            SQL_NTS,

          (SQLCHAR*)&szDriverStringOut[0],

          sizeof(szDriverStringOut),
                                                            &cbDriverStringOut,
                                                            SQL_DRIVER_NOPROMPT );
          if (rc != SUCCEED)
                    HandleErrorDBC(i_hdbc1);

          // Connection 2

          sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                              aptr->server,
                                                              aptr->user,
                                                              aptr->password,
                                                              aptr->database );

          rc = SQLSetConnectOption (w_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          rc = SQLDriverConnect ( w_hdbc1,
                                                                      NULL,

          (SQLCHAR*)&szDriverString[0] ,
                                                                      SQL_NTS,

          (SQLCHAR*)&szDriverStringOut[0],

          sizeof(szDriverStringOut),

          &cbDriverStringOut,

          SQL_DRIVER_NOPROMPT );
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          // Connection 3

          sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                              aptr->server,
                                                               aptr->user,
                                                               aptr->password,
                                                               aptr->database );

          rc = SQLSetConnectOption (c_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

          rc = SQLDriverConnect ( c_hdbc1,
                                                                       NULL,

          (SQLCHAR*)&szDriverString[0] ,
                                                                       SQL_NTS,

          (SQLCHAR*)&szDriverStringOut[0],

          sizeof(szDriverStringOut),

          &cbDriverStringOut,

          SQL_DRIVER_NOPROMPT );
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

          // Connection 4

          sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                               aptr->server,
                                                               aptr->user,
                                                               aptr->password,
                                                               aptr->database );

          rc = SQLSetConnectOption (c_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc2);

          rc = SQLDriverConnect ( c_hdbc2,
                                                                       NULL,

          (SQLCHAR*)&szDriverString[0] ,
                                                                       SQL_NTS,

          (SQLCHAR*)&szDriverStringOut[0],

          sizeof(szDriverStringOut),

          &cbDriverStringOut,

          SQL_DRIVER_NOPROMPT );
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc2);

          // Connection 5

          sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                               aptr->server,
                                                               aptr->user,
                                                               aptr->password,
                                                               aptr->database );

          rc = SQLSetConnectOption (o_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
```

```
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc1);

          rc = SQLDriverConnect ( o_hdbc1,
                                                                       NULL,

          (SQLCHAR*)&szDriverString[0] ,
                                                                       SQL_NTS,

          (SQLCHAR*)&szDriverStringOut[0],

          sizeof(szDriverStringOut),

          &cbDriverStringOut,

          SQL_DRIVER_NOPROMPT );
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc1);

          // Connection 6

          sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                               aptr->server,
                                                               aptr->user,
                                                               aptr->password,
                                                               aptr->database );

          rc = SQLSetConnectOption (o_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc2);

          rc = SQLDriverConnect ( o_hdbc2,
                                                                       NULL,

          (SQLCHAR*)&szDriverString[0] ,
                                                                       SQL_NTS,

          (SQLCHAR*)&szDriverStringOut[0],

          sizeof(szDriverStringOut),

          &cbDriverStringOut,

          SQL_DRIVER_NOPROMPT );
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc2);

          // Connection 7

          sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                               aptr->server,
                                                               aptr->user,
                                                               aptr->password,
                                                               aptr->database );

          rc = SQLSetConnectOption (o_hdbc3, SQL_PACKET_SIZE, aptr->pack_size);
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc3);

          rc = SQLDriverConnect ( o_hdbc3,
                                                                       NULL,
          (SQLCHAR*)&szDriverString[0] ,
                                                                 SQL_NTS,

          (SQLCHAR*)&szDriverStringOut[0],

          sizeof(szDriverStringOut),

          &cbDriverStringOut,

          SQL_DRIVER_NOPROMPT );
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc3);

}


//=======================================================================
//
// Function name: BuildIndex
//
//=======================================================================

void BuildIndex(char          *index_script)
{
          char      cmd[256];

          printf("Starting index creation:  %s\n",index_script);

          sprintf(cmd, "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 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;

}
```

TIME.C

```
//        File:               TIME.C
//                                      Microsoft TPC-C Kit Ver. 4.00
//                                      Copyright Microsoft, 1996, 1997, 1998
//        Purpose:  Source file for time functions


// Includes
#include "tpcc.h"

// Globals
static long start_sec;


//=======================================================================
//
// Function name: TimeNow
//
//=======================================================================
long TimeNow()
{
   long             time_now;
   struct _timeb el_time;

#ifdef DEBUG
    printf("[%ld]DBG: Entering TimeNow()\n", (int) GetCurrentThreadId());
#endif

   _ftime(&el_time);

   time_now = ((el_time.time - start_sec) * 1000) + el_time.millitm;

   return time_now;
}
```

STRINGS.C

```
//        File:               STRINGS.C
//                                      Microsoft TPC-C Kit Ver. 4.00
//                                      Copyright Microsoft, 1996, 1997, 1998
//        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;
          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++)
                    str[i] = chArray[RandomNumber(0, chArrayMax)];
          if ( len < z )
                    memset(str+len, ’ ’, z - len);
          str[len] = 0;

    return len;
}


//=======================================================================
//
// Function name: MakeOriginalAlphaString
//
//=======================================================================

int MakeOriginalAlphaString(int  x,
                                                                int  y,
                                                                      int  z,
                                                                char *str,
                                                            int  percent)
{
    int             len;
    int             val;
    int             start;

#ifdef DEBUG
    printf("[%ld]DBG: Entering MakeOriginalAlphaString()\n", (int)
GetCurrentThreadId());
#endif

    // verify prercentage is valid
    if ((percent < 0) || (percent > 100))
    {
                    printf("MakeOrigianlAlphaString: Invalid percentage: %d\n",
percent);
                    exit(-1);
    }

    // verify string is at least 8 chars in length
    if ((x + y) <= 8)
    {
                    printf("MakeOriginalAlphaString: string length must be >= 8\n");
                    exit(-1);
    }

    // Make Alpha String
    len = MakeAlphaString(x,y, z, str);

    val = RandomNumber(1,100);
    if (val <= percent)
    {
                    start = RandomNumber(0, len - 8);
                    strncpy(str + start, "ORIGINAL", 8);
    }


#ifdef DEBUG
    printf("[%ld]DBG: MakeOriginalAlphaString: : %s\n",
                              (int) GetCurrentThreadId(), str);
#endif

    return strlen(str);
}


//=======================================================================
//
// Function name: MakeNumberString
//
//=======================================================================
int MakeNumberString(int  x, int  y, int  z, char *str)
{
          char tmp[16];

          //MakeNumberString is always called MakeZipNumberString(16, 16, 16,
string)

          memset(str, ’0’, 16);
          itoa(RandomNumber(0, 99999999), tmp, 10);
          memcpy(str, tmp, strlen(tmp));

          itoa(RandomNumber(0, 99999999), tmp, 10);
          memcpy(str+8, tmp, strlen(tmp));

          str[16] = 0;

    return 16;
}


//=======================================================================
//
// Function name: MakeZipNumberString
//
//=======================================================================
int MakeZipNumberString(int  x, int  y, int  z, char *str)
{
          char tmp[16];
          //MakeZipNumberString is always called MakeZipNumberString(9, 9, 9,
string)

          strcpy(str, "000011111");

          itoa(RandomNumber(0, 9999), tmp, 10);
          memcpy(str, tmp, strlen(tmp));

    return 9;
}


//=======================================================================
//
// Function name: InitString
//
//=======================================================================
void InitString(char *str, int len)
{
#ifdef DEBUG
    printf("[%ld]DBG: Entering InitString()\n", (int) GetCurrentThreadId());
#endif

          memset(str, ’ ’, len);
    str[len] = 0;
}


//=======================================================================
// Function name: InitAddress
//
// Description:
//
//=======================================================================

void InitAddress(char *street_1, char *street_2, char *city, char *state, char *zip)
{
          memset(street_1, ’ ’, ADDRESS_LEN+1);
          memset(street_2, ’ ’, ADDRESS_LEN+1);
          memset(city, ’ ’, ADDRESS_LEN+1);

    street_1[ADDRESS_LEN+1] = 0;
    street_2[ADDRESS_LEN+1] = 0;
    city[ADDRESS_LEN+1]     = 0;

          memset(state, ’ ’, STATE_LEN+1);
    state[STATE_LEN+1] = 0;

          memset(zip, ’ ’, ZIP_LEN+1);
    zip[ZIP_LEN+1] = 0;
}


//=======================================================================
//
// Function name: PaddString
//
//=======================================================================

void PaddString(int max, char *name)
{
          int                 len;

          len = strlen(name);
          if ( len < max )
                    memset(name+len, ’ ’, max - len);
          name[max] = 0;

          return;
}
```

RANDOM.C

```
//        File:               RANDOM.C
//                                      Microsoft TPC-C Kit Ver. 4.00
//                                      Copyright Microsoft, 1996, 1997, 1998
//        Purpose:  Random number generation routines for database loader


// Includes
#include "tpcc.h"
#include "math.h"

// Defines
#define A            16807
#define M       2147483647
#define Q           127773      /* M div A */
#define R             2836      /* M mod A */
#define Thread      __declspec(thread)

// Globals
long      Thread Seed = 0;       /* thread local seed */

/******************************************************************************
*                                                                             *
* random -                                                                    *
*       Implements a GOOD pseudo random number generator.  This generator     *
*       will/should? run the complete period before repeating.                *
*                                                                             *
* Copied from:                                                                *
*       Random Numbers Generators: Good Ones Are Hard to Find.                *
*       Communications of the ACM - October 1988 Volume 31 Number 10          *
*                                                                             *
* Machine Dependencies:                                                       *
*       long must be 2 ^ 31 - 1 or greater.                                    *
*                                                                             *
******************************************************************************/

/******************************************************************************
* seed - load the Seed value used in irand and drand.  Should be used before  *
*       first call to irand or drand.                                         *
******************************************************************************/

void seed(long val)
{

#ifdef DEBUG
    printf("[%ld]DBG: Entering seed()...\n", (int) GetCurrentThreadId());
          printf("Old Seed %ld New Seed %ld\n",Seed, val);
#endif
```

```
    if ( val < 0 )
              val = abs(val);

    Seed = val;

}


/*****************************************************************************
*                                                                            *
* irand - returns a 32 bit integer pseudo random number with a period of     *
*       1 to 2 ^ 32 - 1.                                                     *
*                                                                            *
* parameters:                                                                *
*       none.                                                                *
*                                                                            *
* returns:                                                                   *
*       32 bit integer - defined as long ( see above ).                       *
*                                                                            *
* side effects:                                                              *
*       seed get recomputed.                                                 *
*****************************************************************************/

long irand()
{
    register long    s;      /* copy of seed */
    register long    test;   /* test flag */
    register long    hi;     /* tmp value for speed */
    register long    lo;     /* tmp value for speed */

#ifdef DEBUG
    printf("[%ld]DBG: Entering irand()...\n", (int) GetCurrentThreadId());
#endif

    s = Seed;
    hi = s / Q;
    lo = s % Q;

    test = A * lo - R * hi;
    if ( test > 0 )
                    Seed = test;
    else
                    Seed = test + M;

    return( Seed );
}


/*****************************************************************************
*                                                                            *
* drand - returns a double pseudo random number between 0.0 and 1.0.         *
*       See irand.                                                           *
*****************************************************************************/
double drand()
{

#ifdef DEBUG
    printf("[%ld]DBG: Entering drand()...\n", (int) GetCurrentThreadId());
#endif

    return( (double)irand() / 2147483647.0);
}
```

```
//=======================================================================
// Function    : RandomNumber
//
// Description:
//=======================================================================
long RandomNumber(long lower, long upper)
{
    long rand_num;

#ifdef DEBUG
    printf("[%ld]DBG: Entering RandomNumber()...\n", (int) GetCurrentThreadId());
#endif

          if ( upper == lower )          /* pgd 08-13-96 perf enhancement */
                    return lower;

          upper++;

    if ( upper <= lower )
                    rand_num = upper;
    else
                    rand_num = lower + irand() % (upper - lower); /* pgd 08-13-96
perf enhancement */


#ifdef DEBUG
    printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                                        (int) GetCurrentThreadId(), lower, upper,
rand_num);
#endif


    return rand_num;
}


#if 0

//Orginal code pgd 08/13/96

long RandomNumber(long lower,
                                          long upper)
{
    long rand_num;

#ifdef DEBUG
    printf("[%ld]DBG: Entering RandomNumber()...\n", (int) GetCurrentThreadId());
#endif

          upper++;

    if ((upper <= lower))
                    rand_num = upper;
    else
                    rand_num = lower + irand() % ((upper > lower) ? upper - lower :
upper);


#ifdef DEBUG
    printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                                        (int) GetCurrentThreadId(), lower, upper,
rand_num);
#endif


    return rand_num;
}
#endif

//=======================================================================
// Function   : NURand
//
// Description:
//=======================================================================
long NURand(int iConst,
              long x,
              long y,
              long C)
{
    long rand_num;

#ifdef DEBUG
    printf("[%ld]DBG: Entering NURand()...\n", (int) GetCurrentThreadId());
#endif

    rand_num = (((RandomNumber(0,iConst) | RandomNumber(x,y)) + C) % (y-x+1))+x;

#ifdef DEBUG
    printf("[%ld]DBG: NURand: num = %d\n", (int) GetCurrentThreadId(), rand_num);
#endif

    return rand_num;
}
```

GETARGS.C

```
//        File:               GETARGS.C
//                                      Microsoft TPC-C Kit Ver. 4.00
//                                      Copyright Microsoft, 1996, 1997, 1998
//        Purpose:  Source file for command line processing


// Includes
#include "tpcc.h"

//=======================================================================
//
// Function name: GetArgsLoader
//
//=======================================================================

void GetArgsLoader(int argc, char **argv, TPCCLDR_ARGS *pargs)
{
    int             i;
    char  *ptr;

#ifdef DEBUG
    printf("[%ld]DBG: Entering GetArgsLoader()\n", (int) GetCurrentThreadId());
#endif

    /* init args struct with some useful values */
pargs->server                      =  SERVER;
pargs->user                        =  USER;
pargs->password                    =  PASSWORD;
pargs->database                    =  DATABASE;
pargs->batch                       =  BATCH;
pargs->num_warehouses              =  UNDEF;
      pargs->tables_all                      =  TRUE;
      pargs->table_item                      =  FALSE;
      pargs->table_warehouse                 =  FALSE;
      pargs->table_customer                  =  FALSE;
      pargs->table_orders                    =  FALSE;
      pargs->loader_res_file                 =  LOADER_RES_FILE;
      pargs->pack_size                       =  DEFLDPACKSIZE;
      pargs->starting_warehouse    =  DEF_STARTING_WAREHOUSE;
      pargs->build_index                     =  BUILD_INDEX;
      pargs->index_order                     =  INDEX_ORDER;
      pargs->index_script_path     =  INDEX_SCRIPT_PATH;
      pargs->scale_down                      =  SCALE_DOWN;

/* check for zero command line args */
if ( argc == 1 )
                GetArgsLoaderUsage();

for (i = 1; i < argc; ++i)
{
                if (argv[i][0] != '-' && argv[i][0] != '/')
                {
                printf("\nUnrecognized command");
                GetArgsLoaderUsage();
                exit(1);
                }

                ptr = argv[i];

                switch (ptr[1])
                {
                case 'h':     /* Fall throught */
                case 'H':
                                    GetArgsLoaderUsage();
                                    break;

                case 'D':
                                    pargs->database = ptr+2;
                                    break;

                case 'P':
                                    pargs->password = ptr+2;
                                    break;

                case 'S':
                                    pargs->server = ptr+2;
                                    break;

                case 'U':
                                    pargs->user = ptr+2;
                                    break;

                case 'b':
                                    pargs->batch = atol(ptr+2);
                                    break;

                case 'W':
                                    pargs->num_warehouses = atol(ptr+2);
```

```
                                        break;

                    case ’s’:           pargs->starting_warehouse = atol(ptr+2);
                                        break;

                    case ’t’:           {
                                                  pargs->tables_all = FALSE;
                                                  if (strcmp(ptr+2,"item") == 0)
                                                            pargs->table_item =
TRUE;
                                                  else if (strcmp(ptr+2,"warehouse")
== 0)
                                                            pargs->table_warehouse =
TRUE;
                                                  else if (strcmp(ptr+2,"customer")
== 0)
                                                            pargs->table_customer =
TRUE;
                                                  else if (strcmp(ptr+2,"orders") ==
0)
                                                            pargs->table_orders =
TRUE;
                                                  else
                                                  {
                                                  printf("\nUnrecognized command");
                                                  GetArgsLoaderUsage();
                                                  exit(1);
                                                  }

                                                  break;
                                        }

                              case ’f’:
                                        pargs->loader_res_file = ptr+2;
                                        break;

                              case ’p’:
                                        pargs->pack_size = atol(ptr+2);
                                        break;

                              case ’i’:
                                        pargs->build_index = atol(ptr+2);
                                        break;

                              case ’o’:
                                        pargs->index_order = atol(ptr+2);
                                        break;

                              case ’c’:
                                        pargs->scale_down = atol(ptr+2);
                                        break;

                    case ’d’:
                                        pargs->index_script_path = ptr+2;
                                        break;

                    default:
                              GetArgsLoaderUsage();
                              exit(-1);
                              break;
                    }
```

```
    }

    /* check for required args */
    if (pargs->num_warehouses == UNDEF )
    {
                    printf("Number of Warehouses is required\n");
                    exit(-2);
    }

    return;
}


//=======================================================================
//
// Function name: GetArgsLoaderUsage
//
//=======================================================================

void GetArgsLoaderUsage()
{

#ifdef DEBUG
    printf("[%ld]DBG: Entering GetArgsLoaderUsage()\n", (int) GetCurrentThreadId());
#endif


    printf("TPCCLDR:\n\n");
          printf("Parameter                                               Default\n");
          printf("-------------------------------------------------------------
\n");
    printf("-W Number of Warehouses to Load                         Required \n");
    printf("-S Server                                               %s\n", SERVER);
    printf("-U Username                                             %s\n", USER);
    printf("-P Password                                             %s\n", PASSWORD);
    printf("-D Database                                             %s\n", DATABASE);
          printf("-b Batch Size                                           %ld\n",
(long) BATCH);
          printf("-p TDS packet size                                      %ld\n",
(long) DEFLDPACKSIZE);
          printf("-f Loader Results Output Filename                       %s\n",
LOADER_RES_FILE);
          printf("-s Starting Warehouse                                   %ld\n",
(long) DEF_STARTING_WAREHOUSE);
          printf("-i Build Option (data = 0, data and index = 1)          %ld\n",
(long) BUILD_INDEX);
          printf("-o Cluster Index Build Order (before = 1, after = 0)    %ld\n",
(long) INDEX_ORDER);
          printf("-c Build Scaled Database (normal = 0, tiny = 1)         %ld\n",
(long) SCALE_DOWN);
          printf("-d Index Script Path                                    %s\n",
INDEX_SCRIPT_PATH);
          printf("-t Table to Load                                        all tables
\n");
    printf("   [item|warehouse|customer|orders]\n");
    printf("   Notes: \n");
    printf("     - the ’-t’ parameter may be included multiple times to \n");
    printf("       specify multiple tables to be loaded \n");
    printf("     - ’item’ loads ITEM table \n");
    printf("     - ’warehouse’ loads WAREHOUSE, DISTRICT, and STOCK tables \n");
    printf("     - ’customer’ loads CUSTOMER and HISTORY tables \n");
    printf("     - ’orders’ load NEW-ORDER, ORDERS, ORDER-LINE tables \n");
```

```
          printf("\nNote:  Command line switches are case sensitive.\n");

    exit(0);
}
```

Appendix C:
Tunable Parameters

Microsoft Windows NT v. 4.0 Configuration Parameters

The default value for the ProcessorAffinityMask was changed as shown below (using the regedt32 command)

```
[HKEY_LOCAL_MACHINE\SYSTEM\
CurrentControlSet\Services\NDIS\Parameters]
"ProcessorAffinityMask"=dword:00000000
```

The details about this parameter is fully documented in Microsoft Windows NT Server resource Kit (CD) - Online Help File (REGENTRY.HLP)

Microsoft SQL Server Startup Parameters

```
C:\MSSQL7\BINN\SQLSERVR
-eC:\MSSQL7\LOG\ERRORLOG -x -c -t3502
```

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

Microsoft Windows NT v. 4.0 Configuration Parameters

The following services were disabled in the Windows NT Control Panel/Services on both the Server and the Client systems:

- Computer Browser
- License Logging Service
- Messenger
- NT LM Security Support Provider
- Plug and Play
- Server
- Spooler
- TPC/IP Netbios Helper

Set Control Panel -> System -> Performance -> Forground Application Boost to Maximum

Set Control Panel -> Network -> Services -> Server -> Maximize Throughput For Network Applications

Boot.ini Parameters

4GB Tuning was enabled for Windows NT Server Enterprise Edition by setting the “/3GB” switch in the boot.ini file as documented in the “Windows NT Server, Enterprise Edition Administrator’s Guide and Release Notes”.

SQL Server Stack Size

The default stack size for Microsoft SQL Server was changed using the EDITBIN utility. The EDITBIN utility ships with Microsoft Visual C++ V6.0. The command used to change the stack size is:

```
        editbin /Stack:131072 sqlservr.exe
```

This command is fully documented as an article in the Microsoft Knowledge Base on the Microsoft Web Site at www.microsoft.com/support.

Compaq Device Drivers for Windows NT

The following Microsoft Windows NT device drivers were replaced with Compaq-specific device drivers:

- The Microsoft SMART-4200 Array Controller default device driver (CPQARRY2.SYS) was replaced with the Compaq SMART-4200 Array Controller ECI Driver version 1.09 for Microsoft Windows NT 4.0 (CPQSMRT2.SYS). This driver will available on the Compaq Computer Corporation World Wide Web site (www.compaq.com) by August 16, 1999.

Microsoft SQL Server 7.0 Configuration Parameters

```
name
minimum     maximum     config_value run_value
 ----------------------------------------------------
------------------ ----------- ----------- ----------
-- -----------
 affinity mask
0  2147483647           15          15
 allow updates
0           1            0           0
 cost threshold for parallelism
0       32767            5           5
 cursor threshold
-1  2147483647           -1          -1
 default language
0        9999            0           0
 default sortorder id
0         255           50          50
 extended memory size (MB)
0  2147483647            0           0
 fill factor (%)
0         100            0           0
 index create memory (KB)
704     1600000          704         704
 language in cache
3         100            3           3
 language neutral full-text
0           1            0           0
 lightweight pooling
0           1            1           1
 locks
5000  2147483647            0           0
 max async IO
1         255          255         255
 max degree of parallelism
0          32            1           1
 max server memory (MB)
4  2147483647            0  2147483647
 max text repl size (B)
0  2147483647        65536       65536
 max worker threads
10        1024          220         220
 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       65535         4096        4096
 open objects
0  2147483647            0           0
 priority boost
0           1            1           1
 query governor cost limit
0  2147483647            0           0
 query wait (s)
-1  2147483647           -1          -1
 recovery interval (min)
0       32767        32767       32767
 remote access
0           1            1           1
 remote login timeout (s)
0  2147483647            5           5
 remote proc trans
0           1            0           0
 remote query timeout (s)
0  2147483647            0           0
 resource timeout (s)
5  2147483647           10          10
 scan for startup procs
0           1            0           0
 set working set size
0           1            0           0
 show advanced options
0           1            1           1
 spin counter
1  2147483647        10000       10000
 time slice (ms)
50        1000          200         200
 two digit year cutoff
1753        9999         2049        2049
 Unicode comparison style
0  2147483647            0           0
 Unicode locale id
0  2147483647        33280       33280
 user connections
0       32767            0           0
 user options
0        4095            0           0
```

Internet Information Server Registry Parameters

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\InetInfo]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\InetInfo\Parameters]
"BandwidthLevel"=dword:ffffffff
"ListenBackLog"=dword:00001770
"PoolThreadLimit"=dword:00000226
"ThreadTimeout"=dword:00030d40

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\InetInfo\Parameters\Filter]
"FilterType"=dword:00000000
"NumGrantSites"=dword:00000000
"NumDenySites"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\InetInfo\Parameters\MimeMap]
"text/html,htm,,h"=""
"image/gif,gif,,g"=""
"image/jpeg,jpg,,:"=""
"text/plain,txt,,0"=""
"text/html,html,,h"=""
"image/jpeg,jpeg,,:"=""
"image/jpeg,jpe,,:"=""
"image/bmp,bmp,,:"=""
"application/octet-stream,*,,5"=""
"application/pdf,pdf,,5"=""
"application/octet-stream,bin,,5"=""
"application/oda,oda,,5"=""
"application/zip,zip,,9"=""
"application/rtf,rtf,,5"=""
"application/postscript,ps,,5"=""
"application/postscript,ai,,5"=""
"application/postscript,eps,,5"=""
"application/mac-binhex40,hqx,,4"=""
"application/msword,doc,,5"=""
"application/msword,dot,,5"=""
"application/winhlp,hlp,,5"=""
"video/mpeg,mpeg,,;"=""
"video/mpeg,mpg,,;"=""
"video/mpeg,mpe,,;"=""
"video/x-msvideo,avi,,<"=""
"video/quicktime,qt,,;"=""
"video/quicktime,mov,,;"=""
"video/x-sgi-movie,movie,,<"=""
"x-world/x-vrml,wrl,,5"=""
"x-world/x-vrml,xaf,,5"=""
"x-world/x-vrml,xof,,5"=""
"x-world/x-vrml,flr,,5"=""
"x-world/x-vrml,wrz,,5"=""
"application/x-director,dcr,,5"=""
"application/x-director,dir,,5"=""
"application/x-director,dxr,,5"=""
"image/cis-cod,cod,,5"=""
"image/x-cmx,cmx,,5"=""
"application/envoy,evy,,5"=""
"application/x-msaccess,mdb,,5"=""
"application/x-mscardfile,crd,,5"=""
"application/x-msclip,clp,,5"=""
"application/octet-stream,exe,,5"=""
"application/x-msexcel,xla,,5"=""
"application/x-msexcel,xlc,,5"=""
"application/x-msexcel,xlm,,5"=""
"application/x-msexcel,xls,,5"=""
"application/x-msexcel,xlt,,5"=""
"application/x-msexcel,xlw,,5"=""
"application/x-msmediaview,m13,,5"=""
"application/x-msmediaview,m14,,5"=""
"application/x-msmoney,mny,,5"=""
"application/x-mspowerpoint,ppt,,5"=""
"application/x-msproject,mpp,,5"=""
"application/x-mspublisher,pub,,5"=""
"application/x-msterminal,trm,,5"=""
"application/x-msworks,wks,,5"=""
```

```
"application/x-mswrite,wri,,5"=""
"application/x-msmetafile,wmf,,5"=""
"application/x-csh,csh,,5"=""
"application/x-dvi,dvi,,5"=""
"application/x-hdf,hdf,,5"=""
"application/x-latex,latex,,5"=""
"application/x-netcdf,nc,,5"=""
"application/x-netcdf,cdf,,5"=""
"application/x-sh,sh,,5"=""
"application/x-tcl,tcl,,5"=""
"application/x-tex,tex,,5"=""
"application/x-texinfo,texinfo,,5"=""
"application/x-texinfo,texi,,5"=""
"application/x-troff,t,,5"=""
"application/x-troff,tr,,5"=""
"application/x-troff,roff,,5"=""
"application/x-troff-man,man,,5"=""
"application/x-troff-me,me,,5"=""
"application/x-troff-ms,ms,,5"=""
"application/x-wais-source,src,,7"=""
"application/x-bcpio,bcpio,,5"=""
"application/x-cpio,cpio,,5"=""
"application/x-gtar,gtar,,9"=""
"application/x-shar,shar,,5"=""
"application/x-sv4cpio,sv4cpio,,5"=""
"application/x-sv4crc,sv4crc,,5"=""
"application/x-tar,tar,,5"=""
"application/x-ustar,ustar,,5"=""
"audio/basic,au,,<"=""
"audio/basic,snd,,<"=""
"audio/x-aiff,aif,,<"=""
"audio/x-aiff,aiff,,<"=""
"audio/x-aiff,aifc,,<"=""
"audio/x-wav,wav,,<"=""
"audio/x-pn-realaudio,ram,,<"=""
"image/ief,ief,,:"=""
"image/tiff,tiff,,:"=""
"image/tiff,tif,,:"=""
"image/x-cmu-raster,ras,,:"=""
"image/x-portable-anymap,pnm,,:"=""
"image/x-portable-bitmap,pbm,,:"=""
"image/x-portable-graymap,pgm,,:"=""
"image/x-portable-pixmap,ppm,,:"=""
"image/x-rgb,rgb,,:"=""
"image/x-xbitmap,xbm,,:"=""
"image/x-xpixmap,xpm,,:"=""
"image/x-xwindowdump,xwd,,:"=""
"text/html,stm,,h"=""
"text/plain,bas,,0"=""
"text/plain,c,,0"=""
"text/plain,h,,0"=""
"text/richtext,rtx,,0"=""
"text/tab-separated-values,tsv,,0"=""
"text/x-setext,etx,,0"=""
"application/x-perfmon,pmc,,5"=""
"application/x-perfmon,pma,,5"=""
"application/x-perfmon,pmr,,5"=""
"application/x-perfmon,pml,,5"=""
"application/x-perfmon,pmw,,5"=""

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\InetInfo\Performance]
"Library"="infoctrs.DLL"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:0000086c
"Last Help"=dword:0000086d
"First Counter"=dword:0000084e
"First Help"=dword:0000084f
```

World Wide Web Service Registry Parameters

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000000
"ImagePath"=hex(2):43,3a,5c,57,49,4e,4e,54,34,30,5c,5
3,79,73,74,65,6d,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):52,50,43,53,53,00,4e,54,4c,4
d,53,53,50,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\Parameters]
"MajorVersion"=dword:00000002
"MinorVersion"=dword:00000000
"AdminName"="Administrator"
"AdminEmail"="Admin@corp.com"
"MaxConnections"=dword:000186a0
"LogType"=dword:00000000
"LogFileDirectory"=hex(2):43,3a,5c,49,6e,65,74,4c,6f,
67,00
"LogFileTruncateSize"=dword:ffffffff
"LogFilePeriod"=dword:00000001
"LogFileFormat"=dword:00000000
"LogSqlDataSource"="HTTPLOG"
"LogSqlTableName"="Internetlog"
"LogSqlUserName"="InternetAdmin"
"LogSqlPassword"="sqllog"
"Authorization"=dword:00000001
"AnonymousUserName"="IUSR_CLIENT1"
"Default Load File"="Default.htm"
"Dir Browse Control"=dword:4000001e
"CheckForWAISDB"=dword:00000000
"CacheExtensions"=dword:00000001
"GlobalExpire"=dword:ffffffff
"ServerSideIncludesEnabled"=dword:00000001
"ServerSideIncludesExtension"=".stm"
"DebugFlags"=dword:00000008
"ScriptTimeout"=dword:00000384
"ConnectionTimeOut"=dword:00000384
"InstallPath"="C:\\WINNT40\\System32\\inetsrv"
"SecurePort"=dword:000001bb
"Filter
DLLs"="C:\\WINNT40\\System32\\inetsrv\\sspifilt.dll"
"AccessDeniedMessage"="Error: Access is Denied."
"NTAuthenticationProviders"="NTLM"
"ServerComment"=""
"AcceptExOutstanding"=dword:00001770

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\Parameters\Script Map]
".idc"="C:\\WINNT40\\System32\\inetsrv\\httpodbc.dll"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\Parameters\Virtual Roots]
"/,"="C:\\InetPub\\wwwroot,,5"
"/Scripts,"="C:\\InetPub\\scripts,,4"
"/iisadmin,"="C:\\WINNT40\\System32\\inetsrv\\iisadmi
n,,1"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\Performance]
"Library"="w3ctrs.DLL"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"Last Counter"=dword:000008a6
"Last Help"=dword:000008a7
"First Counter"=dword:0000086e
"First Help"=dword:0000086f

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,00,00,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,d8,00,14,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,d8,00,14
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,d8,00,14,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\W3SAMP]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\W3SVC\Enum]
"0"="Root\\LEGACY_W3SVC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

Server Registry Parameters

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control]
"WaitToKillServiceTimeout"="20000"
"CurrentUser"="USERNAME"
"SystemStartOptions"="3GB"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
BootVerificationProgram]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E968-E325-11CE-BFC1-08002BE10318}]
@="Display adapters"
"Class"="Display"
"Icon"="-1"
"Installer32"="Desk.Cpl,DisplayClassInstaller"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E968-E325-11CE-BFC1-08002BE10318}\0000]
"InfPath"="ati.inf"
"InfSection"="ati"
"ProviderName"="ATI Technologies Inc."
"DriverDesc"="ATI Technologies Inc. 3D RAGE IIC"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E96B-E325-11CE-BFC1-08002BE10318}]
@="Keyboard"
"Class"="Keyboard"
"Icon"="-3"
"Installer32"="SysSetup.Dll,KeyboardClassInstaller"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E96B-E325-11CE-BFC1-08002BE10318}\0000]
"InfPath"="keyboard.inf"
"InfSection"="STANDARD_Inst"
"ProviderName"="Microsoft"
"DriverDesc"="PC/AT Enhanced Keyboard (101/102-Key)"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E96B-E325-11CE-BFC1-08002BE10318}\0001]
"InfPath"="keyboard.inf"
"InfSection"="STANDARD_Inst"
"ProviderName"="Microsoft"
"DriverDesc"="PC/AT Enhanced Keyboard (101/102-Key)"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E96C-E325-11CE-BFC1-08002BE10318}]
@="Sound, video and game controllers"
"Class"="MEDIA"
"Icon"="3004"
"Installer32"="MmSys.Cpl,MediaClassInstaller"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E96D-E325-11CE-BFC1-08002BE10318}]
@="Modem"
"Class"="Modem"
"Default Service"="Modem"
"Icon"="0"
"Installer32"="Modem.Cpl,ClassInstall32"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E96F-E325-11CE-BFC1-08002BE10318}]
@="Mouse"
"Class"="Mouse"
"Icon"="-2"
"Installer32"="SysSetup.Dll,MouseClassInstaller"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E96F-E325-11CE-BFC1-08002BE10318}\0000]
"InfPath"="msmouse.inf"
"InfSection"="PS2_Inst"
"ProviderName"="Microsoft"
"DriverDesc"="Microsoft PS/2 Port Mouse"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E96F-E325-11CE-BFC1-08002BE10318}\0001]
"InfPath"="msmouse.inf"
"InfSection"="PS2_Inst"
"ProviderName"="Microsoft"
"DriverDesc"="Microsoft PS/2 Port Mouse"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E96F-E325-11CE-BFC1-08002BE10318}\0002]
"InfPath"="msmouse.inf"
"InfSection"="PS2_Inst"
"ProviderName"="Microsoft"
"DriverDesc"="Logitech PS/2 Port Mouse"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E972-E325-11CE-BFC1-08002BE10318}]
@="Network adapters"
"Class"="Net"
"Icon"="-5"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E978-E325-11CE-BFC1-08002BE10318}]
@="Ports (COM & LPT)"
"Class"="Ports"
"Installer32"="Ports.Cpl,PortsClassInstaller"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E979-E325-11CE-BFC1-08002BE10318}]
@="Printer"
"Class"="Printer"
"Icon"="-4"
"NoDisplayClass"="1"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E97B-E325-11CE-BFC1-08002BE10318}]
@="SCSI controllers"
"Class"="SCSIAdapter"
"Icon"="-10"
"Installer32"="SysSetup.Dll,ScsiClassInstaller"
"LegacyInfOption"="SCSI"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E97B-E325-11CE-BFC1-08002BE10318}\0000]
"InfPath"="scsi.inf"
"InfSection"="atapi_Inst"
"ProviderName"="Microsoft"
"DriverDesc"="IDE CD-ROM (ATAPI 1.2)/Dual-channel PCI
IDE Controller "

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E97B-E325-11CE-BFC1-08002BE10318}\0001]
"InfPath"="oem3.inf"
"InfSection"="cpqarry2_Inst"
"ProviderName"="%Cpq%"
"DriverDesc"="Compaq Integrated Smart Array/42xx
Controllers"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E97B-E325-11CE-BFC1-08002BE10318}\0002]
"InfPath"="oem0.inf"
"InfSection"="CPQSMRT2"
"ProviderName"="Microsoft"
"DriverDesc"="Compaq SMART-2 Array Controllers ECI
Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E97B-E325-11CE-BFC1-08002BE10318}\0003]
"InfPath"="fastscsi.inf"
"InfSection"="cpq32fs2_Inst"
"ProviderName"="%Cpq%"
"DriverDesc"="Compaq 32-Bit SCSI-2 Controllers"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{4D36E97E-E325-11CE-BFC1-08002BE10318}]
@="Other devices"
"Class"="Unknown"
"Icon"="-18"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Class\{6D807884-7D21-11CF-801C-08002BE10318}]
@="Tape drives"
"Class"="TapeDrive"
"Icon"="1610"
"Installer32"="SysSetup.Dll,TapeClassInstaller"
"LegacyInfOption"="TAPE"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
ComputerName]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
ComputerName\ComputerName]
"ComputerName"="TIGGER2"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
ComputerName\ActiveComputerName]
"ComputerName"="TIGGER2"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
CrashControl]
"LogEvent"=dword:00000001
"SendAlert"=dword:00000001
"CrashDumpEnabled"=dword:00000000
"AutoReboot"=dword:00000000
"DumpFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25
,5c,4d,45,4d,4f,52,59,2e,\
  44,4d,50,00
"Overwrite"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
FileSystem]
"Win31FileSystem"=dword:00000000
"NtfsDisable8dot3NameCreation"=dword:00000000
"Win95TruncatedExtensions"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
GraphicsDrivers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
GraphicsDrivers\DCI]
"Timeout"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
GroupOrderList]
"System Bus Extender"=hex:01,00,00,00,01,00,00,00
"SCSI
miniport"=hex:2a,00,00,00,00,01,00,00,01,01,00,00,19,
00,00,00,01,00,00,\

00,02,00,00,00,03,00,00,00,04,00,00,00,05,00,00,00,06
,00,00,00,07,00,00,00,\

08,00,00,00,09,00,00,00,0a,00,00,00,0b,00,00,00,0c,00
,00,00,0d,00,00,00,0e,\

00,00,00,0f,00,00,00,10,00,00,00,11,00,00,00,12,00,00
,00,13,00,00,00,14,00,\

00,00,15,00,00,00,16,00,00,00,17,00,00,00,1a,00,00,00
,18,00,00,00,1b,00,00,\

00,1c,00,00,00,1d,00,00,00,1e,00,00,00,1f,00,00,00,20
,00,00,00,23,00,00,00,\

24,00,00,00,25,00,00,00,26,00,00,00,21,00,00,00,63,00
,00,00,22,00,00,00,02,\
  01,00,00
"Primary
Disk"=hex:04,00,00,00,01,00,00,00,02,00,00,00,03,00,0
0,00,04,00,00,00
"Base"=hex:0e,00,00,00,0e,00,00,00,01,00,00,00,02,00,
00,00,03,00,00,00,04,00,\

00,00,05,00,00,00,06,00,00,00,07,00,00,00,08,00,00,00
,09,00,00,00,0a,00,00,\
  00,0b,00,00,00,0c,00,00,00,0d,00,00,00
```

```
"Keyboard
Port"=hex:02,00,00,00,02,00,00,00,01,00,00,00
"Pointer
Port"=hex:03,00,00,00,01,00,00,00,02,00,00,00,03,00,0
0,00
"Keyboard Class"=hex:01,00,00,00,01,00,00,00
"Pointer Class"=hex:01,00,00,00,01,00,00,00
"Video Init"=hex:01,00,00,00,01,00,00,00
"Video"=hex:01,00,00,00,01,00,00,00
"Video Save"=hex:01,00,00,00,01,00,00,00
"Ndis"=hex:09,00,00,00,01,00,00,00,02,00,00,00,03,00,
00,00,04,00,00,00,05,00,\

00,00,06,00,00,00,07,00,00,00,08,00,00,00,09,00,00,00
"Filter"=hex:06,00,00,00,01,00,00,00,02,00,00,00,03,0
0,00,00,04,00,00,00,05,00,\
  00,00,06,00,00,00
"Parallel arbitrator"=hex:01,00,00,00,01,00,00,00
"Extended
base"=hex:04,00,00,00,01,00,00,00,02,00,00,00,03,00,0
0,00,04,00,00,\
  00
"SpoolerGroup"=hex:02,00,00,00,01,00,00,00,02,00,00,0
0
"SCSI
Class"=hex:03,00,00,00,01,00,00,00,02,00,00,00,03,00,
00,00
"SCSI CDROM
Class"=hex:02,00,00,00,01,00,00,00,02,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
IDConfigDB]
"CurrentConfig"=dword:00000001
"PropertyProviders"=hex(7):70,72,6f,66,65,78,74,2e,64
,6c,6c,00,00
"UserWaitInterval"=dword:0000001e
"IsPortable"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
IDConfigDB\Hardware Profiles]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
IDConfigDB\Hardware Profiles\0001]
"FriendlyName"="Original Configuration"
"PreferenceOrder"=dword:00000000
"DockState"=dword:00000007
"DockID"=""
"SerialNumber"=""

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layout]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layout\DosKeybCodes]
"0000080C"="be"
"00000813"="be"
"00000402"="bg"
"00000416"="br"
"00000C0C"="cf"
"00000405"="cz"
"0000041B"="sl"
"00000406"="dk"
"0000040B"="su"
```

```
"0000040C"="fr"
"00000408"="gk"
"00010408"="gk"
"00020408"="gk"
"00000407"="gr"
"0000040E"="hu"
"00000410"="it"
"00010410"="it"
"0000080A"="la"
"00000413"="nl"
"00000414"="no"
"00000415"="pl"
"00010415"="pl"
"00000816"="po"
"00000418"="ro"
"0000040A"="sp"
"0001040A"="sp"
"0000041D"="sv"
"0000100C"="sf"
"00000807"="sg"
"0000041F"="tr"
"0001041F"="tr"
"00000809"="uk"
"00000409"="us"
"0000041A"="yu"
"00000424"="yu"
"00010409"="dv"
"00020409"="us"
"00030409"="usl"
"00040409"="usr"
"00001009"="us"
"00010C0C"="cf"
"0000040F"="is"
"00000419"="ru"
"0000041C"="us"
"00000422"="us"
"00000423"="us"
"00000425"="us"
"00000426"="us"
"00000427"="us"
"00000C1A"="us"
"00001809"="us"
"00010402"="us"
"00010405"="cz"
"00010407"="gr"
"00010419"="ru"
"0001041B"="sl"
"00010426"="us"
"00010C1A"="us"
"00050408"="gk"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layout\DosKeybIDs]
"0000041F"="179"
"0001041F"="440"
"00000410"="141"
"00010410"="142"
"00010408"="220"
"00020408"="319"
"00010415"="214"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000402]
"Layout Text"="Bulgarian"
"Layout File"="KBDBU.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000405]
"Layout Text"="Czech"
"Layout File"="KBDCZ.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000406]
"Layout Text"="Danish"
"Layout File"="KBDDA.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000407]
"Layout Text"="German"
"Layout File"="KBDGR.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000408]
"Layout Text"="Greek"
"Layout File"="KBDHE.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000409]
"Layout Text"="US"
"Layout File"="KBDUS.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0000040A]
"Layout Text"="Spanish"
"Layout File"="KBDSP.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0000040B]
"Layout Text"="Finnish"
"Layout File"="KBDFI.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0000040C]
"Layout Text"="French"
"Layout File"="KBDFR.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0000040E]
"Layout Text"="Hungarian"
"Layout File"="KBDHU.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0000040F]
"Layout Text"="Icelandic"
"Layout File"="KBDIC.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000410]
"Layout Text"="Italian"
"Layout File"="KBDIT.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000413]
"Layout Text"="Dutch"
"Layout File"="KBDNE.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000414]
"Layout Text"="Norwegian"
"Layout File"="KBDNO.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000415]
"Layout Text"="Polish (Programmers)"
"Layout File"="KBDPL1.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000416]
"Layout Text"="Brazilian (ABNT)"
"Layout File"="KBDBR.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000418]
"Layout Text"="Romanian"
"Layout File"="KBDRO.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000419]
"Layout Text"="Russian"
"Layout File"="KBDRU.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0000041a]
"Layout Text"="Croatian"
"Layout File"="KBDCR.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0000041B]
"Layout Text"="Slovak"
"Layout File"="KBDSL.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0000041C]
"Layout Text"="Albanian"
"Layout File"="KBDAL.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0000041D]
"Layout Text"="Swedish"
"Layout File"="KBDSW.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0000041F]
"Layout Text"="Turkish Q"
"Layout File"="KBDTUQ.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000422]
"Layout Text"="Ukrainian"
"Layout File"="KBDUR.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000423]
"Layout Text"="Belarusian"
"Layout File"="KBDBLR.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000424]
"Layout Text"="Slovenian"
"Layout File"="KBDCR.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000425]
"Layout Text"="Estonian"
"Layout File"="KBDEST.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000426]
"Layout Text"="Latvian"
"Layout File"="KBDLV.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000427]
"Layout Text"="Lithuanian"
"Layout File"="KBDLT.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000807]
"Layout Text"="Swiss German"
"Layout File"="KBDSG.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000809]
"Layout Text"="United Kingdom"
"Layout File"="KBDUK.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0000080A]
"Layout Text"="Latin American"
"Layout File"="KBDLA.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0000080C]
"Layout Text"="Belgian French"
"Layout File"="KBDBE.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000813]
"Layout Text"="Belgian Dutch"
"Layout File"="KBDBE.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000816]
"Layout Text"="Portuguese"
"Layout File"="KBDPO.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000C0C]
"Layout Text"="Canadian French"
"Layout File"="KBDFC.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00000C1A]
"Layout Text"="Serbian (Cyrillic)"
"Layout File"="KBDYCC.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00001009]
"Layout Text"="Canadian English (Multilingual)"
```

```
"Layout File"="KBDCA.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0000100C]
"Layout Text"="Swiss French"
"Layout File"="KBDSF.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00001809]
"Layout Text"="Irish"
"Layout File"="KBDIR.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00010402]
"Layout Text"="Bulgarian (Latin)"
"Layout File"="KBDUS.DLL"
"Layout Id"="0004"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00010405]
"Layout Text"="Czech (QWERTY)"
"Layout File"="KBDCZ1.DLL"
"Layout Id"="0005"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00010407]
"Layout Text"="German (IBM)"
"Layout File"="KBDGR1.DLL"
"Layout Id"="0012"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00010408]
"Layout Text"="Greek (220)"
"Layout File"="KBDHE220.DLL"
"Layout Id"="0016"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00010409]
"Layout Text"="US-Dvorak"
"Layout File"="KBDDV.DLL"
"Layout Id"="0002"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0001040A]
"Layout Text"="Spanish variation"
"Layout File"="KBDES.DLL"
"Layout Id"="0086"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0001040E]
"Layout Text"="Hungarian 101-key"
"Layout File"="KBDHU1.DLL"
"Layout Id"="0006"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00010410]
"Layout Text"="Italian (142)"
"Layout File"="KBDIT142.DLL"
"Layout Id"="0003"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00010415]
"Layout Text"="Polish (214)"
"Layout File"="KBDPL.DLL"
"Layout Id"="0007"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00010419]
"Layout Text"="Russian (Typewriter)"
"Layout File"="KBDRU1.DLL"
"Layout Id"="0008"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0001041B]
"Layout Text"="Slovak (QWERTY)"
"Layout File"="KBDSL1.DLL"
"Layout Id"="0013"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\0001041F]
"Layout Text"="Turkish F"
"Layout File"="KBDTUF.DLL"
"Layout Id"="0014"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00010426]
"Layout Text"="Latvian (QWERTY)"
"Layout File"="KBDLV1.DLL"
"Layout Id"="0015"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00010C0C]
"Layout Text"="Canadian French (Multilingual)"
"Layout File"="KBDCA.DLL"
"Layout Id"="0085"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00010C1A]
"Layout Text"="Serbian (Latin)"
"Layout File"="KBDYCL.DLL"
"Layout Id"="0009"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00011009]
"Layout File"="kbdcan.dll"
"Layout Id"="0020"
"Layout Text"="Canadian Standard"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00020408]
"Layout Text"="Greek (319)"
"Layout File"="KBDHE319.DLL"
"Layout Id"="0018"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00020409]
"Layout Text"="US-International"
"Layout File"="KBDUSX.DLL"
"Layout Id"="0001"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00030408]
"Layout Text"="Greek (220) Latin"
"Layout File"="KBDHELA2.DLL"
"Layout Id"="0017"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00030409]
"Layout Text"="US-Dvorak for left hand"
"Layout File"="KBDUSL.DLL"
"Layout Id"="001A"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00040408]
"Layout Text"="Greek (319) Latin"
"Layout File"="KBDHELA3.DLL"
"Layout Id"="0011"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00040409]
"Layout Text"="US-Dvorak for right hand"
"Layout File"="KBDUSR.DLL"
"Layout Id"="001B"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Keyboard Layouts\00050408]
"Layout Text"="Greek Latin"
"Layout File"="KBDGKL.DLL"
"Layout Id"="0019"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Lsa]
"Authentication
Packages"=hex(7):6d,73,76,31,5f,30,00,00
"Bounds"=hex:00,30,00,00,00,20,00,00
"Notification
Packages"=hex(7):46,50,4e,57,43,4c,4e,54,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Lsa\AccessProviders]
"ProviderOrder"=hex(7):57,69,6e,64,6f,77,73,20,4e,54,
20,41,63,63,65,73,73,20,\
  50,72,6f,76,69,64,65,72,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Lsa\AccessProviders\Windows NT Access Provider]
"ProviderPath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,7
4,25,5c,73,79,73,74,65,6d,\
  33,32,5c,6e,74,6d,61,72,74,61,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Lsa\MSV1_0]
"Auth1"="FPNWCLNT"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
MediaProperties]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
MediaProperties\PrivateProperties]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
MediaProperties\PrivateProperties\MIDI]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
MediaProperties\PrivateProperties\MIDI\Schemes]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
MediaProperties\PrivateProperties\MIDI\Schemes\Defaul
t]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
MediaProperties\PrivateProperties\MIDI\Schemes\Defaul
t\00]
@=""
"Channels"=dword:0000ffff

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
MediaResources]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
MediaResources\MIDI]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
MediaResources\NonGeneralMIDIDriverList]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
NetworkProvider]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
NetworkProvider\Order]
"ProviderOrder"="LanmanWorkstation"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Nls]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Nls\CodePage]
"1250"="c_1250.nls"
"1251"="c_1251.nls"
"1252"="c_1252.nls"
"1253"="c_1253.nls"
"1254"="c_1254.nls"
"1255"="c_1255.nls"
"1256"="c_1256.nls"
"1257"="c_1257.nls"
"1258"="c_1258.nls"
"437"="c_437.nls"
"737"=""
"775"=""
"850"="c_850.nls"
"852"=""
"855"=""
"857"=""
"860"=""
"861"=""
"863"=""
"865"=""
"866"=""
"869"=""
"10000"="c_10000.nls"
"10006"=""
"10007"=""
"10029"=""
"10079"=""
"10081"=""
"37"=""
"500"=""
"875"=""
"1026"=""
"20261"="c_20261.nls"
"ACP"="1252"
"OEMCP"="437"
"MACCP"="10000"
"OEMHAL"="vgaoem.fon"
"20866"="c_20866.nls"
"28592"="c_28592.nls"
"28595"="c_28595.nls"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Nls\Language]
"0409"="l_intl.nls"
"0809"="l_intl.nls"
"0c09"="l_intl.nls"
"1009"="l_intl.nls"
"1409"="l_intl.nls"
"1809"="l_intl.nls"
"1c09"="l_intl.nls"
"2009"="l_intl.nls"
"2409"="l_intl.nls"
"2809"="l_intl.nls"
"2c09"="l_intl.nls"
"0402"="l_intl.nls"
"0405"="l_intl.nls"
"0406"="l_intl.nls"
"0407"="l_intl.nls"
"0807"="l_intl.nls"
"0c07"="l_intl.nls"
"1007"="l_intl.nls"
"1407"="l_intl.nls"
"0408"="l_intl.nls"
"040a"="l_intl.nls"
"080a"="l_intl.nls"
"0c0a"="l_intl.nls"
"100a"="l_intl.nls"
"140a"="l_intl.nls"
"180a"="l_intl.nls"
"1c0a"="l_intl.nls"
"200a"="l_intl.nls"
"240a"="l_intl.nls"
"280a"="l_intl.nls"
"2c0a"="l_intl.nls"
"300a"="l_intl.nls"
"340a"="l_intl.nls"
"380a"="l_intl.nls"
"3c0a"="l_intl.nls"
"400a"="l_intl.nls"
"440a"="l_intl.nls"
"480a"="l_intl.nls"
"4c0a"="l_intl.nls"
"500a"="l_intl.nls"
"040b"="l_intl.nls"
"040c"="l_intl.nls"
"080c"="l_intl.nls"
"0c0c"="l_intl.nls"
"100c"="l_intl.nls"
"140c"="l_intl.nls"
"040e"="l_intl.nls"
"040f"="l_intl.nls"
"0410"="l_intl.nls"
"0810"="l_intl.nls"
"0413"="l_intl.nls"
"0813"="l_intl.nls"
"0414"="l_intl.nls"
"0814"="l_intl.nls"
"0415"="l_intl.nls"
"0416"="l_intl.nls"
"0816"="l_intl.nls"
"0418"="l_intl.nls"
"0419"="l_intl.nls"
"041a"="l_intl.nls"
"081a"="l_intl.nls"
"0c1a"="l_intl.nls"
"041b"="l_intl.nls"
"041d"="l_intl.nls"
"081d"="l_intl.nls"
"041f"="l_intl.nls"
"0424"="l_intl.nls"
"0436"="l_intl.nls"
"041c"="l_intl.nls"
"042d"="l_intl.nls"
"0423"="l_intl.nls"
"0403"="l_intl.nls"
"0425"="l_intl.nls"
"0438"="l_intl.nls"
"0421"="l_intl.nls"
"0426"="l_intl.nls"
"0427"="l_intl.nls"
"0422"="l_intl.nls"
"Default"="0409"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print]
"MajorVersion"=dword:00000002
"MinorVersion"=dword:00000000
"PriorityClass"=dword:00000000
"PortThreadPriority"=dword:00000000
"SchedulerThreadPriority"=dword:00000000
"BeepEnabled"=dword:00000000
"NoRemotePrinterDrivers"="Windows NT Fax Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows 4.0]
"Directory"="WIN40"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows 4.0\Drivers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows 4.0\Print Processors]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows NT Alpha_AXP]
"Directory"="W32ALPHA"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows NT Alpha_AXP\Drivers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows NT Alpha_AXP\Print
Processors]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows NT PowerPC]
"Directory"="W32PPC"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows NT PowerPC\Drivers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows NT PowerPC\Print
Processors]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows NT R4000]
"Directory"="W32MIPS"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows NT R4000\Drivers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows NT R4000\Print Processors]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows NT x86]
"Directory"="W32X86"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows NT x86\Drivers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows NT x86\Print Processors]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Environments\Windows NT x86\Print
Processors\winprint]
"Driver"="winprint.dll"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Monitors]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Monitors\Local Port]
"Driver"="localmon.dll"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Monitors\PJL Language Monitor]
"Driver"="pjlmon.dll"
"EOJTimeout"=dword:0000ea60

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Printers]
"DefaultSpoolDirectory"="C:\\WINNT40\\System32\\spool
\\PRINTERS"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Providers]
"Order"=hex(7):4c,61,6e,4d,61,6e,20,50,72,69,6e,74,20
,53,65,72,76,69,63,65,73,\
  00,00
"EventLog"=dword:0000001f
"NetPopup"=dword:00000001
"RetryPopup"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Print\Providers\LanMan Print Services]
"Name"="win32spl.dll"
"DisplayName"="LanMan Print Services"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
PriorityControl]
"Win32PrioritySeparation"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
ProductOptions]
"ProductType"="ServerNT"
"ProductSuite"=hex(7):45,6e,74,65,72,70,72,69,73,65,0
0,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurePipeServers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurePipeServers\winreg]
"Description"="Registry Server"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurePipeServers\winreg\AllowedPaths]
"Machine"=hex(7):53,79,73,74,65,6d,5c,43,75,72,72,65,
6e,74,43,6f,6e,74,72,6f,\

6c,53,65,74,5c,43,6f,6e,74,72,6f,6c,5c,50,72,6f,64,75
,63,74,4f,70,74,69,6f,\

6e,73,00,53,79,73,74,65,6d,5c,43,75,72,72,65,6e,74,43
,6f,6e,74,72,6f,6c,53,\

65,74,5c,43,6f,6e,74,72,6f,6c,5c,50,72,69,6e,74,5c,50
,72,69,6e,74,65,72,73,\

00,53,79,73,74,65,6d,5c,43,75,72,72,65,6e,74,43,6f,6e
,74,72,6f,6c,53,65,74,\

5c,53,65,72,76,69,63,65,73,5c,45,76,65,6e,74,6c,6f,67
,00,53,6f,66,74,77,61,\

72,65,5c,4d,69,63,72,6f,73,6f,66,74,5c,57,69,6e,64,6f
,77,73,20,4e,54,5c,43,\

75,72,72,65,6e,74,56,65,72,73,69,6f,6e,00,53,79,73,74
,65,6d,5c,43,75,72,72,\

65,6e,74,43,6f,6e,74,72,6f,6c,53,65,74,5c,53,65,72,76
,69,63,65,73,5c,52,65,\
  70,6c,69,63,61,74,6f,72,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders]
"SecurityProviders"="schannel.dll"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\CertificationAuthorities]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\CertificationAuthorities\A
T&T Certificate Services]
"CACert"=hex:30,82,01,e9,30,82,01,52,02,05,02,97,00,0
0,01,30,0d,06,09,2a,86,48,\

86,f7,0d,01,01,02,05,00,30,3b,31,0b,30,09,06,03,55,04
,06,13,02,55,53,31,0d,\

30,0b,06,03,55,04,0a,14,04,41,54,26,54,31,1d,30,1b,06
,03,55,04,0b,13,14,43,\

65,72,74,69,66,69,63,61,74,65,20,53,65,72,76,69,63,65
,73,30,1e,17,0d,39,36,\

30,31,32,39,30,30,30,30,30,30,5a,17,0d,39,39,31,32,33
,31,30,30,30,30,30,30,\

5a,30,3b,31,0b,30,09,06,03,55,04,06,13,02,55,53,31,0d
,30,0b,06,03,55,04,0a,\

14,04,41,54,26,54,31,1d,30,1b,06,03,55,04,0b,13,14,43
,65,72,74,69,66,69,63,\

61,74,65,20,53,65,72,76,69,63,65,73,30,81,9f,30,0d,06
,09,2a,86,48,86,f7,0d,\

01,01,01,05,00,03,81,8d,00,30,81,89,02,81,81,00,e0,1e
,01,64,1b,b8,14,c7,fe,\

e1,e6,08,a2,09,c1,91,d6,06,9b,f2,93,d9,c3,7e,a9,66,41
,9c,51,67,ef,13,66,a1,\

05,27,01,ca,10,4b,f9,7d,75,2d,a9,3a,86,be,37,9d,72,19
,47,44,0c,64,47,46,4f,\

2a,b5,a0,b5,15,8b,d3,4e,63,c3,03,f8,c7,ea,14,97,b9,29
,2e,b6,64,19,58,bb,00,\

cc,bd,08,d6,39,c2,71,4f,9e,de,e8,87,61,30,85,e0,d5,fc
,7c,d5,86,2a,db,93,16,\

f0,18,c3,36,60,ba,e5,3c,c6,10,46,c9,d5,7c,ae,47,b6,68
,21,02,03,01,00,01,30,\

0d,06,09,2a,86,48,86,f7,0d,01,01,02,05,00,03,81,81,00
,18,f2,41,8b,9c,11,58,\

15,27,42,64,84,5d,fb,75,71,41,87,9e,4f,c7,a4,5b,1b,e9
,24,e4,ff,94,71,b0,2a,\

6a,c4,fc,50,2b,d8,a8,d0,6e,be,09,60,3f,88,c4,1c,37,69
,f8,24,9e,d8,c6,0e,25,\

34,f4,7a,a1,44,99,77,5b,80,b7,77,d2,31,30,f1,ff,4b,90
,81,9c,80,32,fa,a8,62,\

64,ff,1d,1a,2d,03,e0,eb,b7,4d,bb,e3,a3,8d,de,4c,85,0d
,c3,98,b2,a4,6b,6d,18,\

7f,fc,f4,20,2a,60,7e,9a,95,73,46,9b,99,5f,49,b7,ea,70
,3d,ed,8a
"Enabled"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\CertificationAuthorities\A
T&T Directory Services]
```

```
"CACert"=hex:30,82,01,df,30,82,01,48,02,01,00,30,0d,0
6,09,2a,86,48,86,f7,0d,01,\

01,04,05,00,30,39,31,0b,30,09,06,03,55,04,06,13,02,55
,53,31,0d,30,0b,06,03,\

55,04,0a,14,04,41,54,26,54,31,1b,30,19,06,03,55,04,0b
,14,12,44,69,72,65,63,\

74,6f,72,79,20,53,65,72,76,69,63,65,73,30,1e,17,0d,39
,36,30,31,31,38,32,31,\

30,33,35,32,5a,17,0d,30,31,30,31,31,36,32,31,30,33,35
,32,5a,30,39,31,0b,30,\

09,06,03,55,04,06,13,02,55,53,31,0d,30,0b,06,03,55,04
,0a,14,04,41,54,26,54,\

31,1b,30,19,06,03,55,04,0b,14,12,44,69,72,65,63,74,6f
,72,79,20,53,65,72,76,\

69,63,65,73,30,81,9d,30,0d,06,09,2a,86,48,86,f7,0d,01
,01,01,05,00,03,81,8b,\

00,30,81,87,02,81,81,00,87,64,72,89,0b,20,8f,87,27,ac
,c6,22,fe,00,40,69,48,\

af,c6,86,cd,23,33,e3,11,c5,31,1a,1f,7e,9e,92,13,b6,a2
,ac,e3,b0,1f,2a,07,6c,\

b6,d4,de,4b,fa,f1,a2,a0,7d,ce,4b,be,be,26,48,09,8c,85
,11,de,cb,22,e7,c2,ee,\

44,51,fe,67,d5,5b,5a,e0,16,37,54,04,b8,3b,32,12,94,83
,9e,b1,4d,80,6c,a4,a9,\

76,ac,b8,a4,97,f7,ab,0b,6c,a5,43,ba,6e,4f,c5,4e,00,30
,16,3c,3f,99,14,da,a2,\

20,08,8b,ba,ed,76,ac,97,00,d5,6d,02,01,0f,30,0d,06,09
,2a,86,48,86,f7,0d,01,\

01,04,05,00,03,81,81,00,38,50,1d,0a,d3,1b,bb,a2,9f,6c
,8d,10,aa,42,1b,05,8d,\

e4,25,ab,fb,55,ae,6d,ba,53,67,15,07,9a,ec,55,9f,72,89
,5f,24,b0,db,ca,64,bd,\

64,aa,c2,8c,d9,3d,a2,45,b7,c6,92,71,51,ef,ed,e1,51,54
,97,56,35,a1,ce,e4,44,\

c4,47,66,ff,91,da,88,9c,23,c2,b3,d4,62,4a,bc,94,55,9c
,80,8e,b3,dd,4f,1a,ed,\

12,5a,b5,2e,bc,f8,4b,ce,c6,d4,70,b3,b3,22,f8,5e,5c,36
,7a,a6,b8,39,73,46,43,\

5c,9b,9a,bd,1e,7e,a7,04,cf,25,36,38,40,2d,aa,0e,aa,00
,2d,aa,0e,c6,d4,70,b3,\

b3,22,f8,5e,5c,36,7a,a6,b8,39,73,46,43,5c,9b,9a,bd,1e
,7e,a7,04,cf,25,36,00,\

00,2d,aa,03,dd,4f,1a,ed,12,5a,b5,aa,00,2d,aa,0e,c6,d4
,70,b3,b3,22,f8,5e,5c,\

36,7a,a6,b8,39,73,46,43,5c,9b,9a,bd,1e,7e,a7,04,cf,25
,36,00,00,2d,aa,03,dd,\
  4f,1a,ed,12,5a,b5
"Enabled"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\CertificationAuthorities\A
T&T Prototype Research CA]
"CACert"=hex:30,82,02,08,30,82,01,71,02,01,00,30,0d,0
6,09,2a,86,48,86,f7,0d,01,\

01,04,05,00,30,4e,31,0b,30,09,06,03,55,04,06,13,02,55
,53,31,1f,30,1d,06,03,\

55,04,0a,14,16,41,54,26,54,20,42,65,6c,6c,20,4c,61,62
,6f,72,61,74,6f,72,69,\

65,73,31,1e,30,1c,06,03,55,04,0b,14,15,50,72,6f,74,6f
,74,79,70,65,20,52,65,\

73,65,61,72,63,68,20,43,41,30,1e,17,0d,39,35,30,34,31
,33,32,31,30,36,35,36,\

5a,17,0d,39,37,30,34,31,32,32,31,30,36,35,36,5a,30,4e
,31,0b,30,09,06,03,55,\

04,06,13,02,55,53,31,1f,30,1d,06,03,55,04,0a,14,16,41
,54,26,54,20,42,65,6c,\

6c,20,4c,61,62,6f,72,61,74,6f,72,69,65,73,31,1e,30,1c
,06,03,55,04,0b,14,15,\

50,72,6f,74,6f,74,79,70,65,20,52,65,73,65,61,72,63,68
,20,43,41,30,81,9c,30,\

0d,06,09,2a,86,48,86,f7,0d,01,01,01,05,00,03,81,8a,00
,30,81,86,02,81,80,79,\

b3,a6,82,d4,82,97,ec,02,61,97,3c,e9,41,98,79,32,d8,f1
,c8,e6,5b,63,f4,14,df,\

13,a1,3f,ee,da,91,02,0a,81,5d,ae,5c,9c,5f,43,91,df,0b
,74,66,1a,45,5c,b5,4d,\

78,39,c7,c0,41,3d,bd,e3,54,8a,1e,e9,08,bd,99,04,5a,a8
,22,84,24,07,59,59,99,\

74,62,56,d8,c6,a2,7c,fc,04,b9,bf,aa,b1,4c,60,76,f4,96
,67,bf,3a,5f,ac,d7,5f,\

26,c0,a5,71,9e,7e,9b,48,07,88,fc,e5,ca,84,b6,95,75,86
,56,93,e6,84,6b,11,03,\

91,59,02,01,0f,30,0d,06,09,2a,86,48,86,f7,0d,01,01,04
,05,00,03,81,81,00,01,\

93,06,d7,89,59,99,9f,1b,6a,19,1a,1d,a4,d5,f5,d4,1f,e2
,9d,85,66,24,13,17,01,\

cf,f5,9c,86,b1,73,c2,f8,b4,f9,74,6e,62,4d,a4,66,ca,33
,48,25,d3,d6,a2,16,43,\

97,de,e4,01,c8,bb,f5,a0,67,76,fb,84,bd,12,8e,8c,b1,ec
,dd,64,46,04,8c,04,46,\

70,1c,9d,8f,4c,cf,cb,bf,cc,81,ca,db,73,e3,61,5c,46,19
,a6,5d,eb,af,4e,27,00,\

f4,09,8c,aa,1f,6d,87,1f,d9,c5,50,66,a8,25,db,2f,df,d0
,a1,5b,43,51,74,b1,3d,\
  fb,2f
"Enabled"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\CertificationAuthorities\G
TE Cybertrust ROOT]
"CACert"=hex:30,82,01,f5,30,82,01,5e,02,01,00,30,0d,0
6,09,2a,86,48,86,f7,0d,01,\

01,04,05,00,30,45,31,0b,30,09,06,03,55,04,06,13,02,55
,53,31,18,30,16,06,03,\

55,04,0a,13,0f,47,54,45,20,43,6f,72,70,6f,72,61,74,69
,6f,6e,31,1c,30,1a,06,\

03,55,04,03,13,13,47,54,45,20,43,79,62,65,72,54,72,75
,73,74,20,52,6f,6f,74,\

30,1a,17,0b,39,36,30,32,32,33,31,39,31,35,5a,17,0b,39
,39,31,32,33,31,32,33,\

35,39,5a,30,45,31,0b,30,09,06,03,55,04,06,13,02,55,53
,31,18,30,16,06,03,55,\

04,0a,13,0f,47,54,45,20,43,6f,72,70,6f,72,61,74,69,6f
,6e,31,1c,30,1a,06,03,\

55,04,03,13,13,47,54,45,20,43,79,62,65,72,54,72,75,73
,74,20,52,6f,6f,74,30,\

81,9f,30,0d,06,09,2a,86,48,86,f7,0d,01,01,01,05,00,03
,81,8d,00,30,81,89,02,\

81,81,00,b8,e6,4f,ba,db,98,7c,71,7c,af,44,b7,d3,0f,46
,d9,64,e5,93,c1,42,8e,\

c7,ba,49,8d,35,2d,7a,e7,8b,bd,e5,05,31,59,c6,b1,2f,0a
,0c,fb,9f,a7,3f,a2,09,\

66,84,56,1e,37,29,1b,87,e9,7e,0c,ca,9a,9f,a5,7f,f5,15
,94,a3,d5,a2,46,82,d8,\

68,4c,d1,37,15,06,68,af,bd,f8,b0,b3,f0,29,f5,95,5a,09
,16,61,77,0a,22,25,d4,\

4f,45,aa,c7,bd,e5,96,df,f9,d4,a8,8e,42,cc,24,c0,1e,91
,27,4a,b5,6d,06,80,63,\

39,c4,a2,5e,38,03,02,03,01,00,01,30,0d,06,09,2a,86,48
,86,f7,0d,01,01,04,05,\
00,03,81,81,00,34,9b,09,05,9d,42,48,7f,7c,24,8e,c2,88
,65,78,6b,4d,2a,0e,af,\

ed,ed,f0,6a,2d,37,eb,57,51,0e,8a,ad,2d,13,02,a1,40,6a
,41,f6,69,72,bf,0f,a0,\

4e,55,e7,82,a6,57,91,8b,5d,43,08,c9,f8,71,07,82,e3,3e
,14,01,24,0c,39,eb,61,\

22,f7,a2,0b,42,23,14,ca,10,18,36,0a,05,d1,c8,4a,f1,0e
,b3,fd,00,d0,c6,ee,55,\

f2,4c,62,b4,19,73,1f,48,18,d8,b0,4e,b0,75,bc,b9,84,8b
,1f,96,8a,83,f1,a0,1a,\
  a2,ed,ff,87,8c,a3,4f,a2
"Enabled"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\CertificationAuthorities\i
nternetMCI Mall]
"CACert"=hex:30,82,01,f3,30,82,01,5c,02,05,02,6d,00,0
0,42,30,0d,06,09,2a,86,48,\

86,f7,0d,01,01,02,05,00,30,40,31,0b,30,09,06,03,55,04
,06,13,02,55,53,31,0c,\

30,0a,06,03,55,04,0a,13,03,4d,43,49,31,14,30,12,06,03
,55,04,0b,13,0b,69,6e,\

74,65,72,6e,65,74,4d,43,49,31,0d,30,0b,06,03,55,04,0b
,13,04,4d,41,4c,4c,30,\

1e,17,0d,39,36,30,37,31,36,30,30,30,30,30,30,5a,17,0d
,39,38,30,37,31,36,32,\

33,35,39,35,39,5a,30,40,31,0b,30,09,06,03,55,04,06,13
,02,55,53,31,0c,30,0a,\

06,03,55,04,0a,13,03,4d,43,49,31,14,30,12,06,03,55,04
,0b,13,0b,69,6e,74,65,\

72,6e,65,74,4d,43,49,31,0d,30,0b,06,03,55,04,0b,13,04
,4d,41,4c,4c,30,81,9f,\

30,0d,06,09,2a,86,48,86,f7,0d,01,01,01,05,00,03,81,8d
,00,30,81,89,02,81,81,\

00,e3,21,23,5f,51,a8,2d,c0,8f,ae,10,f0,6f,ce,48,80,f2
,94,5e,88,83,fc,e4,3f,\

0d,37,1e,34,f0,08,0c,92,4d,a3,fb,cb,a5,06,e9,9e,d6,36
,30,06,27,cc,49,88,df,\

c5,9f,5f,93,6c,1f,a9,41,ec,a0,d4,d2,98,76,55,6f,c5,a3
,a5,42,49,86,f7,25,c6,\

1e,c8,43,31,91,fb,49,e6,df,4e,36,cc,6d,d5,e0,fe,18,8c
,43,f2,eb,13,39,95,27,\

d4,2b,ac,98,eb,a9,0f,2d,21,9b,5c,9b,5b,cf,1c,44,19,26
,81,6e,88,ed,9a,63,bb,\

56,76,51,6f,02,03,01,00,01,30,0d,06,09,2a,86,48,86,f7
,0d,01,01,02,05,00,03,\

81,81,00,5f,ea,bf,de,f3,27,c1,8f,48,86,23,c8,5f,e2,20
,82,d1,0e,7f,85,3a,f5,\

dc,b2,db,b9,3f,4d,dc,36,da,fe,58,84,4c,df,ca,7a,dc,9b
,11,1e,9c,65,7c,72,22,\

e4,eb,95,5b,94,88,fe,9a,10,33,5b,09,a7,a4,5c,4f,94,62
,59,50,6c,0a,be,a4,bd,\

1c,5c,5a,b6,70,72,6e,31,1d,15,ff,60,e5,5c,5a,f6,2b,05
,e9,88,5f,aa,8b,d2,34,\

fa,08,07,24,92,a1,74,85,6b,46,a6,f6,7e,26,7b,d1,f9,18
,08,93,42,93,6a,f1,29,\
  e3,73,23,d6,4e,83
"Enabled"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\CertificationAuthorities\K
eywitness Canada Inc.]
"CACert"=hex:30,82,01,fb,30,82,01,64,02,01,15,30,0d,0
6,09,2a,86,48,86,f7,0d,01,\

01,04,05,00,30,47,31,0b,30,09,06,03,55,04,06,13,02,43
,41,31,38,30,36,06,03,\

55,04,03,13,2f,4b,65,79,77,69,74,6e,65,73,73,20,43,61
,6e,61,64,61,20,49,6e,\

63,2e,20,6b,65,79,77,69,74,6e,65,73,73,40,6b,65,79,77
,69,74,6e,65,73,73,2e,\

63,61,30,1e,17,0d,39,36,30,35,30,37,30,30,30,30,30,30
,5a,17,0d,39,39,30,35,\

30,37,30,30,30,30,30,30,5a,30,47,31,0b,30,09,06,03,55
,04,06,13,02,43,41,31,\

38,30,36,06,03,55,04,03,13,2f,4b,65,79,77,69,74,6e,65
,73,73,20,43,61,6e,61,\

64,61,20,49,6e,63,2e,20,6b,65,79,77,69,74,6e,65,73,73
,40,6b,65,79,77,69,74,\

6e,65,73,73,2e,63,61,30,81,9d,30,0d,06,09,2a,86,48,86
,f7,0d,01,01,01,05,00,\

03,81,8b,00,30,81,87,02,81,81,00,cd,23,fa,2a,e1,ed,98
,f4,e9,d0,93,3e,d7,7a,\

80,02,4c,cc,c1,02,af,5c,b6,1f,7f,fa,57,42,6f,30,d1,20
,c5,b5,21,07,40,2c,a9,\

86,c2,f3,64,84,ae,3d,85,2e,ed,85,bd,54,b0,18,28,ef,6a
,f8,1b,e7,0b,16,1f,93,\

25,4f,c7,f8,8e,c3,b9,ca,98,84,0e,55,d0,2f,ef,78,77,c5
,72,28,5f,60,bf,19,2b,\

d1,72,a2,b7,d8,3f,e0,97,34,5a,01,bd,04,9c,c8,78,45,cd
,93,8d,15,f2,76,10,11,\

ab,b8,5b,2e,9e,52,dd,81,3e,9c,64,c8,29,93,02,01,03,30
,0d,06,09,2a,86,48,86,\

f7,0d,01,01,04,05,00,03,81,81,00,c3,1c,0a,8b,49,99,bf
,7e,64,14,4c,51,30,25,\

99,de,cc,e7,40,15,ab,9a,6a,ad,25,31,25,a2,1b,5c,84,91
,8e,32,1c,d6,55,fe,d7,\

54,66,5e,9c,f2,53,78,62,bc,80,1d,c1,7e,9f,a1,97,15,84
,62,d7,4d,2d,32,30,f9,\

62,eb,33,bd,ca,92,f4,89,ee,14,0d,f9,2c,c0,e7,30,36,af
,0d,0e,7d,a3,4c,c7,4e,\

f7,03,d3,88,c6,45,74,bd,76,bc,81,cd,e0,9c,15,a0,58,3e
,c8,74,24,0f,4d,55,91,\
  0d,13,07,b4,84,ac,fb,8c,6f,42,09,63,b1,98
"Enabled"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\CertificationAuthorities\T
hawte Premium Server CA]
"CACert"=hex:30,82,03,0d,30,82,02,76,02,01,00,30,0d,0
6,09,2a,86,48,86,f7,0d,01,\

01,04,05,00,30,81,ce,31,0b,30,09,06,03,55,04,06,13,02
,5a,41,31,15,30,13,06,\

03,55,04,08,13,0c,57,65,73,74,65,72,6e,20,43,61,70,65
,31,12,30,10,06,03,55,\

04,07,13,09,43,61,70,65,20,54,6f,77,6e,31,1d,30,1b,06
,03,55,04,0a,13,14,54,\

68,61,77,74,65,20,43,6f,6e,73,75,6c,74,69,6e,67,20,63
,63,31,28,30,26,06,03,\

55,04,0b,13,1f,43,65,72,74,69,66,69,63,61,74,69,6f,6e
,20,53,65,72,76,69,63,\

65,73,20,44,69,76,69,73,69,6f,6e,31,21,30,1f,06,03,55
,04,03,13,18,54,68,61,\

77,74,65,20,50,72,65,6d,69,75,6d,20,53,65,72,76,65,72
,20,43,41,31,28,30,26,\

06,09,2a,86,48,86,f7,0d,01,09,01,16,19,70,72,65,6d,69
,75,6d,2d,73,65,72,76,\

65,72,40,74,68,61,77,74,65,2e,63,6f,6d,30,1e,17,0d,39
,36,30,37,32,37,31,38,\

30,37,31,34,5a,17,0d,39,38,30,37,32,37,31,38,30,37,31
,34,5a,30,81,ce,31,0b,\

30,09,06,03,55,04,06,13,02,5a,41,31,15,30,13,06,03,55
,04,08,13,0c,57,65,73,\
74,65,72,6e,20,43,61,70,65,31,12,30,10,06,03,55,04,07
,13,09,43,61,70,65,20,\

54,6f,77,6e,31,1d,30,1b,06,03,55,04,0a,13,14,54,68,61
,77,74,65,20,43,6f,6e,\

73,75,6c,74,69,6e,67,20,63,63,31,28,30,26,06,03,55,04
,0b,13,1f,43,65,72,74,\

69,66,69,63,61,74,69,6f,6e,20,53,65,72,76,69,63,65,73
,20,44,69,76,69,73,69,\

6f,6e,31,21,30,1f,06,03,55,04,03,13,18,54,68,61,77,74
,65,20,50,72,65,6d,69,\

75,6d,20,53,65,72,76,65,72,20,43,41,31,28,30,26,06,09
,2a,86,48,86,f7,0d,01,\

09,01,16,19,70,72,65,6d,69,75,6d,2d,73,65,72,76,65,72
,40,74,68,61,77,74,65,\

2e,63,6f,6d,30,81,9f,30,0d,06,09,2a,86,48,86,f7,0d,01
,01,01,05,00,03,81,8d,\

00,30,81,89,02,81,81,00,d2,36,36,6a,8b,d7,c2,5b,9e,da
,81,41,62,8f,38,ee,49,\

04,55,d6,d0,ef,1c,1b,95,16,47,ef,18,48,35,3a,52,f4,2b
,6a,06,8f,3b,2f,ea,56,\

e3,af,86,8d,9e,17,f7,9e,b4,65,75,02,4d,ef,cb,09,a2,21
,51,d8,9b,d0,67,d0,ba,\

0d,92,06,14,73,d4,93,cb,97,2a,00,9c,5c,4e,0c,bc,fa,15
,52,fc,f2,44,6e,da,11,\

4a,6e,08,9f,2f,2d,e3,f9,aa,3a,86,73,b6,46,53,58,c8,89
,05,bd,83,11,b8,73,3f,\

aa,07,8d,f4,42,4d,e7,40,9d,1c,37,02,03,01,00,01,30,0d
,06,09,2a,86,48,86,f7,\

0d,01,01,04,05,00,03,81,81,00,ba,d1,48,81,34,59,55,87
,ab,21,97,cb,f6,5b,d1,\

c3,d1,22,7e,fb,e8,e4,a4,d9,de,1f,81,8f,e5,73,36,7c,b0
,e1,07,3f,23,25,e4,3c,\

eb,ba,48,88,87,f4,b4,37,b0,29,96,05,dc,69,7b,ea,ab,e2
,5e,fb,ac,b8,f6,19,8b,\

8c,b4,44,85,7e,7d,57,54,02,f3,91,88,02,ec,97,0f,b9,33
,a3,31,f2,22,30,1c,ad,\

fa,e1,5f,f0,05,7c,c7,42,16,fd,09,2e,bf,e8,44,41,cc,8c
,f6,fc,4a,dd,56,5d,2d,\
  b0,95,22,13,68,12,cc,ac,af,10,a9,c5,c2
"Enabled"=dword:00000001
"Type"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\CertificationAuthorities\T
hawte Server CA]
"CACert"=hex:30,82,02,f9,30,82,02,62,02,01,00,30,0d,0
6,09,2a,86,48,86,f7,0d,01,\

01,04,05,00,30,81,c4,31,0b,30,09,06,03,55,04,06,13,02
,5a,41,31,15,30,13,06,\

03,55,04,08,13,0c,57,65,73,74,65,72,6e,20,43,61,70,65
,31,12,30,10,06,03,55,\

04,07,13,09,43,61,70,65,20,54,6f,77,6e,31,1d,30,1b,06
,03,55,04,0a,13,14,54,\

68,61,77,74,65,20,43,6f,6e,73,75,6c,74,69,6e,67,20,63
,63,31,28,30,26,06,03,\

55,04,0b,13,1f,43,65,72,74,69,66,69,63,61,74,69,6f,6e
,20,53,65,72,76,69,63,\

65,73,20,44,69,76,69,73,69,6f,6e,31,19,30,17,06,03,55
,04,03,13,10,54,68,61,\

77,74,65,20,53,65,72,76,65,72,20,43,41,31,26,30,24,06
,09,2a,86,48,86,f7,0d,\

01,09,01,16,17,73,65,72,76,65,72,2d,63,65,72,74,73,40
,74,68,61,77,74,65,2e,\

63,6f,6d,30,1e,17,0d,39,36,30,37,32,37,31,38,30,37,35
,37,5a,17,0d,39,38,30,\

37,32,37,31,38,30,37,35,37,5a,30,81,c4,31,0b,30,09,06
,03,55,04,06,13,02,5a,\

41,31,15,30,13,06,03,55,04,08,13,0c,57,65,73,74,65,72
,6e,20,43,61,70,65,31,\

12,30,10,06,03,55,04,07,13,09,43,61,70,65,20,54,6f,77
,6e,31,1d,30,1b,06,03,\

55,04,0a,13,14,54,68,61,77,74,65,20,43,6f,6e,73,75,6c
,74,69,6e,67,20,63,63,\

31,28,30,26,06,03,55,04,0b,13,1f,43,65,72,74,69,66,69
,63,61,74,69,6f,6e,20,\

53,65,72,76,69,63,65,73,20,44,69,76,69,73,69,6f,6e,31
,19,30,17,06,03,55,04,\

03,13,10,54,68,61,77,74,65,20,53,65,72,76,65,72,20,43
,41,31,26,30,24,06,09,\

2a,86,48,86,f7,0d,01,09,01,16,17,73,65,72,76,65,72,2d
,63,65,72,74,73,40,74,\

68,61,77,74,65,2e,63,6f,6d,30,81,9f,30,0d,06,09,2a,86
,48,86,f7,0d,01,01,01,\

05,00,03,81,8d,00,30,81,89,02,81,81,00,d3,a4,50,6e,c8
,ff,56,6b,e6,cf,5d,b6,\
```

```
ea,0c,68,75,47,a2,aa,c2,da,84,25,fc,a8,f4,47,51,da,85
,b5,20,74,94,86,1e,0f,\

75,c9,e9,08,61,f5,06,6d,30,6e,15,19,02,e9,52,c0,62,db
,4d,99,9e,e2,6a,0c,44,\

38,cd,fe,be,e3,64,09,70,c5,fe,b1,6b,29,b6,2f,49,c8,3b
,d4,27,04,25,10,97,2f,\

e7,90,6d,c0,28,42,99,d7,4c,43,de,c3,f5,21,6d,54,9f,5d
,c3,58,e1,c0,e4,d9,5b,\

b0,b8,dc,b4,7b,df,36,3a,c2,b5,66,22,12,d6,87,0d,02,03
,01,00,01,30,0d,06,09,\

2a,86,48,86,f7,0d,01,01,04,05,00,03,81,81,00,8b,2f,9f
,b8,9f,5f,74,54,22,bb,\

d8,5e,da,48,e0,33,9f,01,19,13,a2,0c,26,ea,8e,ce,c1,57
,65,f7,7c,85,84,37,17,\

ee,1e,6d,d1,76,75,d4,c5,00,33,38,8a,75,d7,b7,ae,64,ef
,cd,46,08,50,26,28,63,\

96,f4,df,62,30,18,c4,ef,76,27,25,2b,e4,93,37,a3,4f,da
,6e,67,bc,50,0c,a8,94,\

f9,80,2e,4e,fa,3f,e3,06,e6,51,43,88,b4,00,c6,10,af,91
,78,95,3f,28,04,99,e1,\

81,a7,f0,e8,f2,fc,68,36,36,bc,c1,c6,48,f9,7d,fb,bb,9f
"Enabled"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\CertificationAuthorities\V
erisign Class 2 Public Primary CA]
"CACert"=hex:30,82,02,31,30,82,01,9a,02,05,02,a3,00,0
0,01,30,0d,06,09,2a,86,48,\

86,f7,0d,01,01,02,05,00,30,5f,31,0b,30,09,06,03,55,04
,06,13,02,55,53,31,17,\

30,15,06,03,55,04,0a,13,0e,56,65,72,69,53,69,67,6e,2c
,20,49,6e,63,2e,31,37,\

30,35,06,03,55,04,0b,13,2e,43,6c,61,73,73,20,32,20,50
,75,62,6c,69,63,20,50,\

72,69,6d,61,72,79,20,43,65,72,74,69,66,69,63,61,74,69
,6f,6e,20,41,75,74,68,\

6f,72,69,74,79,30,1e,17,0d,39,36,30,31,32,39,30,30,30
,30,30,30,5a,17,0d,39,\

39,31,32,33,31,32,33,35,39,35,39,5a,30,5f,31,0b,30,09
,06,03,55,04,06,13,02,\

55,53,31,17,30,15,06,03,55,04,0a,13,0e,56,65,72,69,53
,69,67,6e,2c,20,49,6e,\
63,2e,31,37,30,35,06,03,55,04,0b,13,2e,43,6c,61,73,73
,20,32,20,50,75,62,6c,\

69,63,20,50,72,69,6d,61,72,79,20,43,65,72,74,69,66,69
,63,61,74,69,6f,6e,20,\

41,75,74,68,6f,72,69,74,79,30,81,9f,30,0d,06,09,2a,86
,48,86,f7,0d,01,01,01,\

05,00,03,81,8d,00,30,81,89,02,81,81,00,b6,5a,8b,a3,0d
,6a,23,83,80,6b,cf,39,\

87,f4,21,13,33,06,4c,25,a2,ed,55,12,97,c5,a7,80,b9,fa
,83,c1,20,a0,fa,2f,15,\

0d,7c,a1,60,6b,7e,79,2c,fa,06,0f,3a,ae,f6,1b,6f,b1,d2
,ff,2f,28,52,5f,83,7d,\

4b,c4,7a,b7,f8,66,1f,80,54,fc,b7,c2,8e,59,4a,14,57,46
,d1,9a,93,be,41,91,03,\

bb,15,80,93,5c,eb,e7,cc,08,6c,3f,3e,b3,4a,fc,ff,4b,6c
,23,d5,50,82,26,44,19,\

8e,23,c3,71,ea,19,24,47,04,9e,75,bf,c8,a6,00,1f,02,03
,01,00,01,30,0d,06,09,\

2a,86,48,86,f7,0d,01,01,02,05,00,03,81,81,00,7b,af,a4
,1c,2f,c0,9b,0e,3b,5c,\

fc,a8,e6,d1,96,0f,48,72,49,85,e5,af,1a,23,00,e3,e9,bb
,80,34,6c,b2,3f,84,a8,\

3c,72,70,b3,4c,de,73,fa,7c,84,d2,92,a5,00,37,09,c2,03
,34,23,03,b7,ed,76,96,\

dc,3e,4f,85,fd,58,17,f4,04,be,ec,51,1e,dc,7a,db,db,ae
,e8,61,78,da,65,38,47,\

ac,1d,03,70,6d,e9,e3,bb,a0,4a,9f,40,ea,bc,db,73,bf,67
,4d,5d,c6,7c,20,5c,e7,\

e8,a2,0a,2b,c2,09,74,d7,4c,fe,db,e9,6b,02,63,5d,39,a7
"Enabled"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\CertificationAuthorities\V
erisign Class 3 Public Primary CA]
"CACert"=hex:30,82,02,31,30,82,01,9a,02,05,02,a1,00,0
0,01,30,0d,06,09,2a,86,48,\

86,f7,0d,01,01,02,05,00,30,5f,31,0b,30,09,06,03,55,04
,06,13,02,55,53,31,17,\

30,15,06,03,55,04,0a,13,0e,56,65,72,69,53,69,67,6e,2c
,20,49,6e,63,2e,31,37,\

30,35,06,03,55,04,0b,13,2e,43,6c,61,73,73,20,33,20,50
,75,62,6c,69,63,20,50,\

72,69,6d,61,72,79,20,43,65,72,74,69,66,69,63,61,74,69
,6f,6e,20,41,75,74,68,\

6f,72,69,74,79,30,1e,17,0d,39,36,30,31,32,39,30,30,30
,30,30,30,5a,17,0d,39,\

39,31,32,33,31,32,33,35,39,35,39,5a,30,5f,31,0b,30,09
,06,03,55,04,06,13,02,\

55,53,31,17,30,15,06,03,55,04,0a,13,0e,56,65,72,69,53
,69,67,6e,2c,20,49,6e,\

63,2e,31,37,30,35,06,03,55,04,0b,13,2e,43,6c,61,73,73
,20,33,20,50,75,62,6c,\

69,63,20,50,72,69,6d,61,72,79,20,43,65,72,74,69,66,69
,63,61,74,69,6f,6e,20,\

41,75,74,68,6f,72,69,74,79,30,81,9f,30,0d,06,09,2a,86
,48,86,f7,0d,01,01,01,\

05,00,03,81,8d,00,30,81,89,02,81,81,00,c9,5c,59,9e,f2
,1b,8a,01,14,b4,10,df,\

04,40,db,e3,57,af,6a,45,40,8f,84,0c,0b,d1,33,d9,d9,11
,cf,ee,02,58,1f,25,f7,\

2a,a8,44,05,aa,ec,03,1f,78,7f,9e,93,b9,9a,00,aa,23,7d
,d6,ac,85,a2,63,45,c7,\

72,27,cc,f4,4c,c6,75,71,d2,39,ef,4f,42,f0,75,df,0a,90
,c6,8e,20,6f,98,0f,f8,\

ac,23,5f,70,29,36,a4,c9,86,e7,b1,9a,20,cb,53,a5,85,e7
,3d,be,7d,9a,fe,24,45,\

33,dc,76,15,ed,0f,a2,71,64,4c,65,2e,81,68,45,a7,02,03
,01,00,01,30,0d,06,09,\

2a,86,48,86,f7,0d,01,01,02,05,00,03,81,81,00,75,66,6c
,3e,d1,cd,81,db,b5,f8,\

2f,36,51,b6,f7,42,bc,cd,42,af,dc,0e,fa,15,6c,f8,67,93
,57,3a,eb,b6,92,e8,b6,\

01,ca,8c,b7,8e,43,b4,49,65,f9,3e,ee,bd,75,46,2e,c9,fc
,25,5d,a8,c7,2f,8b,9b,\

8f,68,cf,b4,9c,97,18,c0,4d,ef,1f,d9,af,82,b3,e6,64,b8
,84,5c,8a,9a,07,52,43,\

61,fb,74,9e,5b,3a,36,fc,4c,b2,fc,1a,3f,15,2e,a5,5b,3c
,1b,90,ec,88,29,e4,59,\

16,f9,ce,07,ad,ec,e9,dd,da,d2,31,8a,4f,d6,d8,ef,17,8d
"Enabled"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\CertificationAuthorities\V
erisign Class 4 Public Primary CA]
"CACert"=hex:30,82,02,31,30,82,01,9a,02,05,02,a6,00,0
0,01,30,0d,06,09,2a,86,48,\

86,f7,0d,01,01,02,05,00,30,5f,31,0b,30,09,06,03,55,04
,06,13,02,55,53,31,17,\

30,15,06,03,55,04,0a,13,0e,56,65,72,69,53,69,67,6e,2c
,20,49,6e,63,2e,31,37,\

30,35,06,03,55,04,0b,13,2e,43,6c,61,73,73,20,34,20,50
,75,62,6c,69,63,20,50,\

72,69,6d,61,72,79,20,43,65,72,74,69,66,69,63,61,74,69
,6f,6e,20,41,75,74,68,\

6f,72,69,74,79,30,1e,17,0d,39,36,30,31,32,39,30,30,30
,30,30,30,5a,17,0d,39,\

39,31,32,33,31,32,33,35,39,35,39,5a,30,5f,31,0b,30,09
,06,03,55,04,06,13,02,\

55,53,31,17,30,15,06,03,55,04,0a,13,0e,56,65,72,69,53
,69,67,6e,2c,20,49,6e,\

63,2e,31,37,30,35,06,03,55,04,0b,13,2e,43,6c,61,73,73
,20,34,20,50,75,62,6c,\

69,63,20,50,72,69,6d,61,72,79,20,43,65,72,74,69,66,69
,63,61,74,69,6f,6e,20,\

41,75,74,68,6f,72,69,74,79,30,81,9f,30,0d,06,09,2a,86
,48,86,f7,0d,01,01,01,\

05,00,03,81,8d,00,30,81,89,02,81,81,00,d0,b2,75,f6,78
,d0,ae,5a,50,f4,e9,50,\

a9,9f,8c,d7,ef,91,94,70,e8,d2,24,90,76,89,85,d6,df,ac
,e6,01,17,32,80,f0,9d,\

93,47,bc,9a,65,9d,1f,97,ae,bf,e9,86,75,63,20,89,bd,80
,58,9d,04,0c,9d,a8,c1,\

24,e9,0b,e5,31,78,bd,fc,2d,0c,37,6a,9e,78,80,e9,46,75
,f9,ed,a3,fb,13,7b,c8,\

c1,4c,d2,a3,ef,f5,3c,b0,62,8f,4a,5d,3b,dd,95,67,8f,13
,b9,c1,3c,d6,a7,26,9b,\

ec,c3,3b,7a,d9,4d,bc,6d,9b,e8,15,01,e3,f0,47,a9,02,03
,01,00,01,30,0d,06,09,\

2a,86,48,86,f7,0d,01,01,02,05,00,03,81,81,00,53,dd,d3
,f0,9c,24,7e,40,aa,e2,\

fc,00,1a,d7,da,0c,fc,32,61,b8,15,0d,96,f3,fa,57,1b,7f
,33,7c,af,e9,98,9a,61,\

c8,7a,b3,b7,ff,b1,dc,99,83,dc,ac,12,fc,70,c9,1f,38,42
,ed,44,f6,80,2e,5b,6b,\

33,69,ac,9c,d3,5c,e7,5f,5a,18,c7,b1,2d,79,04,96,41,91
,99,41,b1,3c,0d,ba,84,\
39,c6,3b,97,f0,26,c9,8e,ee,bd,cc,42,95,ff,1e,c7,02,3f
,54,0c,78,f5,bc,aa,60,\

7c,02,69,e8,dc,ac,e2,02,76,61,c4,3e,03,ea,d2,8a,24,d1
"Enabled"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\CertificationAuthorities\V
erisign/RSA Commercial]
"CACert"=hex:30,82,02,23,30,82,01,90,02,05,02,41,00,0
0,16,30,0d,06,09,2a,86,48,\

86,f7,0d,01,01,02,05,00,30,5c,31,0b,30,09,06,03,55,04
,06,13,02,55,53,31,20,\

30,1e,06,03,55,04,0a,13,17,52,53,41,20,44,61,74,61,20
,53,65,63,75,72,69,74,\

79,2c,20,49,6e,63,2e,31,2b,30,29,06,03,55,04,0b,13,22
,43,6f,6d,6d,65,72,63,\

69,61,6c,20,43,65,72,74,69,66,69,63,61,74,69,6f,6e,20
,41,75,74,68,6f,72,69,\

74,79,30,1e,17,0d,39,34,31,31,30,34,31,38,35,38,33,34
,5a,17,0d,39,39,31,31,\

30,33,31,38,35,38,33,34,5a,30,5c,31,0b,30,09,06,03,55
,04,06,13,02,55,53,31,\

20,30,1e,06,03,55,04,0a,13,17,52,53,41,20,44,61,74,61
,20,53,65,63,75,72,69,\

74,79,2c,20,49,6e,63,2e,31,2b,30,29,06,03,55,04,0b,13
,22,43,6f,6d,6d,65,72,\

63,69,61,6c,20,43,65,72,74,69,66,69,63,61,74,69,6f,6e
,20,41,75,74,68,6f,72,\

69,74,79,30,81,9b,30,0d,06,09,2a,86,48,86,f7,0d,01,01
,01,05,00,03,81,89,00,\

30,81,85,02,7e,00,a4,fb,81,62,7b,ce,10,27,dd,e8,f7,be
,6c,6e,c6,70,99,db,b8,\

d5,05,03,69,28,82,9c,72,7f,96,3f,8e,ec,ac,29,92,3f,8a
,14,f8,42,76,be,bd,5d,\

03,b9,90,d4,d0,bc,06,b2,51,33,5f,c4,c2,bf,b6,8b,8f,99
,b6,62,22,60,dd,db,df,\

20,82,b4,ca,a2,2f,2d,50,ed,94,32,de,e0,55,8d,d4,68,e2
,e0,4c,d2,cd,05,16,2e,\

95,66,5c,61,52,38,1e,51,a8,82,a1,c4,ef,25,e9,0a,e6,8b
,2b,8e,31,66,d9,f8,d9,\

fd,bd,3b,69,d9,eb,02,03,01,00,01,30,0d,06,09,2a,86,48
,86,f7,0d,01,01,02,05,\

00,03,7e,00,76,b5,b6,10,fe,23,f7,f7,59,62,4b,b0,5f,9c
,c1,68,bc,49,bb,b3,49,\

6f,21,47,5d,2b,9d,54,c4,00,28,3f,98,b9,f2,8a,83,9b,60
,7f,eb,50,c7,ab,05,10,\

2d,3d,ed,38,02,c1,a5,48,d2,fe,65,a0,c0,bc,ea,a6,23,16
,66,6c,1b,24,a9,f3,ec,\

79,35,18,4f,26,c8,e3,af,50,4a,c7,a7,31,6b,d0,7c,18,9d
,50,bf,a9,26,fa,26,2b,\

46,9c,14,a9,bb,5b,30,98,42,28,b5,4b,53,bb,43,09,92,40
,ba,a8,aa,5a,a4,c6,b6,\
  8b,57,4d,c5
"Enabled"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\CertificationAuthorities\V
erisign/RSA Secure Server]
"CACert"=hex:30,82,02,29,30,82,01,96,02,05,02,41,00,0
0,01,30,0d,06,09,2a,86,48,\

86,f7,0d,01,01,02,05,00,30,5f,31,0b,30,09,06,03,55,04
,06,13,02,55,53,31,20,\

30,1e,06,03,55,04,0a,13,17,52,53,41,20,44,61,74,61,20
,53,65,63,75,72,69,74,\

79,2c,20,49,6e,63,2e,31,2e,30,2c,06,03,55,04,0b,13,25
,53,65,63,75,72,65,20,\

53,65,72,76,65,72,20,43,65,72,74,69,66,69,63,61,74,69
,6f,6e,20,41,75,74,68,\

6f,72,69,74,79,30,1e,17,0d,39,34,31,31,30,39,32,33,35
,34,31,37,5a,17,0d,39,\

39,31,32,33,31,32,33,35,34,31,37,5a,30,5f,31,0b,30,09
,06,03,55,04,06,13,02,\

55,53,31,20,30,1e,06,03,55,04,0a,13,17,52,53,41,20,44
,61,74,61,20,53,65,63,\

75,72,69,74,79,2c,20,49,6e,63,2e,31,2e,30,2c,06,03,55
,04,0b,13,25,53,65,63,\

75,72,65,20,53,65,72,76,65,72,20,43,65,72,74,69,66,69
,63,61,74,69,6f,6e,20,\

41,75,74,68,6f,72,69,74,79,30,81,9b,30,0d,06,09,2a,86
,48,86,f7,0d,01,01,01,\

05,00,03,81,89,00,30,81,85,02,7e,00,92,ce,7a,c1,ae,83
,3e,5a,aa,89,83,57,ac,\

25,01,76,0c,ad,ae,8e,2c,37,ce,eb,35,78,64,54,03,e5,84
,40,51,c9,bf,8f,08,e2,\

8a,82,08,d2,16,86,37,55,e9,b1,21,02,ad,76,68,81,9a,05
,a2,4b,c9,4b,25,66,22,\

56,6c,88,07,8f,f7,81,59,6d,84,07,65,70,13,71,76,3e,9b
,77,4c,e3,50,89,56,98,\

48,b9,1d,a7,29,1a,13,2e,4a,11,59,9c,1e,15,d5,49,54,2c
,73,3a,69,82,b1,97,39,\

9c,6d,70,67,48,e5,dd,2d,d6,c8,1e,7b,02,03,01,00,01,30
,0d,06,09,2a,86,48,86,\

f7,0d,01,01,02,05,00,03,7e,00,88,d1,d1,79,21,ce,e2,8b
,e8,f8,c1,7d,34,53,3f,\

61,83,d9,b6,0b,38,17,b6,e8,be,21,8d,8f,00,b8,8b,53,7e
,44,67,1e,22,bd,97,27,\

e0,9c,85,cc,4a,f6,85,3b,b2,e2,be,92,d3,e5,0d,e9,af,5c
,0e,0c,46,95,ff,a1,1c,\

5e,3e,e8,36,58,7a,73,a6,0a,f8,22,11,6b,c3,09,38,7e,26
,bb,73,ef,00,bd,02,a4,\

f3,14,0d,30,3f,61,70,7b,20,fe,32,a3,9f,b3,f4,67,52,dc
,b4,ee,84,8c,96,36,20,\
  de,81,08,83,71,21,8a,0f,9e,a9
"Enabled"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\Ciphers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\Hashes]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\KeyExchangeAlgorithms]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\Protocols]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\Protocols\Multi-Protocol
Unified Hello]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\Protocols\Multi-Protocol
Unified Hello\Client]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\Protocols\Multi-Protocol
Unified Hello\Server]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\Protocols\PCT 1.0]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\Protocols\PCT 1.0\Client]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\Protocols\PCT 1.0\Server]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\Protocols\SSL 2.0]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\Protocols\SSL 2.0\Client]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\Protocols\SSL 2.0\Server]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\Protocols\SSL 3.0]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\Protocols\SSL 3.0\Client]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SecurityProviders\SCHANNEL\Protocols\SSL 3.0\Server]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
ServiceGroupOrder]
"List"=hex(7):53,79,73,74,65,6d,20,42,75,73,20,45,78,
74,65,6e,64,65,72,00,53,\

43,53,49,20,6d,69,6e,69,70,6f,72,74,00,70,6f,72,74,00
,50,72,69,6d,61,72,79,\

20,64,69,73,6b,00,53,43,53,49,20,63,6c,61,73,73,00,53
,43,53,49,20,43,44,52,\

4f,4d,20,63,6c,61,73,73,00,53,75,70,65,72,44,69,73,6b
,00,66,69,6c,74,65,72,\

00,62,6f,6f,74,20,66,69,6c,65,20,73,79,73,74,65,6d,00
,42,61,73,65,00,50,6f,\

69,6e,74,65,72,20,50,6f,72,74,00,4b,65,79,62,6f,61,72
,64,20,50,6f,72,74,00,\

50,6f,69,6e,74,65,72,20,43,6c,61,73,73,00,4b,65,79,62
,6f,61,72,64,20,43,6c,\

61,73,73,00,56,69,64,65,6f,20,49,6e,69,74,00,56,69,64
,65,6f,00,56,69,64,65,\

6f,20,53,61,76,65,00,66,69,6c,65,20,73,79,73,74,65,6d
,00,45,76,65,6e,74,20,\

6c,6f,67,00,53,74,72,65,61,6d,73,20,44,72,69,76,65,72
,73,00,50,4e,50,5f,54,\

44,49,00,4e,44,49,53,00,54,44,49,00,4e,65,74,42,49,4f
,53,47,72,6f,75,70,00,\

53,70,6f,6f,6c,65,72,47,72,6f,75,70,00,4e,65,74,44,44
,45,47,72,6f,75,70,00,\

50,61,72,61,6c,6c,65,6c,20,61,72,62,69,74,72,61,74,6f
,72,00,65,78,74,65,6e,\

64,65,64,20,62,61,73,65,00,52,65,6d,6f,74,65,56,61,6c
,69,64,61,74,69,6f,6e,\

00,50,43,49,20,43,6f,6e,66,69,67,75,72,61,74,69,6f,6e
,00,4d,53,20,54,72,61,\
  6e,73,61,63,74,69,6f,6e,73,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
ServiceProvider]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
ServiceProvider\Order]
"ExcludedProviders"=hex(7):00
"ProviderOrder"=hex(7):54,63,70,69,70,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session Manager]
"ObjectDirectories"=hex(7):5c,57,69,6e,64,6f,77,73,00
,5c,52,50,43,20,43,6f,6e,\
  74,72,6f,6c,00,00
"GlobalFlag"=dword:00000000
"ProtectionMode"=dword:00000000
"BootExecute"=hex(7):61,75,74,6f,63,68,65,63,6b,20,61
,75,74,6f,63,68,6b,20,2a,\
  00,00
"EnableMCE"=dword:00000000
"EnableMCA"=dword:00000001
"HeapSegmentReserve"=dword:00000000
"HeapSegmentCommit"=dword:00000000
"HeapDeCommitTotalFreeThreshold"=dword:00000000
"HeapDeCommitFreeBlockThreshold"=dword:00000000
"CriticalSectionTimeout"=dword:00278d00
"ResourceTimeoutCount"=dword:0009e340
"ExcludeFromKnownDlls"=hex(7):00,00
"ProcessorControl"=dword:00000002
"RegisteredProcessors"=dword:00000008
"LicensedProcessors"=dword:00000008

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session Manager\AppPatches]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session Manager\AppPatches\CWD]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session
Manager\AppPatches\CWD\ff060102423da0000407108e0500]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session
Manager\AppPatches\CWD\ff060102423da0000407108e0500\1
]
"Add1"=hex:02,15,40,a0,10,1e,b8,23,00,8e,d8,8b,0e,14,
07,81,e1,00,02,1f,c3
"Change1"=hex:01,1d,50,48,0c,55,8b,ec,b8,00,00,9c,59,
81,e1,00,02,55,8b,ec,b8,\
  00,00,e8,e7,57,90,90,90

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session Manager\AppPatches\MYST]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session
Manager\AppPatches\MYST\ff060102423bab000407102e0600]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session
Manager\AppPatches\MYST\ff060102423bab000407102e0600\
1]
"Add1"=hex:02,15,40,ab,10,1e,b8,23,00,8e,d8,8b,0e,14,
07,81,e1,00,02,1f,c3
"Change1"=hex:01,1d,50,49,0c,55,8b,ec,b8,00,00,9c,59,
81,e1,00,02,55,8b,ec,b8,\
  00,00,e8,e7,61,90,90,90

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session Manager\AppPatches\PALED40]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session
Manager\AppPatches\PALED40\ff060102420032000407401b01
00]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session
Manager\AppPatches\PALED40\ff060102420032000407401b01
00\1]
"Change1"=hex:01,07,b7,21,01,d8,0c

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session Manager\AppPatches\USA]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session
Manager\AppPatches\USA\ff06010242059b00040710780600]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session
Manager\AppPatches\USA\ff06010242059b00040710780600\1
]
"Change1"=hex:01,1d,95,44,0c,55,8b,ec,b8,00,00,9c,59,
81,e1,00,02,55,8b,ec,b8,\
  00,00,e8,67,56,90,90,90
"Change2"=hex:01,25,05,9b,10,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,\
  1e,b8,23,00,8e,d8,8b,0e,14,07,81,e1,00,02,1f,c3

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session Manager\AppPatches\VB]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session
Manager\AppPatches\VB\ff060102ec353f00040780c81300]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session
Manager\AppPatches\VB\ff060102ec353f00040780c81300\12
]
"Change1"=hex:01,11,1b,03,06,81,3e,ba,31,34,03,81,3e,
ba,31,09,03

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session Manager\AppPatches\VB40016]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session
Manager\AppPatches\VB40016\ff0702021401ee3e000407d046
0e00]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session
Manager\AppPatches\VB40016\ff0702021401ee3e000407d046
0e00\16]
"Change1"=hex:01,11,6d,2a,06,81,3e,6e,36,34,03,81,3e,
6e,36,09,03

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session Manager\DOS Devices]
"PRN"="\\DosDevices\\LPT1"
"AUX"="\\DosDevices\\COM1"
"NUL"="\\Device\\Null"
"PIPE"="\\Device\\NamedPipe"
"MAILSLOT"="\\Device\\MailSlot"
"UNC"="\\Device\\Mup"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session Manager\Environment]
"ComSpec"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,
5c,73,79,73,74,65,6d,33,\
  32,5c,63,6d,64,2e,65,78,65,00
"NUMBER_OF_PROCESSORS"="4"
"OS"="Windows_NT"
"Os2LibPath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,
25,5c,73,79,73,74,65,6d,\
  33,32,5c,6f,73,32,5c,64,6c,6c,3b,00
"Path"=hex(2):43,3a,5c,6d,6b,73,6e,74,3b,43,3a,5c,57,
49,4e,4e,54,34,30,5c,73,\

79,73,74,65,6d,33,32,3b,43,3a,5c,57,49,4e,4e,54,34,30
,3b,43,3a,5c,4d,53,53,\
  51,4c,37,5c,42,49,4e,4e,00
"PROCESSOR_ARCHITECTURE"="x86"
"PROCESSOR_IDENTIFIER"="x86 Family 6 Model 7 Stepping
2, GenuineIntel"
"PROCESSOR_LEVEL"="6"
"PROCESSOR_REVISION"="0702"
"windir"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,0
0
"ROOTDIR"="C:/"
"SHELL"="C:/mksnt/sh.exe"
"HOME"="C:/"
"TMPDIR"="C:/TEMP"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session Manager\Executive]
"AdditionalCriticalWorkerThreads"=dword:00000000
"AdditionalDelayedWorkerThreads"=dword:00000000
"PriorityQuantumMatrix"=hex:f4,9a,d1,62,00,00,00,00,2
d,13,be,01

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session Manager\FileRenameOperations]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session Manager\KnownDLLs]
"DllDirectory"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,7
4,25,5c,73,79,73,74,65,6d,\
  33,32,00
"kernel32"="kernel32.dll"
"gdi32"="gdi32.dll"
"user32"="user32.dll"
"rpcrt4"="rpcrt4.dll"
"advapi32"="advapi32.dll"
"comdlg32"="comdlg32.dll"
"crtdll"="crtdll.dll"
"shell32"="shell32.dll"
"lz32"="lz32.dll"
"olecli32"="olecli32.dll"
"olesvr32"="olesvr32.dll"
"version"="version.dll"
"ole32"="ole32.dll"
"oleaut32"="oleaut32.dll"
"olecnv32"="olecnv32.dll"
"olethk32"="olethk32.dll"
"SHLWAPI"="SHLWAPI.DLL"
"WININET"="WININET.DLL"
"URLMON"="URLMON.DLL"
"NETAPI32"="NETAPI32.DLL"
"MSRATING"="MSRATING.DLL"
"URL"="URL.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session Manager\Memory Management]
"PagedPoolSize"=dword:00000000
"NonPagedPoolSize"=dword:00000000
"PagedPoolQuota"=dword:00000000
"NonPagedPoolQuota"=dword:00000000
"IoPageLockLimit"=dword:00000000
"LargeSystemCache"=dword:00000000
"PagingFiles"=hex(7):43,3a,5c,70,61,67,65,66,69,6c,65
,2e,73,79,73,20,35,30,30,\

20,35,30,30,00,44,3a,5c,70,61,67,65,66,69,6c,65,2e,73
,79,73,20,31,37,30,30,\
  20,31,37,30,30,00,00
"SystemPages"=dword:00000000
"SecondLevelDataCache"=dword:00000000
"DisablePagingExecutive"=dword:00000000
"ClearPageFileAtShutdown"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Session Manager\SubSystems]
"Required"=hex(7):44,65,62,75,67,00,57,69,6e,64,6f,77
,73,00,00
"Optional"=hex(7):4f,73,32,00,50,6f,73,69,78,00,00
"Debug"=hex(2):00
"Windows"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,
5c,73,79,73,74,65,6d,33,\

32,5c,63,73,72,73,73,2e,65,78,65,20,4f,62,6a,65,63,74
,44,69,72,65,63,74,6f,\

72,79,3d,5c,57,69,6e,64,6f,77,73,20,53,68,61,72,65,64
,53,65,63,74,69,6f,6e,\

3d,31,30,32,34,2c,33,30,37,32,20,57,69,6e,64,6f,77,73
,3d,4f,6e,20,53,75,62,\

53,79,73,74,65,6d,54,79,70,65,3d,57,69,6e,64,6f,77,73
,20,53,65,72,76,65,72,\

44,6c,6c,3d,62,61,73,65,73,72,76,2c,31,20,53,65,72,76
,65,72,44,6c,6c,3d,77,\

69,6e,73,72,76,3a,55,73,65,72,53,65,72,76,65,72,44,6c
,6c,49,6e,69,74,69,61,\

6c,69,7a,61,74,69,6f,6e,2c,33,20,53,65,72,76,65,72,44
,6c,6c,3d,77,69,6e,73,\

72,76,3a,43,6f,6e,53,65,72,76,65,72,44,6c,6c,49,6e,69
,74,69,61,6c,69,7a,61,\

74,69,6f,6e,2c,32,20,50,72,6f,66,69,6c,65,43,6f,6e,74
,72,6f,6c,3d,4f,66,66,\

20,4d,61,78,52,65,71,75,65,73,74,54,68,72,65,61,64,73
,3d,31,36,00
"Kmode"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c
,73,79,73,74,65,6d,33,32,\
  5c,77,69,6e,33,32,6b,2e,73,79,73,00
"Os2"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,7
3,79,73,74,65,6d,33,32,5c,\
  6f,73,32,73,73,2e,65,78,65,00
"Posix"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c
,73,79,73,74,65,6d,33,32,\
  5c,70,73,78,73,73,2e,65,78,65,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps\HOTDOGPRO4.EXE]
"hotdogpro4.exe"="5006"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps\ileap1-1.exe]
"ileap1-1.exe"="5001"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps\LAUNCHER.EXE]
"hatchery.exe"="5013"
"Versionhatchery.exe"=hex:00,00,01,00,01,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps\PRODINST.EXE]
"PRODINST.EXE"="1092"
"VersionPRODINST.EXE"=hex:00,00,01,00,01,00,00,00
"FlagsPRODINST.EXE"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps\setup.exe]
"money96.inf"="5004"
"Versionmoney96.inf"=hex:14,00,01,00,7b,07,00,00
"Flagsmoney96.inf"="I"
"winsetup.bin"="5014"
"Versionwinsetup.bin"=hex:00,00,04,00,57,04,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\access.exe]
"Flagsaccssuns.exe"="I"
"accssuns.exe"="5019"
"Versionaccess.exe"=hex:00,00,01,00,01,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\InterLeap.exe]
"InterLeap.exe"="5001"
"VersionInterLeap.exe"=hex:01,00,01,00,00,00,00,00
"FlagsInterLeap.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE20.exe]
"MSIE20.exe"="5003"
"FlagsMSIE20.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE30.exe]
"MSIE30.exe"="5003"
"FlagsMSIE30.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE301-AXP.exe]
"MSIE301-AXP.exe"="5003"
"FlagsMSIE301-AXP.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE301.exe]
"MSIE301.exe"="5003"
"FlagsMSIE301.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE301M95.exe]
"MSIE301M95.exe"="5003"
"FlagsMSIE301M95.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE301MNT.exe]
"MSIE301MNT.exe"="5003"
"FlagsMSIE301MNT.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE301r.exe]
"MSIE301r.exe"="5003"
"FlagsMSIE301r.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE301US.exe]
"MSIE301US.exe"="5003"
"FlagsMSIE301US.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE302-AXP.exe]
"MSIE302-AXP.exe"="5003"
"FlagsMSIE302-AXP.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE302.exe]
"MSIE302.exe"="5003"
"FlagsMSIE302.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE302M95.exe]
"MSIE302M95.exe"="5003"
"FlagsMSIE302M95.exe"="I"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE302MNT.exe]
"MSIE302MNT.exe"="5003"
"FlagsMSIE302MNT.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE302R.exe]
"MSIE302R.exe"="5003"
"FlagsMSIE302R.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE302US.exe]
"MSIE302US.exe"="5003"
"FlagsMSIE302US.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE30M.exe]
"MSIE30M.exe"="5003"
"FlagsMSIE30M.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE30r.exe]
"MSIE30r.exe"="5003"
"FlagsMSIE30r.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\MSIE30US.exe]
"MSIE30US.exe"="5003"
"FlagsMSIE30US.exe"="I"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\NT4SP3_I.EXE]
"nt4sp3_i.exe"="5015"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\OLEUPD.EXE]
"oleupd.exe"="5011"
"Versionoleupd.exe"=hex:47,00,04,00,00,00,9a,01

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\sp1unins.exe]
"sp1unins.exe"="5008"
"Flagssp1unins.exe"="I"
"Versionsp1unins.exe"=hex:00,00,04,00,b6,03,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\symapudo.exe]
"symapudo.exe"="5007"
"Flagssymapudo.exe"="I"
"Versionsymapudo.exe"=hex:01,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SessionManager\CheckBadApps400\UPDATE.EXE]
"krnl386.exe"="5015"
"Versionkrnl386.exe"=hex:00,00,04,00,04,00,65,05

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Setup]
"pointer"="msps2"
"video"="VGA"
"keyboard"="STANDARD"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SystemResources]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SystemResources\AssignmentOrdering]
"Isa"="PCFlat"
"Eisa"="PCFlat"
"MCA"="PCFlat"
"PCMCIA"="PCFlat"
"PCFlat"=hex(a):a8,04,00,00,00,00,00,00,00,00,00,00,0
0,00,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,01,00,00,00,01,00,01,00,24,00,00
,00,00,01,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,05,00,00,00,00,00,00
,ff,ff,00,00,00,00,00,\

00,08,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,40
,01,00,00,00,00,00,00,\

7f,01,00,00,00,00,00,00,08,01,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

02,00,00,00,00,00,00,ff,02,00,00,00,00,00,00,08,01,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,03,00,00,00,00,00,00,6f,03,00,00
,00,00,00,00,08,01,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,78,03,00,00,00
,00,00,00,7a,03,00,00,\

00,00,00,00,08,01,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,e8,02,00,00,00,\

00,00,00,ff,02,00,00,00,00,00,00,08,01,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,f0,01,00,00,00,00,00,00,f8,01,00,00,00,00,00,00
,08,01,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,b0,03,00,00,00,00,00,00,cf
,03,00,00,00,00,00,00,\

08,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,e8,03
,00,00,00,00,00,00,ff,\

03,00,00,00,00,00,00,08,01,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,ce,01,\

00,00,00,00,00,00,cf,01,00,00,00,00,00,00,08,01,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,01,00,00,00,00,00,00,ff,03,00,00,00
,00,00,00,00,02,00,00,\

00,00,00,00,0f,00,00,00,0f,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,08,02,00,00,00,00,00,00,0d,00,00,00,0d,00,00
,00,00,00,00,00,00,00,\
00,00,00,00,00,00,00,00,00,00,08,02,00,00,00,00,00,00
,0c,00,00,00,0c,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,08
,02,00,00,00,00,00,00,\

09,00,00,00,09,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,08,\

02,00,00,00,00,00,00,08,00,00,00,08,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,08,02,00,00,00,00,00,00,07,00,00,00
,07,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,08,02,00,00,00
,00,00,00,0b,00,00,00,\

0b,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,08,02,00,00,00,\

00,00,00,0a,00,00,00,0a,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,08,02,00,00,00,00,00,00,02,00,00,00,02,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,08,02,00,00,00,00,00,00,05
,00,00,00,05,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,08,02
,00,00,00,00,00,00,03,\

00,00,00,03,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,08,02,\

00,00,00,00,00,00,04,00,00,00,04,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,08,02,00,00,00,00,00,00,0e,00,00,00,0e
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,08,02,00,00,00,00
,00,00,06,00,00,00,06,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,08,02,00,00,00,00,\

00,00,0c,00,00,00,0c,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,08,02,00,00,00,00,00,00,01,00,00,00,01,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,08,02,00,00,00,00,00,00,00,00
,00,00,0f,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,03,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,10,00,00,00,00,00,ff,ff,af,ff
,00,00,00,00,08,03,00,\
```

```
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0f,00,00
,00,00,00,ff,ff,0f,00,\

00,00,00,00,08,03,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,08,00,00,\

00,00,00,ff,ff,0b,00,00,00,00,00,08,03,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,08,00,00,00,00,00,ff,ff,0f,00,00,00,00,00
,08,03,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,08,00,00,00,00,00,ff
,ff,f7,ff,00,00,00,00,\

00,04,00,00,00,00,00,00,06,00,00,00,0f,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,08,04,00,00,00,00,00,00,03,00,00
,00,04,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,08,04,00,00
,00,00,00,00,00,00,00,\

00,0f,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00
"PCI"=hex(a):e8,08,00,00,00,00,00,00,00,00,00,00,00,0
0,00,00,00,00,00,00,00,00,\

00,00,00,00,00,00,01,00,00,00,01,00,01,00,46,00,00,00
,00,01,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,fc,00,00,00,00,00,00,ff
,fc,00,00,00,00,00,00,\

08,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,f8
,00,00,00,00,00,00,ff,\

f9,00,00,00,00,00,00,08,01,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,f4,\

00,00,00,00,00,00,ff,f4,00,00,00,00,00,00,08,01,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,f0,00,00,00,00,00,00,ff,f0,00,00,00
,00,00,00,08,01,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,ec,00,00,00,00
,00,00,ff,ec,00,00,00,\

00,00,00,08,01,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,e8,00,00,00,00,\

00,00,ff,e9,00,00,00,00,00,00,08,01,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,e4,00,00,00,00,00,00,ff,e4,00,00,00,00,00,00,08
,01,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,e0,00,00,00,00,00,00,ff,e0
,00,00,00,00,00,00,08,\
```

```
01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,dc,00
,00,00,00,00,00,ff,dc,\

00,00,00,00,00,00,08,01,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,d8,00,\

00,00,00,00,00,ff,d9,00,00,00,00,00,00,08,01,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,d4,00,00,00,00,00,00,ff,d4,00,00,00,00
,00,00,08,01,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,d0,00,00,00,00,00
,00,ff,d0,00,00,00,00,\

00,00,08,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,cc,00,00,00,00,00,\

00,ff,cc,00,00,00,00,00,00,08,01,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,c8,00,00,00,00,00,00,ff,c9,00,00,00,00,00,00,08,01
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,c4,00,00,00,00,00,00,ff,c4,00
,00,00,00,00,00,08,01,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,c0,00,00
,00,00,00,00,ff,c0,00,\

00,00,00,00,00,08,01,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,bc,00,00,\

00,00,00,00,ff,bc,00,00,00,00,00,00,08,01,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,b8,00,00,00,00,00,00,ff,b9,00,00,00,00,00
,00,08,01,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,b4,00,00,00,00,00,00
,ff,b4,00,00,00,00,00,\

00,08,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,b0,00,00,00,00,00,00,\

ff,b0,00,00,00,00,00,00,08,01,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

ac,00,00,00,00,00,00,ff,ac,00,00,00,00,00,00,08,01,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,a8,00,00,00,00,00,00,ff,a9,00,00
,00,00,00,00,08,01,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,a4,00,00,00
,00,00,00,ff,a4,00,00,\

00,00,00,00,08,01,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,a0,00,00,00,\

00,00,00,ff,a0,00,00,00,00,00,00,08,01,00,00,00,00,00
,00,00,00,00,00,00,00,\
00,00,00,9c,00,00,00,00,00,00,ff,9c,00,00,00,00,00,00
,08,01,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,98,00,00,00,00,00,00,ff
,99,00,00,00,00,00,00,\

08,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,94
,00,00,00,00,00,00,ff,\

94,00,00,00,00,00,00,08,01,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,90,\

00,00,00,00,00,00,ff,90,00,00,00,00,00,00,08,01,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,8c,00,00,00,00,00,00,ff,8c,00,00,00
,00,00,00,08,01,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,88,00,00,00,00
,00,00,ff,89,00,00,00,\

00,00,00,08,01,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,84,00,00,00,00,\

00,00,ff,84,00,00,00,00,00,00,08,01,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,80,00,00,00,00,00,00,ff,80,00,00,00,00,00,00,08
,01,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,7c,00,00,00,00,00,00,ff,7c
,00,00,00,00,00,00,08,\

01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,78,00
,00,00,00,00,00,ff,79,\

00,00,00,00,00,00,08,01,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,74,00,\

00,00,00,00,00,ff,74,00,00,00,00,00,00,08,01,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,70,00,00,00,00,00,00,ff,70,00,00,00,00
,00,00,08,01,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,6c,00,00,00,00,00
,00,ff,6c,00,00,00,00,\

00,00,08,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,68,00,00,00,00,00,\

00,ff,69,00,00,00,00,00,00,08,01,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,64,00,00,00,00,00,00,ff,64,00,00,00,00,00,00,08,01
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,60,00,00,00,00,00,00,ff,60,00
,00,00,00,00,00,08,01,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,5c,00,00
,00,00,00,00,ff,5c,00,\
```

```
00,00,00,00,00,08,01,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,58,00,00,\

00,00,00,00,ff,59,00,00,00,00,00,00,08,01,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,54,00,00,00,00,00,00,ff,54,00,00,00,00,00
,00,08,01,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,50,00,00,00,00,00,00
,ff,50,00,00,00,00,00,\

00,08,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,4c,00,00,00,00,00,00,\

ff,4c,00,00,00,00,00,00,08,01,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

48,00,00,00,00,00,00,ff,49,00,00,00,00,00,00,08,01,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,44,00,00,00,00,00,00,ff,44,00,00
,00,00,00,00,08,01,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,40,00,00,00
,00,00,00,ff,40,00,00,\

00,00,00,00,08,01,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,3c,00,00,00,\

00,00,00,ff,3c,00,00,00,00,00,00,08,01,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,38,00,00,00,00,00,00,ff,39,00,00,00,00,00,00
,08,01,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,34,00,00,00,00,00,00,ff
,34,00,00,00,00,00,00,\

08,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,30
,00,00,00,00,00,00,ff,\

30,00,00,00,00,00,00,08,01,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,2c,\

00,00,00,00,00,00,ff,1c,00,00,00,00,00,00,08,01,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,28,00,00,00,00,00,00,ff,19,00,00,00
,00,00,00,08,01,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,24,00,00,00,00
,00,00,ff,14,00,00,00,\

00,00,00,08,01,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,20,00,00,00,00,\

00,00,ff,10,00,00,00,00,00,00,08,01,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,1c,00,00,00,00,00,00,ff,1c,00,00,00,00,00,00,08
,01,00,00,00,00,00,00,\
```

```
00,00,00,00,00,00,00,00,00,18,00,00,00,00,00,00,ff,19
,00,00,00,00,00,00,08,\

01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,14,00
,00,00,00,00,00,ff,14,\

00,00,00,00,00,00,08,01,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,10,00,\

00,00,00,00,00,ff,10,00,00,00,00,00,00,08,01,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,0c,00,00,00,00,00,00,ff,0c,00,00,00,00
,00,00,08,01,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,08,00,00,00,00,00
,00,ff,09,00,00,00,00,\

00,00,08,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,05,00,00,00,00,00,\

00,ff,ff,ff,ff,00,00,00,00,00,02,00,00,00,00,00,00,00
,00,00,00,ff,ff,ff,ff,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,03
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,10,00,00,00,00,00,ff,ff,af
,ff,00,00,00,00,08,03,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0f,00
,00,00,00,00,ff,ff,0f,\

00,00,00,00,00,08,03,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,08,00,\

00,00,00,00,ff,ff,0b,00,00,00,00,00,08,03,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,08,00,00,00,00,00,ff,ff,0f,00,00,00,00
,00,08,03,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,08,00,00,00,00,00
,ff,ff,f7,ff,00,00,00,\

00,00,04,00,00,00,00,00,00,06,00,00,00,ff,00,00,00,00
,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SystemResources\BusValues]
"Internal"=hex:00,00,00,00,00,00,00,00
"Isa"=hex:01,00,00,00,00,00,00,00
"Eisa"=hex:02,00,00,00,01,00,00,00
"MCA"=hex:03,00,00,00,01,00,00,00
"TurboChannel"=hex:04,00,00,00,00,00,00,00
"PCI"=hex:05,00,00,00,01,00,00,00
"VME"=hex:06,00,00,00,00,00,00,00
"NuBus"=hex:07,00,00,00,00,00,00,00
"PCMCIA"=hex:08,00,00,00,01,00,00,00
"CBus"=hex:09,00,00,00,00,00,00,00
"MPI"=hex:0a,00,00,00,00,00,00,00
```

```
"MPSA"=hex:0b,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
SystemResources\ReservedResources]
"Isa"=hex(8):01,00,00,00,01,00,00,00,00,00,00,00,00,0
0,00,00,28,00,00,00,01,01,\

00,00,00,00,00,00,00,00,00,00,00,01,00,00,01,03,00,00
,e8,42,00,00,00,00,00,\

00,08,00,00,00,01,03,00,00,e8,4a,00,00,00,00,00,00,08
,00,00,00,01,03,00,00,\

e8,82,00,00,00,00,00,00,08,00,00,00,01,03,00,00,e8,86
,00,00,00,00,00,00,08,\

00,00,00,01,03,00,00,e8,8a,00,00,00,00,00,00,08,00,00
,00,01,03,00,00,e8,8e,\

00,00,00,00,00,00,08,00,00,00,01,03,00,00,e8,92,00,00
,00,00,00,00,08,00,00,\

00,01,03,00,00,e8,96,00,00,00,00,00,00,08,00,00,00,01
,03,00,00,e8,9a,00,00,\

00,00,00,00,08,00,00,00,01,03,00,00,e8,9e,00,00,00,00
,00,00,08,00,00,00,01,\

03,00,00,e8,a2,00,00,00,00,00,00,08,00,00,00,01,03,00
,00,e8,a6,00,00,00,00,\

00,00,08,00,00,00,01,03,00,00,e8,aa,00,00,00,00,00,00
,08,00,00,00,01,03,00,\

00,e8,ae,00,00,00,00,00,00,08,00,00,00,01,03,00,00,e8
,b6,00,00,00,00,00,00,\

08,00,00,00,01,03,00,00,e8,ba,00,00,00,00,00,00,08,00
,00,00,01,03,00,00,e8,\

be,00,00,00,00,00,00,08,00,00,00,01,03,00,00,e8,c2,00
,00,00,00,00,00,08,00,\

00,00,01,03,00,00,e8,c6,00,00,00,00,00,00,08,00,00,00
,01,03,00,00,e8,ca,00,\

00,00,00,00,00,08,00,00,00,01,03,00,00,e8,ce,00,00,00
,00,00,00,08,00,00,00,\

01,03,00,00,e8,d2,00,00,00,00,00,00,08,00,00,00,01,03
,00,00,e8,d6,00,00,00,\

00,00,00,08,00,00,00,01,03,00,00,e8,da,00,00,00,00,00
,00,08,00,00,00,01,03,\

00,00,e8,de,00,00,00,00,00,00,08,00,00,00,01,03,00,00
,e8,e2,00,00,00,00,00,\

00,08,00,00,00,01,03,00,00,e8,e6,00,00,00,00,00,00,08
,00,00,00,01,03,00,00,\

e8,ea,00,00,00,00,00,00,08,00,00,00,01,03,00,00,e8,ee
,00,00,00,00,00,00,08,\
```

```
00,00,00,01,03,00,00,ee,f6,00,00,00,00,00,00,02,00,00
,00,01,03,00,00,ee,fa,\

00,00,00,00,00,00,02,00,00,00,01,03,00,00,ee,fe,00,00
,00,00,00,00,02,00,00,\

00,02,03,00,00,03,00,00,00,03,00,00,00,ff,ff,ff,ff,02
,03,00,00,04,00,00,00,\

04,00,00,00,ff,ff,ff,ff,02,03,00,00,0e,00,00,00,0e,00
,00,00,ff,ff,ff,ff,02,\

03,00,00,06,00,00,00,06,00,00,00,ff,ff,ff,ff,02,03,00
,00,0c,00,00,00,0c,00,\

00,00,ff,ff,ff,ff,02,03,00,00,01,00,00,00,01,00,00,00
,ff,ff,ff,ff,03,03,00,\
  00,ff,ff,bf,ff,00,00,00,00,00,00,40,00
"HAL_PCI_1"=hex(8):01,00,00,00,05,00,00,00,01,00,00,0
0,00,00,00,00,7e,00,00,00,\

01,01,01,00,b0,07,00,00,00,00,00,00,0b,00,00,00,01,01
,01,00,c0,07,00,00,00,\

00,00,00,1f,00,00,00,01,01,01,00,b0,0b,00,00,00,00,00
,00,0b,00,00,00,01,01,\

01,00,c0,0b,00,00,00,00,00,00,1f,00,00,00,01,01,01,00
,b0,0f,00,00,00,00,00,\

00,0b,00,00,00,01,01,01,00,c0,0f,00,00,00,00,00,00,1f
,00,00,00,01,01,01,00,\

b0,13,00,00,00,00,00,00,0b,00,00,00,01,01,01,00,c0,13
,00,00,00,00,00,00,1f,\

00,00,00,01,01,01,00,b0,17,00,00,00,00,00,00,0b,00,00
,00,01,01,01,00,c0,17,\

00,00,00,00,00,00,1f,00,00,00,01,01,01,00,b0,1b,00,00
,00,00,00,00,0b,00,00,\

00,01,01,01,00,c0,1b,00,00,00,00,00,00,1f,00,00,00,01
,01,01,00,b0,1f,00,00,\

00,00,00,00,0b,00,00,00,01,01,01,00,c0,1f,00,00,00,00
,00,00,1f,00,00,00,01,\

01,01,00,b0,23,00,00,00,00,00,00,0b,00,00,00,01,01,01
,00,c0,23,00,00,00,00,\

00,00,1f,00,00,00,01,01,01,00,b0,27,00,00,00,00,00,00
,0b,00,00,00,01,01,01,\

00,c0,27,00,00,00,00,00,00,1f,00,00,00,01,01,01,00,b0
,2b,00,00,00,00,00,00,\

0b,00,00,00,01,01,01,00,c0,2b,00,00,00,00,00,00,1f,00
,00,00,01,01,01,00,b0,\

2f,00,00,00,00,00,00,0b,00,00,00,01,01,01,00,c0,2f,00
,00,00,00,00,00,1f,00,\
```

```
00,00,01,01,01,00,b0,33,00,00,00,00,00,00,0b,00,00,00
,01,01,01,00,c0,33,00,\

00,00,00,00,00,1f,00,00,00,01,01,01,00,b0,37,00,00,00
,00,00,00,0b,00,00,00,\

01,01,01,00,c0,37,00,00,00,00,00,00,1f,00,00,00,01,01
,01,00,b0,3b,00,00,00,\

00,00,00,0b,00,00,00,01,01,01,00,c0,3b,00,00,00,00,00
,00,1f,00,00,00,01,01,\

01,00,b0,3f,00,00,00,00,00,00,0b,00,00,00,01,01,01,00
,c0,3f,00,00,00,00,00,\

00,1f,00,00,00,01,01,01,00,b0,43,00,00,00,00,00,00,0b
,00,00,00,01,01,01,00,\

c0,43,00,00,00,00,00,00,1f,00,00,00,01,01,01,00,b0,47
,00,00,00,00,00,00,0b,\

00,00,00,01,01,01,00,c0,47,00,00,00,00,00,00,1f,00,00
,00,01,01,01,00,b0,4b,\

00,00,00,00,00,00,0b,00,00,00,01,01,01,00,c0,4b,00,00
,00,00,00,00,1f,00,00,\

00,01,01,01,00,b0,4f,00,00,00,00,00,00,0b,00,00,00,01
,01,01,00,c0,4f,00,00,\

00,00,00,00,1f,00,00,00,01,01,01,00,b0,53,00,00,00,00
,00,00,0b,00,00,00,01,\

01,01,00,c0,53,00,00,00,00,00,00,1f,00,00,00,01,01,01
,00,b0,57,00,00,00,00,\

00,00,0b,00,00,00,01,01,01,00,c0,57,00,00,00,00,00,00
,1f,00,00,00,01,01,01,\

00,b0,5b,00,00,00,00,00,00,0b,00,00,00,01,01,01,00,c0
,5b,00,00,00,00,00,00,\

1f,00,00,00,01,01,01,00,b0,5f,00,00,00,00,00,00,0b,00
,00,00,01,01,01,00,c0,\

5f,00,00,00,00,00,00,1f,00,00,00,01,01,01,00,b0,63,00
,00,00,00,00,00,0b,00,\

00,00,01,01,01,00,c0,63,00,00,00,00,00,00,1f,00,00,00
,01,01,01,00,b0,67,00,\

00,00,00,00,00,0b,00,00,00,01,01,01,00,c0,67,00,00,00
,00,00,00,1f,00,00,00,\

01,01,01,00,b0,6b,00,00,00,00,00,00,0b,00,00,00,01,01
,01,00,c0,6b,00,00,00,\

00,00,00,1f,00,00,00,01,01,01,00,b0,6f,00,00,00,00,00
,00,0b,00,00,00,01,01,\

01,00,c0,6f,00,00,00,00,00,00,1f,00,00,00,01,01,01,00
,b0,73,00,00,00,00,00,\
00,0b,00,00,00,01,01,01,00,c0,73,00,00,00,00,00,00,1f
,00,00,00,01,01,01,00,\

b0,77,00,00,00,00,00,00,0b,00,00,00,01,01,01,00,c0,77
,00,00,00,00,00,00,1f,\

00,00,00,01,01,01,00,b0,7b,00,00,00,00,00,00,0b,00,00
,00,01,01,01,00,c0,7b,\

00,00,00,00,00,00,1f,00,00,00,01,01,01,00,b0,7f,00,00
,00,00,00,00,0b,00,00,\

00,01,01,01,00,c0,7f,00,00,00,00,00,00,1f,00,00,00,01
,01,01,00,b0,83,00,00,\

00,00,00,00,0b,00,00,00,01,01,01,00,c0,83,00,00,00,00
,00,00,1f,00,00,00,01,\

01,01,00,b0,87,00,00,00,00,00,00,0b,00,00,00,01,01,01
,00,c0,87,00,00,00,00,\

00,00,1f,00,00,00,01,01,01,00,b0,8b,00,00,00,00,00,00
,0b,00,00,00,01,01,01,\

00,c0,8b,00,00,00,00,00,00,1f,00,00,00,01,01,01,00,b0
,8f,00,00,00,00,00,00,\

0b,00,00,00,01,01,01,00,c0,8f,00,00,00,00,00,00,1f,00
,00,00,01,01,01,00,b0,\

93,00,00,00,00,00,00,0b,00,00,00,01,01,01,00,c0,93,00
,00,00,00,00,00,1f,00,\

00,00,01,01,01,00,b0,97,00,00,00,00,00,00,0b,00,00,00
,01,01,01,00,c0,97,00,\

00,00,00,00,00,1f,00,00,00,01,01,01,00,b0,9b,00,00,00
,00,00,00,0b,00,00,00,\

01,01,01,00,c0,9b,00,00,00,00,00,00,1f,00,00,00,01,01
,01,00,b0,9f,00,00,00,\

00,00,00,0b,00,00,00,01,01,01,00,c0,9f,00,00,00,00,00
,00,1f,00,00,00,01,01,\

01,00,b0,a3,00,00,00,00,00,00,0b,00,00,00,01,01,01,00
,c0,a3,00,00,00,00,00,\

00,1f,00,00,00,01,01,01,00,b0,a7,00,00,00,00,00,00,0b
,00,00,00,01,01,01,00,\

c0,a7,00,00,00,00,00,00,1f,00,00,00,01,01,01,00,b0,ab
,00,00,00,00,00,00,0b,\

00,00,00,01,01,01,00,c0,ab,00,00,00,00,00,00,1f,00,00
,00,01,01,01,00,b0,af,\

00,00,00,00,00,00,0b,00,00,00,01,01,01,00,c0,af,00,00
,00,00,00,00,1f,00,00,\

00,01,01,01,00,b0,b3,00,00,00,00,00,00,0b,00,00,00,01
,01,01,00,c0,b3,00,00,\

00,00,00,00,1f,00,00,00,01,01,01,00,b0,b7,00,00,00,00
,00,00,0b,00,00,00,01,\

01,01,00,c0,b7,00,00,00,00,00,00,1f,00,00,00,01,01,01
,00,b0,bb,00,00,00,00,\

00,00,0b,00,00,00,01,01,01,00,c0,bb,00,00,00,00,00,00
,1f,00,00,00,01,01,01,\

00,b0,bf,00,00,00,00,00,00,0b,00,00,00,01,01,01,00,c0
,bf,00,00,00,00,00,00,\

1f,00,00,00,01,01,01,00,b0,c3,00,00,00,00,00,00,0b,00
,00,00,01,01,01,00,c0,\

c3,00,00,00,00,00,00,1f,00,00,00,01,01,01,00,b0,c7,00
,00,00,00,00,00,0b,00,\

00,00,01,01,01,00,c0,c7,00,00,00,00,00,00,1f,00,00,00
,01,01,01,00,b0,cb,00,\

00,00,00,00,00,0b,00,00,00,01,01,01,00,c0,cb,00,00,00
,00,00,00,1f,00,00,00,\

01,01,01,00,b0,cf,00,00,00,00,00,00,0b,00,00,00,01,01
,01,00,c0,cf,00,00,00,\

00,00,00,1f,00,00,00,01,01,01,00,b0,d3,00,00,00,00,00
,00,0b,00,00,00,01,01,\

01,00,c0,d3,00,00,00,00,00,00,1f,00,00,00,01,01,01,00
,b0,d7,00,00,00,00,00,\

00,0b,00,00,00,01,01,01,00,c0,d7,00,00,00,00,00,00,1f
,00,00,00,01,01,01,00,\

b0,db,00,00,00,00,00,00,0b,00,00,00,01,01,01,00,c0,db
,00,00,00,00,00,00,1f,\

00,00,00,01,01,01,00,b0,df,00,00,00,00,00,00,0b,00,00
,00,01,01,01,00,c0,df,\

00,00,00,00,00,00,1f,00,00,00,01,01,01,00,b0,e3,00,00
,00,00,00,00,0b,00,00,\

00,01,01,01,00,c0,e3,00,00,00,00,00,00,1f,00,00,00,01
,01,01,00,b0,e7,00,00,\

00,00,00,00,0b,00,00,00,01,01,01,00,c0,e7,00,00,00,00
,00,00,1f,00,00,00,01,\

01,01,00,b0,eb,00,00,00,00,00,00,0b,00,00,00,01,01,01
,00,c0,eb,00,00,00,00,\

00,00,1f,00,00,00,01,01,01,00,b0,ef,00,00,00,00,00,00
,0b,00,00,00,01,01,01,\

00,c0,ef,00,00,00,00,00,00,1f,00,00,00,01,01,01,00,b0
,f3,00,00,00,00,00,00,\

0b,00,00,00,01,01,01,00,c0,f3,00,00,00,00,00,00,1f,00
,00,00,01,01,01,00,b0,\

f7,00,00,00,00,00,00,0b,00,00,00,01,01,01,00,c0,f7,00
,00,00,00,00,00,1f,00,\

00,00,01,01,01,00,b0,fb,00,00,00,00,00,00,0b,00,00,00
,01,01,01,00,c0,fb,00,\

00,00,00,00,00,1f,00,00,00,01,01,01,00,b0,ff,00,00,00
,00,00,00,0b,00,00,00,\
  01,01,01,00,c0,ff,00,00,00,00,00,00,1f,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
TimeZoneInformation]
"Bias"=dword:00000168
"StandardName"="Central Standard Time"
"StandardBias"=dword:00000000
"StandardStart"=hex:00,00,0a,00,05,00,02,00,00,00,00,
00,00,00,00,00
"DaylightName"="Central Daylight Time"
"DaylightBias"=dword:ffffffc4
"DaylightStart"=hex:00,00,04,00,01,00,02,00,00,00,00,
00,00,00,00,00
"ActiveTimeBias"=dword:0000012c

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Update]
"UpdateMode"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
VirtualDeviceDrivers]
"VDD"=hex(7):00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
Windows]
"Directory"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,00
"SystemDirectory"=hex(2):25,53,79,73,74,65,6d,52,6f,6
f,74,25,5c,73,79,73,74,65,\
  6d,33,32,00
"ErrorMode"=dword:00000000
"NoInteractiveServices"=dword:00000000
"CSDVersion"=dword:00000400
"ShutdownTime"=hex:d1,3c,92,87,f1,b0,be,01

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
WOW]
"cmdline"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,
5c,73,79,73,74,65,6d,33,\
  32,5c,6e,74,76,64,6d,2e,65,78,65,00
"size"="0"
"wowcmdline"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,
25,5c,73,79,73,74,65,6d,\

33,32,5c,6e,74,76,64,6d,2e,65,78,65,20,2d,61,20,25,53
,79,73,74,65,6d,52,6f,\

6f,74,25,5c,73,79,73,74,65,6d,33,32,5c,6b,72,6e,6c,33
,38,36,00
"wowsize"="16"
"DefaultSeparateVDM"="no"
"LPT_timeout"="15"
"KnownDLLs"="comm.drv commdlg.dll ctl3dv2.dll
ddeml.dll keyboard.drv lanman.drv mapi.dll
```

```
mmsystem.dll mouse.drv netapi.dll olecli.dll
olesvr.dll pmspl.dll shell.dll sound.drv system.drv
toolhelp.dll vga.drv wfwnet.drv win87em.dll
winoldap.mod winsock.dll winspool.exe wowdeb.exe
timer.drv rasapi16.dll compobj.dll storage.dll
ole2.dll ole2disp.dll ole2nls.dll typelib.dll
msvideo.dll avifile.dll msacm.dll mciavi.drv
mciseq.drv mciwave.drv progman.exe avicap.dll"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\
hivelist]
"\\REGISTRY\\MACHINE\\HARDWARE"=""
"\\REGISTRY\\MACHINE\\SECURITY"="\\Device\\Harddisk20
\\Partition2\\WINNT40\\System32\\Config\\SECURITY"
"\\REGISTRY\\MACHINE\\SOFTWARE"="\\Device\\Harddisk20
\\Partition2\\WINNT40\\System32\\Config\\SOFTWARE"
"\\REGISTRY\\MACHINE\\SYSTEM"="\\Device\\Harddisk20\\
Partition2\\WINNT40\\System32\\Config\\SYSTEM"
"\\REGISTRY\\USER\\.DEFAULT"="\\Device\\Harddisk20\\P
artition2\\WINNT40\\System32\\Config\\DEFAULT"
"\\REGISTRY\\MACHINE\\SAM"="\\Device\\Harddisk20\\Par
tition2\\WINNT40\\System32\\Config\\SAM"
"\\REGISTRY\\USER\\S-1-5-21-2141264861-564274737-
619646970-
500"="\\Device\\Harddisk20\\Partition2\\WINNT40\\Prof
iles\\Administrator\\ntuser.dat"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\HTR
EE]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\HTR
EE\ROOT]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\HTR
EE\ROOT\0]
"FoundAtEnum"=dword:00000001
"AttachedComponents"=hex(7):52,6f,6f,74,5c,4c,45,47,4
1,43,59,5f,43,50,51,41,52,\

52,41,59,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41
,43,59,5f,41,54,41,50,\

49,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59
,5f,53,59,4d,43,38,31,\

30,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59
,5f,44,49,53,4b,5c,30,\

30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,46,41
,53,54,46,41,54,5c,30,\

30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,46,4c
,4f,50,50,59,5c,30,30,\

30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,43,44,52
,4f,4d,5c,30,30,30,30,\

00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,46,53,5f,52,45
,43,5c,30,30,30,30,00,\

52,6f,6f,74,5c,4c,45,47,41,43,59,5f,4e,55,4c,4c,5c,30
,30,30,30,00,52,6f,6f,\

74,5c,4c,45,47,41,43,59,5f,4b,53,45,43,44,44,5c,30,30
,30,30,00,52,6f,6f,74,\

5c,4c,45,47,41,43,59,5f,42,45,45,50,5c,30,30,30,30,00
,52,6f,6f,74,5c,4c,45,\

47,41,43,59,5f,4d,4f,55,43,4c,41,53,53,5c,30,30,30,30
,00,52,6f,6f,74,5c,4c,\

45,47,41,43,59,5f,4b,42,44,43,4c,41,53,53,5c,30,30,30
,30,00,52,6f,6f,74,5c,\

4c,45,47,41,43,59,5f,56,47,41,53,41,56,45,5c,30,30,30
,30,00,52,6f,6f,74,5c,\

4c,45,47,41,43,59,5f,4d,53,46,53,5c,30,30,30,30,00,52
,6f,6f,74,5c,4c,45,47,\

41,43,59,5f,4e,50,46,53,5c,30,30,30,30,00,52,6f,6f,74
,5c,4c,45,47,41,43,59,\

5f,4e,44,49,53,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45
,47,41,43,59,5f,4e,54,\

46,53,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43
,59,5f,50,41,52,41,4c,\

4c,45,4c,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41
,43,59,5f,50,41,52,50,\

4f,52,54,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41
,43,59,5f,50,41,52,56,\

44,4d,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43
,59,5f,52,50,43,53,53,\

5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f
,53,45,52,49,41,4c,5c,\

30,30,30,30,00,52,6f,6f,74,5c,2a,50,4e,50,30,33,30,62
,5c,31,5f,31,35,5f,32,\

32,5f,30,5f,33,32,5f,30,00,52,6f,6f,74,5c,2a,50,4e,50
,30,46,30,33,5c,31,5f,\

31,35,5f,32,31,5f,30,5f,33,31,5f,30,00,52,6f,6f,74,5c
,4c,45,47,41,43,59,5f,\

43,44,46,53,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47
,41,43,59,5f,4e,45,54,\

44,45,54,45,43,54,5c,30,30,30,30,00,52,6f,6f,74,5c,4c
,45,47,41,43,59,5f,53,\

50,4f,4f,4c,45,52,5c,30,30,30,30,00,52,6f,6f,74,5c,4c
,45,47,41,43,59,5f,54,\

43,50,49,50,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47
,41,43,59,5f,4c,41,4e,\

4d,41,4e,57,4f,52,4b,53,54,41,54,49,4f,4e,5c,30,30,30
,30,00,52,6f,6f,74,5c,\

4c,45,47,41,43,59,5f,52,44,52,5c,30,30,30,30,00,52,6f
,6f,74,5c,4c,45,47,41,\

43,59,5f,4d,55,50,5c,30,30,30,30,00,52,6f,6f,74,5c,4c
,45,47,41,43,59,5f,4e,\

45,54,42,54,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47
,41,43,59,5f,43,50,51,\

4e,46,33,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41
,43,59,5f,41,46,44,5c,\

30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,4e
,45,54,42,49,4f,53,5c,\

30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,4c
,4d,48,4f,53,54,53,5c,\

30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,41
,4c,45,52,54,45,52,5c,\

30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,42
,52,4f,57,53,45,52,5c,\

30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,4c
,41,4e,4d,41,4e,53,45,\

52,56,45,52,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47
,41,43,59,5f,4c,49,43,\

45,4e,53,45,53,45,52,56,49,43,45,5c,30,30,30,30,00,52
,6f,6f,74,5c,4c,45,47,\

41,43,59,5f,4d,45,53,53,45,4e,47,45,52,5c,30,30,30,30
,00,52,6f,6f,74,5c,4c,\

45,47,41,43,59,5f,53,52,56,5c,30,30,30,30,00,52,6f,6f
,74,5c,2a,50,4e,50,30,\

33,30,62,5c,31,5f,30,5f,32,32,5f,30,5f,33,32,5f,30,00
,52,6f,6f,74,5c,2a,50,\

4e,50,30,46,30,33,5c,31,5f,30,5f,32,31,5f,30,5f,33,31
,5f,30,00,52,6f,6f,74,\

5c,4c,45,47,41,43,59,5f,4e,42,46,5c,30,30,30,30,00,52
,6f,6f,74,5c,4c,45,47,\

41,43,59,5f,50,52,4f,54,45,43,54,45,44,53,54,4f,52,41
,47,45,5c,30,30,30,30,\

00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,53,59,53,4d,47
,4d,54,5c,30,30,30,30,\

00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,43,50,51,53,45
,54,55,50,5c,30,30,30,\

30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,43,50,51,33
,32,46,53,32,5c,30,30,\
30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,4e,43,52
,43,37,31,30,5c,30,30,\

30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,43,50,51
,46,57,53,32,45,5c,30,\

30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,53,43
,53,49,53,43,41,4e,5c,\

30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,4d
,53,53,51,4c,53,45,52,\

56,45,52,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41
,43,59,5f,4e,54,4c,4d,\

53,53,50,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41
,43,59,5f,44,49,53,4b,\

50,45,52,46,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47
,41,43,59,5f,41,54,49,\

5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f
,4e,31,30,30,5c,30,30,\

30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,43,50,51
,4b,44,44,5c,30,30,30,\

30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,41,54,41,4d
,41,4e,20,54,45,4c,4e,\

45,54,44,20,53,45,52,56,45,52,5c,30,30,30,30,00,52,6f
,6f,74,5c,4c,45,47,41,\

43,59,5f,41,54,41,4d,41,4e,20,52,4c,4f,47,49,4e,44,20
,53,45,52,56,45,52,5c,\

30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,41
,54,41,4d,41,4e,20,52,\

45,58,45,43,44,20,53,45,52,56,45,52,5c,30,30,30,30,00
,52,6f,6f,74,5c,4c,45,\

47,41,43,59,5f,5a,5a,5a,20,41,54,41,4d,41,4e,20,43,4f
,4d,4d,41,4e,44,20,53,\

54,41,52,54,45,52,3a,20,33,30,30,36,35,38,39,30,5c,30
,30,30,30,00,52,6f,6f,\

74,5c,4c,45,47,41,43,59,5f,5a,5a,5a,20,41,54,41,4d,41
,4e,20,43,4f,4d,4d,41,\

4e,44,20,53,54,41,52,54,45,52,3a,20,32,34,36,34,36,35
,39,33,5c,30,30,30,30,\

00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,5a,5a,5a,20,41
,54,41,4d,41,4e,20,43,\

4f,4d,4d,41,4e,44,20,53,54,41,52,54,45,52,3a,20,32,35
,37,35,34,34,30,36,5c,\

30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,5a
,5a,5a,20,41,54,41,4d,\
```

```
41,4e,20,43,4f,4d,4d,41,4e,44,20,53,54,41,52,54,45,52
,3a,20,31,39,31,35,36,\

32,34,32,31,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47
,41,43,59,5f,44,53,50,\

43,49,43,46,47,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45
,47,41,43,59,5f,5a,5a,\

5a,20,41,54,41,4d,41,4e,20,43,4f,4d,4d,41,4e,44,20,53
,54,41,52,54,45,52,3a,\

20,31,33,34,33,33,35,39,33,37,5c,30,30,30,30,00,52,6f
,6f,74,5c,4c,45,47,41,\

43,59,5f,5a,5a,5a,20,41,54,41,4d,41,4e,20,43,4f,4d,4d
,41,4e,44,20,53,54,41,\

52,54,45,52,3a,20,35,36,35,39,34,34,32,31,5c,30,30,30
,30,00,52,6f,6f,74,5c,\

4c,45,47,41,43,59,5f,5a,5a,5a,20,41,54,41,4d,41,4e,20
,43,4f,4d,4d,41,4e,44,\

20,53,54,41,52,54,45,52,3a,20,35,36,39,34,30,33,35,39
,5c,30,30,30,30,00,52,\

6f,6f,74,5c,4c,45,47,41,43,59,5f,5a,5a,5a,20,41,54,41
,4d,41,4e,20,43,4f,4d,\

4d,41,4e,44,20,53,54,41,52,54,45,52,3a,20,36,37,39,30
,30,37,38,31,5c,30,30,\

30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,5a,5a,5a
,20,41,54,41,4d,41,4e,\

20,43,4f,4d,4d,41,4e,44,20,53,54,41,52,54,45,52,3a,20
,36,38,33,37,37,38,32,\

38,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59
,5f,5a,5a,5a,20,41,54,\

41,4d,41,4e,20,43,4f,4d,4d,41,4e,44,20,53,54,41,52,54
,45,52,3a,20,36,38,35,\

32,33,30,39,33,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45
,47,41,43,59,5f,5a,5a,\

5a,20,41,54,41,4d,41,4e,20,43,4f,4d,4d,41,4e,44,20,53
,54,41,52,54,45,52,3a,\

20,35,33,39,36,38,37,36,35,5c,30,30,30,30,00,52,6f,6f
,74,5c,4c,45,47,41,43,\

59,5f,5a,5a,5a,20,41,54,41,4d,41,4e,20,43,4f,4d,4d,41
,4e,44,20,53,54,41,52,\

54,45,52,3a,20,37,30,39,31,33,38,35,39,5c,30,30,30,30
,00,52,6f,6f,74,5c,4c,\

45,47,41,43,59,5f,5a,5a,5a,20,41,54,41,4d,41,4e,20,43
,4f,4d,4d,41,4e,44,20,\
```

```
53,54,41,52,54,45,52,3a,20,38,37,30,31,37,33,31,32,5c
,30,30,30,30,00,52,6f,\

6f,74,5c,4c,45,47,41,43,59,5f,5a,5a,5a,20,41,54,41,4d
,41,4e,20,43,4f,4d,4d,\

41,4e,44,20,53,54,41,52,54,45,52,3a,20,39,33,37,33,33
,35,34,36,5c,30,30,30,\

30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,5a,5a,5a,20
,41,54,41,4d,41,4e,20,\

43,4f,4d,4d,41,4e,44,20,53,54,41,52,54,45,52,3a,20,31
,30,31,33,33,30,32,38,\

31,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59
,5f,5a,5a,5a,20,41,54,\

41,4d,41,4e,20,43,4f,4d,4d,41,4e,44,20,53,54,41,52,54
,45,52,3a,20,31,30,36,\

38,36,38,31,30,39,5c,30,30,30,30,00,52,4f,4f,54,5c,53
,43,53,49,41,44,41,50,\

54,45,52,5c,4f,45,4d,32,2e,49,4e,46,26,4d,41,43,44,49
,53,4b,00,52,6f,6f,74,\

5c,4c,45,47,41,43,59,5f,5a,5a,5a,20,41,54,41,4d,41,4e
,20,43,4f,4d,4d,41,4e,\

44,20,53,54,41,52,54,45,52,3a,20,32,31,39,33,35,33,35
,39,5c,30,30,30,30,00,\

52,6f,6f,74,5c,4c,45,47,41,43,59,5f,5a,5a,5a,20,41,54
,41,4d,41,4e,20,43,4f,\

4d,4d,41,4e,44,20,53,54,41,52,54,45,52,3a,20,31,36,31
,33,36,31,35,36,5c,30,\

30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,5a,5a
,5a,20,41,54,41,4d,41,\

4e,20,43,4f,4d,4d,41,4e,44,20,53,54,41,52,54,45,52,3a
,20,32,31,38,35,37,34,\

33,37,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43
,59,5f,5a,5a,5a,20,41,\

54,41,4d,41,4e,20,43,4f,4d,4d,41,4e,44,20,53,54,41,52
,54,45,52,3a,20,32,32,\

33,30,37,38,32,38,5c,30,30,30,30,00,52,6f,6f,74,5c,4c
,45,47,41,43,59,5f,5a,\

5a,5a,20,41,54,41,4d,41,4e,20,43,4f,4d,4d,41,4e,44,20
,53,54,41,52,54,45,52,\

3a,20,32,38,33,34,39,35,30,30,5c,30,30,30,30,00,52,6f
,6f,74,5c,4c,45,47,41,\

43,59,5f,5a,5a,5a,20,41,54,41,4d,41,4e,20,43,4f,4d,4d
,41,4e,44,20,53,54,41,\
52,54,45,52,3a,20,35,30,36,37,36,37,31,38,5c,30,30,30
,30,00,52,6f,6f,74,5c,\

4c,45,47,41,43,59,5f,5a,5a,5a,20,41,54,41,4d,41,4e,20
,43,4f,4d,4d,41,4e,44,\

20,53,54,41,52,54,45,52,3a,20,35,34,31,32,33,30,30,30
,5c,30,30,30,30,00,52,\

6f,6f,74,5c,4c,45,47,41,43,59,5f,5a,5a,5a,20,41,54,41
,4d,41,4e,20,43,4f,4d,\

4d,41,4e,44,20,53,54,41,52,54,45,52,3a,20,35,34,33,38
,32,34,38,34,5c,30,30,\

30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,5a,5a,5a
,20,41,54,41,4d,41,4e,\

20,43,4f,4d,4d,41,4e,44,20,53,54,41,52,54,45,52,3a,20
,36,33,31,35,30,31,30,\

39,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59
,5f,5a,5a,5a,20,41,54,\

41,4d,41,4e,20,43,4f,4d,4d,41,4e,44,20,53,54,41,52,54
,45,52,3a,20,31,37,30,\

39,39,37,33,34,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45
,47,41,43,59,5f,41,42,\

49,4f,53,44,53,4b,5c,30,30,30,30,00,52,6f,6f,74,5c,4c
,45,47,41,43,59,5f,4d,\

4f,4e,49,54,4f,52,5c,30,30,30,30,00,52,6f,6f,74,5c,4c
,45,47,41,43,59,5f,43,\

50,51,4b,44,44,50,5c,30,30,30,30,00,52,6f,6f,74,5c,4c
,45,47,41,43,59,5f,43,\

50,51,44,47,44,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45
,47,41,43,59,5f,53,43,\

4d,4f,4e,49,54,4f,52,5c,30,30,30,30,00,52,6f,6f,74,5c
,4c,45,47,41,43,59,5f,\

50,53,45,33,36,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45
,47,41,43,59,5f,53,43,\

41,43,48,45,5c,30,30,30,30,00,52,6f,6f,74,5c,2a,50,4e
,50,30,46,31,32,5c,31,\

5f,30,5f,32,31,5f,30,5f,33,31,5f,30,00,52,6f,6f,74,5c
,4c,45,47,41,43,59,5f,\

5a,5a,5a,20,41,54,41,4d,41,4e,20,43,4f,4d,4d,41,4e,44
,20,53,54,41,52,54,45,\

52,3a,20,36,34,35,30,31,37,31,5c,30,30,30,30,00,52,6f
,6f,74,5c,4c,45,47,41,\

43,59,5f,5a,5a,5a,20,41,54,41,4d,41,4e,20,43,4f,4d,4d
,41,4e,44,20,53,54,41,\

52,54,45,52,3a,20,31,30,36,36,33,39,38,34,5c,30,30,30
,30,00,52,4f,4f,54,5c,\

4c,45,47,41,43,59,5f,43,50,51,41,52,52,59,32,5c,30,30
,30,30,00,52,4f,4f,54,\

5c,53,43,53,49,41,44,41,50,54,45,52,5c,4f,45,4d,30,2e
,49,4e,46,26,43,50,51,\

53,4d,52,54,32,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f
,5a,5a,5a,20,41,54,41,\

4d,41,4e,20,43,4f,4d,4d,41,4e,44,20,53,54,41,52,54,45
,52,3a,20,31,34,30,31,\

38,33,32,38,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47
,41,43,59,5f,5a,5a,5a,\

20,41,54,41,4d,41,4e,20,43,4f,4d,4d,41,4e,44,20,53,54
,41,52,54,45,52,3a,20,\

32,33,32,39,37,38,39,30,5c,30,30,30,30,00,52,6f,6f,74
,5c,4c,45,47,41,43,59,\

5f,5a,5a,5a,20,41,54,41,4d,41,4e,20,43,4f,4d,4d,41,4e
,44,20,53,54,41,52,54,\

45,52,3a,20,39,38,35,31,32,37,33,34,5c,30,30,30,30,00
,52,6f,6f,74,5c,4c,45,\

47,41,43,59,5f,5a,5a,5a,20,41,54,41,4d,41,4e,20,43,4f
,4d,4d,41,4e,44,20,53,\

54,41,52,54,45,52,3a,20,31,38,37,34,30,30,33,35,39,5c
,30,30,30,30,00,52,6f,\

6f,74,5c,4c,45,47,41,43,59,5f,5a,5a,5a,20,41,54,41,4d
,41,4e,20,43,4f,4d,4d,\

41,4e,44,20,53,54,41,52,54,45,52,3a,20,31,39,35,38,38
,30,32,38,31,5c,30,30,\

30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,5a,5a,5a
,20,41,54,41,4d,41,4e,\

20,43,4f,4d,4d,41,4e,44,20,53,54,41,52,54,45,52,3a,20
,31,38,39,35,31,37,31,\

5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f
,5a,5a,5a,20,41,54,41,\

4d,41,4e,20,43,4f,4d,4d,41,4e,44,20,53,54,41,52,54,45
,52,3a,20,31,38,36,34,\

31,38,39,30,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47
,41,43,59,5f,5a,5a,5a,\

20,41,54,41,4d,41,4e,20,43,4f,4d,4d,41,4e,44,20,53,54
,41,52,54,45,52,3a,20,\

34,35,35,38,37,33,32,38,5c,30,30,30,30,00,52,6f,6f,74
,5c,4c,45,47,41,43,59,\

5f,5a,5a,5a,20,41,54,41,4d,41,4e,20,43,4f,4d,4d,41,4e
,44,20,53,54,41,52,54,\

45,52,3a,20,31,37,37,30,35,36,37,31,5c,30,30,30,30,00
,52,6f,6f,74,5c,4c,45,\

47,41,43,59,5f,5a,5a,5a,20,41,54,41,4d,41,4e,20,43,4f
,4d,4d,41,4e,44,20,53,\

54,41,52,54,45,52,3a,20,31,35,30,33,35,33,37,35,5c,30
,30,30,30,00,52,6f,6f,\

74,5c,4c,45,47,41,43,59,5f,5a,5a,5a,20,41,54,41,4d,41
,4e,20,43,4f,4d,4d,41,\

4e,44,20,53,54,41,52,54,45,52,3a,20,34,32,37,36,36,31
,32,35,5c,30,30,30,30,\

00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,4e,58,50,45,52
,46,5c,30,30,30,30,00,\

52,6f,6f,74,5c,4c,45,47,41,43,59,5f,5a,5a,5a,20,41,54
,41,4d,41,4e,20,43,4f,\

4d,4d,41,4e,44,20,53,54,41,52,54,45,52,3a,20,33,36,31
,37,33,33,35,39,5c,30,\

30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,5a,5a
,5a,20,41,54,41,4d,41,\

4e,20,43,4f,4d,4d,41,4e,44,20,53,54,41,52,54,45,52,3a
,20,34,35,32,31,32,34,\

33,37,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43
,59,5f,5a,5a,5a,20,41,\

54,41,4d,41,4e,20,43,4f,4d,4d,41,4e,44,20,53,54,41,52
,54,45,52,3a,20,32,37,\

33,32,34,37,38,31,5c,30,30,30,30,00,52,6f,6f,74,5c,4c
,45,47,41,43,59,5f,5a,\

5a,5a,20,41,54,41,4d,41,4e,20,43,4f,4d,4d,41,4e,44,20
,53,54,41,52,54,45,52,\

3a,20,31,38,32,34,33,37,5c,30,30,30,30,00,52,6f,6f,74
,5c,4c,45,47,41,43,59,\

5f,5a,5a,5a,20,41,54,41,4d,41,4e,20,43,4f,4d,4d,41,4e
,44,20,53,54,41,52,54,\

45,52,3a,20,34,35,38,36,37,36,35,5c,30,30,30,30,00,52
,6f,6f,74,5c,4c,45,47,\

41,43,59,5f,5a,5a,5a,20,41,54,41,4d,41,4e,20,43,4f,4d
,4d,41,4e,44,20,53,54,\

41,52,54,45,52,3a,20,33,31,35,37,34,37,39,36,5c,30,30
,30,30,00,52,6f,6f,74,\

5c,4c,45,47,41,43,59,5f,5a,5a,5a,20,41,54,41,4d,41,4e
,20,43,4f,4d,4d,41,4e,\
44,20,53,54,41,52,54,45,52,3a,20,34,35,37,32,37,33,34
,33,5c,30,30,30,30,00,\

52,6f,6f,74,5c,4c,45,47,41,43,59,5f,5a,5a,5a,20,41,54
,41,4d,41,4e,20,43,4f,\

4d,4d,41,4e,44,20,53,54,41,52,54,45,52,3a,20,34,36,33
,30,39,33,34,33,5c,30,\

30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f,5a,5a
,5a,20,41,54,41,4d,41,\

4e,20,43,4f,4d,4d,41,4e,44,20,53,54,41,52,54,45,52,3a
,20,35,35,38,33,37,35,\

5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47,41,43,59,5f
,5a,5a,5a,20,41,54,41,\

4d,41,4e,20,43,4f,4d,4d,41,4e,44,20,53,54,41,52,54,45
,52,3a,20,32,33,35,32,\

32,36,35,36,5c,30,30,30,30,00,52,6f,6f,74,5c,4c,45,47
,41,43,59,5f,54,41,52,\
  44,49,53,5c,30,30,30,30,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP030b]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP030b\1_0_22_0_32_0]
"HardwareID"=hex(7):2a,50,4e,50,30,33,30,62,00
"BaseDevicePath"="HTREE\\ROOT\\0"
"FoundAtEnum"=dword:00000001
"Service"="i8042prt"
"ClassGUID"="{4D36E96B-E325-11CE-BFC1-08002BE10318}"
"Class"="Keyboard"
"Driver"="{4D36E96B-E325-11CE-BFC1-
08002BE10318}\\0001"
"Mfg"="(Standard keyboards)"
"DeviceDesc"="PC/AT Enhanced Keyboard (101/102-Key)"
"ConfigFlags"=dword:00000000
"Problem"=dword:00000000
"StatusFlags"=dword:00000008

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP030b\1_0_22_0_32_0\LogConf]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP030b\1_0_22_0_32_0\Control]
"ActiveService"="i8042prt"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP030b\1_15_22_0_32_0]
"HardwareID"=hex(7):2a,50,4e,50,30,33,30,62,00
"BaseDevicePath"="HTREE\\ROOT\\0"
"FoundAtEnum"=dword:00000001
"Service"="i8042prt"
"ClassGUID"="{4D36E96B-E325-11CE-BFC1-08002BE10318}"
"Class"="Keyboard"
```

```
"Driver"="{4D36E96B-E325-11CE-BFC1-
08002BE10318}\\0000"
"Mfg"="(Standard keyboards)"
"DeviceDesc"="PC/AT Enhanced Keyboard (101/102-Key)"
"ConfigFlags"=dword:00000000
"Problem"=dword:00000000
"StatusFlags"=dword:00000008

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP030b\1_15_22_0_32_0\LogConf]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP030b\1_15_22_0_32_0\Control]
"ActiveService"="i8042prt"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP0F03]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP0F03\1_0_21_0_31_0]
"HardwareID"=hex(7):2a,50,4e,50,30,46,30,33,00
"BaseDevicePath"="HTREE\\ROOT\\0"
"FoundAtEnum"=dword:00000001
"Service"="i8042prt"
"ClassGUID"="{4D36E96F-E325-11CE-BFC1-08002BE10318}"
"Class"="Mouse"
"Driver"="{4D36E96F-E325-11CE-BFC1-
08002BE10318}\\0001"
"Mfg"="Microsoft"
"DeviceDesc"="Microsoft PS/2 Port Mouse"
"ConfigFlags"=dword:00000000
"Problem"=dword:00000000
"StatusFlags"=dword:00000008

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP0F03\1_0_21_0_31_0\LogConf]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP0F03\1_0_21_0_31_0\Control]
"ActiveService"="i8042prt"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP0F03\1_15_21_0_31_0]
"HardwareID"=hex(7):2a,50,4e,50,30,46,30,33,00
"BaseDevicePath"="HTREE\\ROOT\\0"
"FoundAtEnum"=dword:00000001
"Service"="i8042prt"
"ClassGUID"="{4D36E96F-E325-11CE-BFC1-08002BE10318}"
"Class"="Mouse"
"Driver"="{4D36E96F-E325-11CE-BFC1-
08002BE10318}\\0000"
"Mfg"="Microsoft"
"DeviceDesc"="Microsoft PS/2 Port Mouse"
"ConfigFlags"=dword:00000000
"Problem"=dword:00000000
"StatusFlags"=dword:00000008

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP0F03\1_15_21_0_31_0\LogConf]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP0F03\1_15_21_0_31_0\Control]
"ActiveService"="i8042prt"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP0F12]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP0F12\1_0_21_0_31_0]
"HardwareID"=hex(7):2a,50,4e,50,30,46,31,32,00,00
"BaseDevicePath"="HTREE\\ROOT\\0"
"FoundAtEnum"=dword:00000001
"ClassGUID"="{4d36e96f-e325-11ce-bfc1-08002be10318}"
"Class"="Mouse"
"Service"="i8042prt"
"Driver"="{4D36E96F-E325-11CE-BFC1-
08002BE10318}\\0002"
"Mfg"="Logitech"
"DeviceDesc"="Logitech PS/2 Port Mouse"
"ConfigFlags"=dword:00000000
"Problem"=dword:00000000
"StatusFlags"=dword:00000008

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP0F12\1_0_21_0_31_0\LogConf]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\*PNP0F12\1_0_21_0_31_0\Control]
"ActiveService"="i8042prt"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ABIOSDSK]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ABIOSDSK\0000]
"Service"="Abiosdsk"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Abiosdsk Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_AFD]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_AFD\0000]
"Service"="Afd"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="AFD Networking Support Environment"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_AFD\0000\Control]
"ActiveService"="Afd"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ALERTER]
```

```
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ALERTER\0000]
"Service"="Alerter"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Alerter"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ATAMAN REXECD SERVER]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ATAMAN REXECD SERVER\0000]
"Service"="Ataman Rexecd Server"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Ataman Rexecd Server"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ATAMAN RLOGIND SERVER]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ATAMAN RLOGIND SERVER\0000]
"Service"="Ataman Rlogind Server"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Ataman Rlogind Server"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ATAMAN TELNETD SERVER]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ATAMAN TELNETD SERVER\0000]
"Service"="Ataman Telnetd Server"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Ataman Telnetd Server"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ATAMAN TELNETD SERVER\0000\Control]
"ActiveService"="Ataman Telnetd Server"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ATAPI]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ATAPI\0000]
"Service"="atapi"
"FoundAtEnum"=dword:00000001
"Class"="SCSIAdapter"
"ClassGUID"="{4D36E97B-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="IDE CD-ROM (ATAPI 1.2)/Dual-channel PCI
IDE Controller "
"Driver"="{4D36E97B-E325-11CE-BFC1-
08002BE10318}\\0000"
"Mfg"="(Standard mass storage controllers)"
"HardwareID"=hex(7):61,74,61,70,69,5f,73,63,73,69,00,
00
"ConfigFlags"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ATAPI\0000\LogConf]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ATAPI\0000\Control]
"ActiveService"="atapi"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ATI]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ATI\0000]
"BaseDevicePath"="HTREE\\ROOT\\0"
"FoundAtEnum"=dword:00000001
"ClassGUID"="{4D36E968-E325-11CE-BFC1-08002BE10318}"
"Class"="Display"
"StatusFlags"=dword:00000008
"Driver"="{4D36E968-E325-11CE-BFC1-
08002BE10318}\\0000"
"Mfg"="ATI Technologies Inc."
"Service"="ati"
"DeviceDesc"="ATI Technologies Inc. mach 8 Graphics
Accelerator"
"ConfigFlags"=dword:00000000
"Problem"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ATI\0000\LogConf]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ATI\0000\Control]
"ActiveService"="ati"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_BEEP]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_BEEP\0000]
"Service"="Beep"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Beep Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_BEEP\0000\Control]
"ActiveService"="Beep"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_BROWSER]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_BROWSER\0000]
"Service"="Browser"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Computer Browser"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CDFS]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CDFS\0000]
"Service"="Cdfs"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Cdfs Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CDFS\0000\Control]
"ActiveService"="Cdfs"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CDROM]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CDROM\0000]
"Service"="Cdrom"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Cdrom Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CDROM\0000\Control]
"ActiveService"="Cdrom"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQ32FS2]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQ32FS2\0000]
"BaseDevicePath"="HTREE\\ROOT\\0"
"FoundAtEnum"=dword:00000001
"ClassGUID"="{4D36E97B-E325-11CE-BFC1-08002BE10318}"
"Class"="SCSIAdapter"
"StatusFlags"=dword:00000008
"Driver"="{4D36E97B-E325-11CE-BFC1-
08002BE10318}\\0003"
"Mfg"="Compaq"
"HardwareID"=hex(7):63,70,71,33,32,66,73,32,5f,73,63,
73,69,00,00
"Service"="cpq32fs2"
"DeviceDesc"="Compaq 32-Bit SCSI-2 Controllers"
"ConfigFlags"=dword:00000000
"Problem"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQ32FS2\0000\LogConf]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQ32FS2\0000\Control]
"ActiveService"="cpq32fs2"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQARRAY]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQARRY2]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQARRY2\0000]
"BaseDevicePath"="HTREE\\ROOT\\0"
"FoundAtEnum"=dword:00000001
"ClassGUID"="{4D36E97B-E325-11CE-BFC1-08002BE10318}"
"Class"="SCSIAdapter"
"StatusFlags"=dword:00000000
"Driver"="{4D36E97B-E325-11CE-BFC1-
08002BE10318}\\0001"
"Mfg"="Compaq"
"HardwareID"=hex(7):63,70,71,61,72,72,79,32,5f,73,63,
73,69,00,00
"Service"="cpqarry2"
"DeviceDesc"="Compaq Integrated Smart Array/42xx
Controllers"
"ConfigFlags"=dword:00000000
"Problem"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQARRY2\0000\LogConf]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQDGD]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQDGD\0000]
"Service"="CPQDGD"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="CPQDGD"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQFWS2E]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQFWS2E\0000]
"Service"="CPQFWS2E"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="CPQFWS2E Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQKDD]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQKDD\0000]
"Service"="CPQKDD"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="CPQKDD"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQKDDP]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQKDDP\0000]
"Service"="CPQKDDP"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Compaq Kernel Device Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQNF3]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQNF3\0000]
"Service"="CpqNF3"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Compaq NetFlex-3 Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQSETUP]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_CPQSETUP\0000]
"Service"="CPQSETUP"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Compaq Support Software Service"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_DISK]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_DISK\0000]
"Service"="Disk"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Disk Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_DISK\0000\Control]
"ActiveService"="Disk"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_DISKPERF]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_DISKPERF\0000]
"Service"="Diskperf"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Diskperf Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_DSPCICFG]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_DSPCICFG\0000]
"Service"="DsPciCfg"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
```

```
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="DsPciCfg"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_DSPCICFG\0000\Control]
"ActiveService"="DsPciCfg"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_FASTFAT]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_FASTFAT\0000]
"Service"="Fastfat"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Fastfat Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_FASTFAT\0000\Control]
"ActiveService"="Fastfat"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_FLOPPY]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_FLOPPY\0000]
"Service"="Floppy"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Floppy Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_FLOPPY\0000\Control]
"ActiveService"="Floppy"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_FS_REC]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_FS_REC\0000]
"Service"="Fs_Rec"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Fs_Rec Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_FS_REC\0000\Control]
"ActiveService"="Fs_Rec"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_KBDCLASS]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_KBDCLASS\0000]
"Service"="Kbdclass"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Kbdclass Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_KBDCLASS\0000\Control]
"ActiveService"="Kbdclass"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_KSECDD]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_KSECDD\0000]
"Service"="KSecDD"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="KSecDD Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_KSECDD\0000\Control]
"ActiveService"="KSecDD"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_LANMANSERVER]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_LANMANSERVER\0000]
"Service"="LanmanServer"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Server"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_LANMANSERVER\0000\Control]
"ActiveService"="LanmanServer"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_LANMANWORKSTATION]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_LANMANWORKSTATION\0000]
"Service"="LanmanWorkstation"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Workstation"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_LANMANWORKSTATION\0000\Control]
"ActiveService"="LanmanWorkstation"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_LICENSESERVICE]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_LICENSESERVICE\0000]
"Service"="LicenseService"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="License Logging Service"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_LMHOSTS]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_LMHOSTS\0000]
"Service"="LmHosts"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="TCP/IP NetBIOS Helper"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_MESSENGER]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_MESSENGER\0000]
"Service"="Messenger"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Messenger"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_MONITOR]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_MONITOR\0000]
"Service"="Monitor"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Monitor Service"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_MOUCLASS]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_MOUCLASS\0000]
"Service"="Mouclass"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Mouclass Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_MOUCLASS\0000\Control]
"ActiveService"="Mouclass"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_MSFS]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_MSFS\0000]
"Service"="Msfs"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Msfs Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_MSFS\0000\Control]
"ActiveService"="Msfs"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_MSSQLSERVER]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_MSSQLSERVER\0000]
"Service"="MSSQLServer"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="MSSQLServer"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_MUP]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_MUP\0000]
"Service"="Mup"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Mup"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_MUP\0000\Control]
"ActiveService"="Mup"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_N100]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_N100\0000]
"Service"="N100"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Compaq Ethernet or Fast Ethernet NIC
Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_N100\0000\Control]
"ActiveService"="N100"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NBF]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NBF\0000]
"Service"="Nbf"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="NetBEUI Protocol"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NCRC710]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NCRC710\0000]
"Service"="NCRC710"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="NCRC710 Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NDIS]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NDIS\0000]
"Service"="NDIS"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Microsoft NDIS System Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NDIS\0000\Control]
"ActiveService"="NDIS"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NETBIOS]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NETBIOS\0000]
"Service"="NetBIOS"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="NetBIOS Interface"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NETBT]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NETBT\0000]
"Service"="NetBT"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="WINS Client(TCP/IP)"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NETBT\0000\Control]
"ActiveService"="NetBT"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NETDETECT]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NETDETECT\0000]
"Service"="NetDetect"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="NetDetect"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NETDETECT\0000\Control]
"ActiveService"="NetDetect"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NPFS]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NPFS\0000]
"Service"="Npfs"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Npfs Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NPFS\0000\Control]
"ActiveService"="Npfs"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NTFS]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NTFS\0000]
"Service"="Ntfs"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Ntfs Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NTFS\0000\Control]
"ActiveService"="Ntfs"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NTLMSSP]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NTLMSSP\0000]
"Service"="NtLmSsp"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="NT LM Security Support Provider"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NULL]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NULL\0000]
"Service"="Null"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Null Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NULL\0000\Control]
"ActiveService"="Null"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NXPERF]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_NXPERF\0000]
"Service"="NxPerf"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="NxPerf Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_PARALLEL]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_PARALLEL\0000]
"Service"="Parallel"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Parallel Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_PARALLEL\0000\Control]
"ActiveService"="Parallel"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_PARPORT]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_PARPORT\0000]
"Service"="Parport"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Parport Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_PARPORT\0000\Control]
"ActiveService"="Parport"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_PARVDM]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_PARVDM\0000]
"Service"="ParVdm"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ParVdm Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_PARVDM\0000\Control]
"ActiveService"="ParVdm"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_PROTECTEDSTORAGE]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_PROTECTEDSTORAGE\0000]
"Service"="ProtectedStorage"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Protected Storage"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_PSE36]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_PSE36\0000]
"Service"="PSE36"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
```

```
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="PSE36 Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_RDR]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_RDR\0000]
"Service"="Rdr"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Rdr"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_RDR\0000\Control]
"ActiveService"="Rdr"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_RPCSS]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_RPCSS\0000]
"Service"="RpcSs"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Remote Procedure Call (RPC) Service"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_RPCSS\0000\Control]
"ActiveService"="RpcSs"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SCACHE]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SCACHE\0000]
"Service"="SCache"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="SCache Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SCMONITOR]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SCMONITOR\0000]
"Service"="ScMonitor"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="SuperCache-NT Monitor"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SCSISCAN]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SCSISCAN\0000]
"Service"="Scsiscan"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Scsiscan Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SERIAL]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SERIAL\0000]
"Service"="Serial"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Serial Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SERIAL\0000\Control]
"ActiveService"="Serial"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SPOOLER]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SPOOLER\0000]
"Service"="Spooler"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Spooler Service"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SRV]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SRV\0000]
"Service"="Srv"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Srv"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SRV\0000\Control]
"ActiveService"="Srv"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SYMC810]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SYSMGMT]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_SYSMGMT\0000]
"Service"="SysMgmt"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Compaq System Management Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_TARDIS]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_TARDIS\0000]
"Service"="Tardis"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="Tardis time service"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_TARDIS\0000\Control]
"ActiveService"="Tardis"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_TCPIP]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_TCPIP\0000]
"Service"="Tcpip"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
```

```
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="TCP/IP Service"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_TCPIP\0000\Control]
"ActiveService"="Tcpip"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_VGASAVE]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_VGASAVE\0000]
"Service"="VgaSave"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000008
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="VgaSave Device"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_VGASAVE\0000\Control]
"ActiveService"="VgaSave"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 101330281]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 101330281\0000]
"Service"="ZZZ Ataman Command Starter: 101330281"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 101330281"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 10663984]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 10663984\0000]
"Service"="ZZZ Ataman Command Starter: 10663984"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 10663984"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 106868109]
"NextInstance"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 106868109\0000]
"Service"="ZZZ Ataman Command Starter: 106868109"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 106868109"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 134335937]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 134335937\0000]
"Service"="ZZZ Ataman Command Starter: 134335937"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 134335937"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 14018328]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 14018328\0000]
"Service"="ZZZ Ataman Command Starter: 14018328"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 14018328"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 15035375]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 15035375\0000]
"Service"="ZZZ Ataman Command Starter: 15035375"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 15035375"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 16136156]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 16136156\0000]
"Service"="ZZZ Ataman Command Starter: 16136156"
```

```
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 16136156"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 17099734]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 17099734\0000]
"Service"="ZZZ Ataman Command Starter: 17099734"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 17099734"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 17705671]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 17705671\0000]
"Service"="ZZZ Ataman Command Starter: 17705671"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 17705671"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 182437]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 182437\0000]
"Service"="ZZZ Ataman Command Starter: 182437"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 182437"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 18641890]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 18641890\0000]
"Service"="ZZZ Ataman Command Starter: 18641890"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
```

```
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 18641890"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 187400359]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 187400359\0000]
"Service"="ZZZ Ataman Command Starter: 187400359"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 187400359"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 1895171]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 1895171\0000]
"Service"="ZZZ Ataman Command Starter: 1895171"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 1895171"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 191562421]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 191562421\0000]
"Service"="ZZZ Ataman Command Starter: 191562421"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 191562421"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 195880281]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 195880281\0000]
"Service"="ZZZ Ataman Command Starter: 195880281"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 195880281"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 21857437]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 21857437\0000]
"Service"="ZZZ Ataman Command Starter: 21857437"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 21857437"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 21935359]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 21935359\0000]
"Service"="ZZZ Ataman Command Starter: 21935359"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 21935359"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 22307828]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 22307828\0000]
"Service"="ZZZ Ataman Command Starter: 22307828"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 22307828"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 23297890]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 23297890\0000]
"Service"="ZZZ Ataman Command Starter: 23297890"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 23297890"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 23522656]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 23522656\0000]
"Service"="ZZZ Ataman Command Starter: 23522656"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 23522656"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 24646593]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 24646593\0000]
"Service"="ZZZ Ataman Command Starter: 24646593"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 24646593"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 25754406]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 25754406\0000]
"Service"="ZZZ Ataman Command Starter: 25754406"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 25754406"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 27324781]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 27324781\0000]
"Service"="ZZZ Ataman Command Starter: 27324781"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 27324781"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 28349500]
"NextInstance"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 28349500\0000]
"Service"="ZZZ Ataman Command Starter: 28349500"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 28349500"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 30065890]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 30065890\0000]
"Service"="ZZZ Ataman Command Starter: 30065890"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 30065890"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 31574796]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 31574796\0000]
"Service"="ZZZ Ataman Command Starter: 31574796"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 31574796"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 36173359]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 36173359\0000]
"Service"="ZZZ Ataman Command Starter: 36173359"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 36173359"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 42766125]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 42766125\0000]
"Service"="ZZZ Ataman Command Starter: 42766125"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 42766125"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 45212437]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 45212437\0000]
"Service"="ZZZ Ataman Command Starter: 45212437"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 45212437"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 45587328]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 45587328\0000]
"Service"="ZZZ Ataman Command Starter: 45587328"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 45587328"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 45727343]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 45727343\0000]
"Service"="ZZZ Ataman Command Starter: 45727343"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 45727343"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 4586765]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 4586765\0000]
"Service"="ZZZ Ataman Command Starter: 4586765"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 4586765"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 46309343]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 46309343\0000]
"Service"="ZZZ Ataman Command Starter: 46309343"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 46309343"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 50676718]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 50676718\0000]
"Service"="ZZZ Ataman Command Starter: 50676718"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 50676718"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 53968765]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 53968765\0000]
"Service"="ZZZ Ataman Command Starter: 53968765"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 53968765"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 54123000]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 54123000\0000]
"Service"="ZZZ Ataman Command Starter: 54123000"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
```

```
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 54123000"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 54382484]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 54382484\0000]
"Service"="ZZZ Ataman Command Starter: 54382484"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 54382484"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 558375]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 558375\0000]
"Service"="ZZZ Ataman Command Starter: 558375"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 558375"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 56594421]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 56594421\0000]
"Service"="ZZZ Ataman Command Starter: 56594421"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 56594421"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 56940359]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 56940359\0000]
"Service"="ZZZ Ataman Command Starter: 56940359"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 56940359"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 63150109]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 63150109\0000]
"Service"="ZZZ Ataman Command Starter: 63150109"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 63150109"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 6450171]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 6450171\0000]
"Service"="ZZZ Ataman Command Starter: 6450171"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 6450171"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 67900781]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 67900781\0000]
"Service"="ZZZ Ataman Command Starter: 67900781"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 67900781"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 68377828]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 68377828\0000]
"Service"="ZZZ Ataman Command Starter: 68377828"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 68377828"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 68523093]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 68523093\0000]
"Service"="ZZZ Ataman Command Starter: 68523093"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 68523093"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 70913859]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 70913859\0000]
"Service"="ZZZ Ataman Command Starter: 70913859"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 70913859"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 87017312]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 87017312\0000]
"Service"="ZZZ Ataman Command Starter: 87017312"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 87017312"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 93733546]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 93733546\0000]
"Service"="ZZZ Ataman Command Starter: 93733546"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 93733546"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 98512734]
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\LEGACY_ZZZ ATAMAN COMMAND STARTER: 98512734\0000]
```

```
"Service"="ZZZ Ataman Command Starter: 98512734"
"FoundAtEnum"=dword:00000001
"Class"="Unknown"
"ClassGUID"="{4D36E97E-E325-11CE-BFC1-08002BE10318}"
"Problem"=dword:00000000
"StatusFlags"=dword:00000000
"BaseDevicePath"="HTREE\\ROOT\\0"
"DeviceDesc"="ZZZ Ataman Command Starter: 98512734"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\SCSIADAPTER]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\SCSIADAPTER\OEM0.INF&CPQSMRT2]
"BaseDevicePath"="HTREE\\ROOT\\0"
"FoundAtEnum"=dword:00000001
"ClassGUID"="{4D36E97B-E325-11CE-BFC1-08002BE10318}"
"Class"="SCSIAdapter"
"StatusFlags"=dword:00000008
"Driver"="{4D36E97B-E325-11CE-BFC1-
08002BE10318}\\0002"
"Service"="CpqSMRT2"
"Mfg"="(Additional models)"
"DeviceDesc"="Compaq SMART-2 Array Controllers ECI
Driver"
"ConfigFlags"=dword:00000000
"Problem"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\Roo
t\SCSIADAPTER\OEM0.INF&CPQSMRT2\Control]
"ActiveService"="CpqSMRT2"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001\Software]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001\Software\Microsoft]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001\Software\Microsoft\Windows]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001\Software\Microsoft\Windows\CurrentVersi
on]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001\Software\Microsoft\Windows\CurrentVersi
on\Internet Settings]
"EnableAutodial"=hex:00,00,00,00
"ProxyEnable"=hex:00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001\System]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001\System\CurrentControlSet]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001\System\CurrentControlSet\Control]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001\System\CurrentControlSet\Control\Class]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001\System\CurrentControlSet\Enum]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001\System\CurrentControlSet\Services]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001\System\CurrentControlSet\Services\ati]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001\System\CurrentControlSet\Services\ati\D
evice0]
"DefaultSettings.BitsPerPel"=dword:00000010
"DefaultSettings.XResolution"=dword:00000400
"DefaultSettings.YResolution"=dword:00000300
"DefaultSettings.VRefresh"=dword:0000003c
"DefaultSettings.Flags"=dword:00000000
"DefaultSettings.XPanning"=dword:00000000
"DefaultSettings.YPanning"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001\System\CurrentControlSet\Services\VgaSa
ve]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\0001\System\CurrentControlSet\Services\VgaSa
ve\Device0]
"DefaultSettings.BitsPerPel"=dword:00000004
"DefaultSettings.XResolution"=dword:00000280
"DefaultSettings.YResolution"=dword:000001e0
"DefaultSettings.VRefresh"=dword:00000001
"DefaultSettings.Flags"=dword:00000000
"DefaultSettings.XPanning"=dword:00000000
"DefaultSettings.YPanning"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current\Software]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current\Software\Microsoft]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current\Software\Microsoft\Windows]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current\Software\Microsoft\Windows\CurrentVe
rsion]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current\Software\Microsoft\Windows\CurrentVe
rsion\Internet Settings]
"EnableAutodial"=hex:00,00,00,00
"ProxyEnable"=hex:00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current\System]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current\System\CurrentControlSet]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current\System\CurrentControlSet\Control]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current\System\CurrentControlSet\Control\Cla
ss]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current\System\CurrentControlSet\Enum]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current\System\CurrentControlSet\Services]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current\System\CurrentControlSet\Services\at
i]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current\System\CurrentControlSet\Services\at
i\Device0]
"DefaultSettings.BitsPerPel"=dword:00000010
"DefaultSettings.XResolution"=dword:00000400
"DefaultSettings.YResolution"=dword:00000300
"DefaultSettings.VRefresh"=dword:0000003c
"DefaultSettings.Flags"=dword:00000000
"DefaultSettings.XPanning"=dword:00000000
"DefaultSettings.YPanning"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current\System\CurrentControlSet\Services\Vg
aSave]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware
Profiles\Current\System\CurrentControlSet\Services\Vg
aSave\Device0]
"DefaultSettings.BitsPerPel"=dword:00000004
"DefaultSettings.XResolution"=dword:00000280
"DefaultSettings.YResolution"=dword:000001e0
"DefaultSettings.VRefresh"=dword:00000001
"DefaultSettings.Flags"=dword:00000000
"DefaultSettings.XPanning"=dword:00000000
"DefaultSettings.YPanning"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Abiosdsk]
"Type"=dword:00000001
"Start"=dword:00000004
"ErrorControl"=dword:00000000
"Group"="Primary disk"
"Tag"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Abiosdsk\Enum]
"0"="Root\\LEGACY_ABIOSDSK\\0000"
```

```
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Afd]
"Type"=dword:00000001
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5
c,53,79,73,74,65,6d,33,32,\
  5c,64,72,69,76,65,72,73,5c,61,66,64,2e,73,79,73,00
"DisplayName"="AFD Networking Support Environment"
"Group"="TDI"
"DependOnGroup"=hex(7):00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Afd\Linkage]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Afd\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Afd\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Afd\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,00,00,00,01,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,00,00,00,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,00,00,00
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,00,00,00,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Afd\Enum]
"0"="Root\\LEGACY_AFD\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\AFTRegistration]
"Compatible3"="Compaq,N100,qNFT,3601,32634"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Aha154x]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000006

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Aha174x]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000008

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\aic78xx]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:0000001e

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Alerter]
"Type"=dword:00000020
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="Alerter"
"DependOnService"=hex(7):4c,61,6e,6d,61,6e,57,6f,72,6
b,73,74,61,74,69,6f,6e,00,\
  00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Alerter\Linkage]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Alerter\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Alerter\Parameters]
"AlertNames"=hex(7):00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Alerter\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,25,02,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\
```

```
00,00,00,05,20,00,00,00,23,02,00,00,c8,00,14,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,c8,00,14
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,c8,00,14,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Alerter\Enum]
"0"="Root\\LEGACY_ALERTER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Always]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000014

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ami0nt]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:0000000d

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\amsint]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000024

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Arrow]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000015

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ataman Rexecd Server]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,61,74,61,6d,61,6e,5c,52,4
5,58,45,43,44,2e,45,58,45,\
  00
"DisplayName"="Ataman Rexecd Server"
```

```
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ataman Rexecd Server\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,74,00,73,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,76,00,63,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,76,00,63
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,76,00,63,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ataman Rexecd Server\Enum]
"0"="Root\\LEGACY_ATAMAN REXECD SERVER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ataman Rlogind Server]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,61,74,61,6d,61,6e,5c,52,4
c,4f,47,49,4e,44,2e,45,58,\
  45,00
"DisplayName"="Ataman Rlogind Server"
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ataman Rlogind Server\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,74,00,73,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,76,00,63,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,76,00,63
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,76,00,63,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ataman Rlogind Server\Enum]
"0"="Root\\LEGACY_ATAMAN RLOGIND SERVER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ataman Telnetd Server]
"Type"=dword:00000010
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,61,74,61,6d,61,6e,5c,54,4
5,4c,4e,45,54,44,2e,45,58,\
  45,00
"DisplayName"="Ataman Telnetd Server"
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ataman Telnetd Server\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,67,00,65,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,6e,00,74,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,6e,00,74
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,6e,00,74,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ataman Telnetd Server\Enum]
"0"="Root\\LEGACY_ATAMAN TELNETD SERVER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\atapi]
"Type"=dword:00000001
"Start"=dword:00000000
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000019
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,4
9,56,45,52,53,5c,61,74,61,\
  70,69,2e,73,79,73,00
"PlugPlayServiceType"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\atapi\Enum]
"0"="Root\\LEGACY_ATAPI\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Atdisk]
"Type"=dword:00000001
"Start"=dword:00000004
"ErrorControl"=dword:00000000
"Group"="Primary disk"
"Tag"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ati]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="Video"
"ErrorControl"=dword:00000000
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,4
9,56,45,52,53,5c,61,74,69,\
  2e,73,79,73,00
"Tag"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ati\Device0]
"VgaCompatible"=dword:00000000
"InstalledDisplayDrivers"=hex(7):61,74,69,00,00
"Device Description"="ATI Technologies Inc. 3D RAGE
IIC"
"AllowedPixelDepth"=hex:00
"AllowedRefreshRate"=hex:00
"AllowedResolution"=hex:00
"BlockWriteDisabled"=dword:00000000
"DDC2Disabled"=dword:00000000
"FastFillDisabled"=dword:00000000
"HDTVEnabled"=dword:00000000
"TimingSelection"=dword:00000002
"HardwareInformation.Crc32"=hex:67,fb,1a,5b
"DDC2Supported"=hex:00,00,00,00
"FileToRead"=hex:5c,00,53,00,79,00,73,00,74,00,65,00,
6d,00,52,00,6f,00,6f,00,\
74,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00
,32,00,5c,00,64,00,72,\

00,69,00,76,00,65,00,72,00,73,00,5c,00,61,00,74,00,69
,00,76,00,64,00,69,00,\
  66,00,2e,00,64,00,61,00,74,00,00,00
"HardwareInformation.ChipType"=hex:41,00,54,00,49,00,
20,00,33,00,44,00,20,00,\

52,00,41,00,47,00,45,00,20,00,49,00,49,00,43,00,20,00
,28,00,55,00,4d,00,43,\
  00,20,00,47,00,54,00,2d,00,42,00,29,00,00,00
"HardwareInformation.DacType"=hex:41,00,54,00,49,00,2
0,00,49,00,6e,00,74,00,65,\

00,72,00,6e,00,61,00,6c,00,20,00,44,00,41,00,43,00,00
,00
"HardwareInformation.MemorySize"=hex:00,00,20,00
"HardwareInformation.AdapterString"=hex:41,00,54,00,4
9,00,20,00,47,00,72,00,61,\

00,70,00,68,00,69,00,63,00,73,00,20,00,41,00,63,00,63
,00,65,00,6c,00,65,00,\
  72,00,61,00,74,00,6f,00,72,00,00,00
"HardwareInformation.BiosString"=hex:53,00,47,00,52,0
0,41,00,4d,00,20,00,42,00,\

49,00,4f,00,53,00,20,00,33,00,2e,00,30,00,39,00,35,00
,00,00
"MonitorID"=hex:00,00,00,00,00,00,47,00,65,00,6e,00,6
5,00,72,00,69,00,63,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00
"CustomCRTTimings"=hex:00
"TVOutAdjustments"=hex:00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ati\Enum]
"0"="Root\\LEGACY_ATI\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Beep]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="Base"
"ErrorControl"=dword:00000001
"Tag"=dword:00000002
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Beep\Enum]
"0"="Root\\LEGACY_BEEP\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Browser]
"Type"=dword:00000020
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="Computer Browser"
"DependOnService"=hex(7):4c,61,6e,6d,61,6e,57,6f,72,6
b,73,74,61,74,69,6f,6e,00,\

4c,61,6e,6d,61,6e,53,65,72,76,65,72,00,4c,6d,48,6f,73
,74,73,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Browser\Linkage]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Browser\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Browser\Parameters]
"MaintainServerList"="Auto"
"IsDomainMaster"="FALSE"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Browser\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,25,02,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,c8,00,14,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,c8,00,14
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,c8,00,14,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Browser\Enum]
"0"="Root\\LEGACY_BROWSER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\BusLogic]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:0000001b

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Busmouse]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Pointer Port"
"ErrorControl"=dword:00000000
"Tag"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Busmouse\Parameters]
"MouseDataQueueSize"=dword:00000064
"NumberOfButtons"=dword:00000002
"SampleRate"=dword:00000032
"PointerDeviceBaseName"="PointerPort"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Cdaudio]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="Filter"
"ErrorControl"=dword:00000000
"Tag"=dword:00000006

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Cdaudio\Enum]
"Count"=dword:00000000
"NextInstance"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Cdfs]
"Type"=dword:00000002
"Start"=dword:00000004
"Group"="File system"
"ErrorControl"=dword:00000001
"DependOnGroup"=hex(7):53,43,53,49,20,43,44,52,4f,4d,
20,43,6c,61,73,73,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Cdfs\Enum]
"0"="Root\\LEGACY_CDFS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Cdrom]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="SCSI CDROM Class"
```

```
"ErrorControl"=dword:00000000
"Tag"=dword:00000002
"DependOnGroup"=hex(7):53,43,53,49,20,6d,69,6e,69,70,
6f,72,74,00,00
"Autorun"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Cdrom\Enum]
"0"="Root\\LEGACY_CDROM\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Changer]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="Filter"
"ErrorControl"=dword:00000000
"Tag"=dword:00000005

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Changer\Enum]
"Count"=dword:00000000
"NextInstance"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cirrus]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cirrus\Device0]
"VgaCompatible"=dword:00000001
"InstalledDisplayDrivers"=hex(7):76,67,61,00,63,69,72
,72,75,73,00,76,67,61,32,\
  35,36,00,76,67,61,36,34,4b,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ClipSrv]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,73,79,73,74,65,6d,33,\
  32,5c,63,6c,69,70,73,72,76,2e,65,78,65,00
"DependOnService"=hex(7):4e,65,74,44,44,45,00,00
"DisplayName"="ClipBook Server"
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ClipSrv\Security]
"Security"=hex:01,00,14,80,9c,00,00,00,a8,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,68,00,04,00,00,00,00,03,18,00
,8d,00,02,00,01,01,00,\
```

```

00,00,00,00,01,00,00,00,00,00,00,00,00,00,03,18,00,ff
,01,0f,00,01,02,00,00,\

00,00,00,05,20,00,00,00,20,02,00,00,00,03,18,00,8f,00
,02,00,01,02,00,00,00,\

00,00,05,20,00,00,00,23,02,00,00,00,03,18,00,9d,00,00
,00,01,01,00,00,00,00,\

00,05,04,00,00,00,23,02,00,00,01,01,00,00,00,00,00,05
,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cpq32fs2]
"Type"=dword:00000001
"Start"=dword:00000000
"ErrorControl"=dword:00000001
"Tag"=dword:0000001a
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,4
9,56,45,52,53,5c,63,70,71,\
  33,32,66,73,32,2e,73,79,73,00
"Group"="SCSI Miniport"
"PlugPlayServiceType"=dword:00000001
"RequestedSystemResources"=hex(a):48,01,00,00,05,00,0
0,00,04,00,00,00,29,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,01,00,00,00,01
,00,01,00,09,00,00,00,\

00,02,03,00,00,00,00,00,25,00,00,00,25,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,01,01,01,00,01,00,00,00,00,01,00
,00,00,01,00,00,00,4c,\

00,00,00,00,00,00,ff,4c,00,00,00,00,00,00,08,01,01,00
,01,00,00,00,00,01,00,\

00,00,01,00,00,00,4c,00,00,00,00,00,00,ff,4c,00,00,00
,00,00,00,01,03,01,00,\

00,00,00,00,00,01,00,00,00,01,00,00,00,00,bb,f7,00,00
,00,00,ff,00,bb,f7,00,\

00,00,00,08,03,01,00,00,00,00,00,00,01,00,00,00,01,00
,00,00,00,bb,f7,00,00,\

00,00,ff,00,bb,f7,00,00,00,00,08,03,01,00,00,00,00,00
,00,01,00,00,00,01,00,\

00,00,00,ba,f7,00,00,00,00,ff,0f,ba,f7,00,00,00,00,01
,03,01,00,00,00,00,00,\

00,10,00,00,00,10,00,00,00,00,ba,f7,00,00,00,00,ff,0f
,ba,f7,00,00,00,00,08,\

03,01,00,00,00,00,00,00,10,00,00,00,10,00,00,00,00,bb
,f7,00,00,00,00,ff,00,\

bb,f7,00,00,00,00,08,03,01,00,00,00,00,00,00,10,00,00
,00,10,00,00,00,00,ba,\
```

```
  f7,00,00,00,00,ff,0f,ba,f7,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cpq32fs2\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cpq32fs2\Parameters\Device]
"NumberOfRequest"=dword:000000a4
"NumberOfRequests"=dword:000000a4

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cpq32fs2\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,76,00,63,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,00,00,16,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,00,00,16
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,00,00,16,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cpq32fs2\Enum]
"0"="Root\\LEGACY_CPQ32FS2\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Cpqarray]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000100
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5
c,53,79,73,74,65,6d,33,32,\

5c,44,52,49,56,45,52,53,5c,63,70,71,61,72,72,61,79,2e
,73,79,73,00
"PlugPlayServiceType"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cpqarry2]
"Type"=dword:00000001
"Start"=dword:00000000
```

```
"ErrorControl"=dword:00000001
"Tag"=dword:00000102
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,4
9,56,45,52,53,5c,63,70,71,\
  61,72,72,79,32,2e,73,79,73,00
"Group"="SCSI Miniport"
"PlugPlayServiceType"=dword:00000001
"RequestedSystemResources"=hex(a):c8,00,00,00,05,00,0
0,00,08,00,00,00,05,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,01,00,00,00,01
,00,01,00,05,00,00,00,\

00,02,03,00,00,00,00,00,14,00,00,00,14,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,01,03,01,00,00,00,00,00,00,10,00
,00,00,10,00,00,00,00,\

df,f7,00,00,00,00,ff,0f,df,f7,00,00,00,00,08,03,01,00
,00,00,00,00,00,10,00,\

00,00,10,00,00,00,00,df,f7,00,00,00,00,ff,0f,df,f7,00
,00,00,00,01,01,01,00,\

01,00,00,00,00,01,00,00,00,01,00,00,00,60,00,00,00,00
,00,00,ff,60,00,00,00,\

00,00,00,08,01,01,00,01,00,00,00,00,01,00,00,00,01,00
,00,00,60,00,00,00,00,\
  00,00,ff,60,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cpqarry2\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cpqarry2\Parameters\Device]
"NumberOfRequests"=dword:00000080

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cpqarry2\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,00,00,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,00,00,00,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,00,00,00
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,00,00,00,00,00,00,18,\
```

```
00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cpqarry2\Enum]
"0"="Root\\LEGACY_CPQARRY2\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cpqfws2e]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000026

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\cpqfws2e\Enum]
"0"="Root\\LEGACY_CPQFWS2E\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\CpqNF3]
"Type"=dword:00000001
"Start"=dword:00000004
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5
c,53,79,73,74,65,6d,33,32,\

5c,44,72,69,76,65,72,73,5c,4e,65,74,46,6c,78,33,2e,73
,79,73,00
"DisplayName"="Compaq NetFlex-3 Driver"
"Group"="NDIS"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\CpqNF3\Linkage]
"Bind"=hex(7):00
"Export"=hex(7):00
"Route"=hex(7):00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\CpqNF3\Linkage\Disabled]
"Bind"=hex(7):00
"Export"=hex(7):00
"Route"=hex(7):00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\CpqNF3\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\CpqNF3\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\
```

```
00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,74,00,69,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,76,00,65,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,76,00,65
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,76,00,65,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\CpqNF3\Enum]
"0"="Root\\LEGACY_CPQNF3\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\CPQSET]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\CPQSET\Adapters]
"CP100"="Compaq"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\CPQSET\CP100]
"AdapterDescription"="DIAG100_GetAdapterDescription"
"Configure"="DIAG100_Configure"
"Detect"="DIAG100_Detect"
"DeviceExist"="DIAG100_DeviceExist"
"Diagnose"="DIAG100_Diagnose"
"DLL"="CP100.DLL"
"GetExtendedFeatures"="DIAG100_GetExtendedFeatures"
"Help"="N100SET.HLP"
"InstallAnyway"="DIAG100_InstallAnyway"
"RegistryKey"="DIAG100_GetRegistryKey"
"Summary"="DIAG100_Resource_Summary"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\CPQSET\CP100\Parameters]
"Adaptive_IFS"="1,7,Adaptive Inter-Frame
Spacing,0,2,1,0,255,1"
"BusNumber"="0,7,BusNumber,0,2,0,0,16,1"
"BusType"="0,7,BusType,0,2,5,2,5,1"
"BusTypeLocal"="0,7,BusTypeLocal,0,2,5,2,5,1"
"Eid"="0,7,Eid,0,2,0,0,4294967295,1"
"Fifo"="0,3,Fifo Depth,0,2,12,0,15,1"
"ForceDpx"="1,4,Duplex Mode,0,1,Auto,Auto,Half,Full"
"MapRegisters"=""
"MediaType"="0,7,MediaType,0,2,1,1,1,1"
"MsPciScan"="0,4,MsPciScan,0,2,1,0,1,1"
"NumCoalesce"="1,7,Coalesce Buffers,0,2,8,1,32,1"
```

```
"NumRfd"="1,7,Receive Buffers,0,2,32,1,1024,1"
"NumTbd"="0,3,Transmit Buffer
Descriptors,0,2,64,1,65535,1"
"NumTbdPerTcb"="0,4,Transmit Buffers per
Frame,0,2,12,1,16,1"
"NumTcb"="1,7,Transmit Control Blocks,0,2,16,1,80,1"
"Off"="1,3,Off Timer,0,2,2,1,65535,1"
"On"="1,3,On Timer,0,2,32768,1,65535,1"
"PerfOptims"="0,4,PerfOptims,0,2,0,0,65535,1"
"RxDmaCount"="0,4,RxDmaCount,0,2,0,0,63,1"
"RxFifo"="0,4,Receive Fifo Depth,0,2,8,0,15,1"
"Slot"=""
"Speed"="1,7,Network
Speed,0,4,Auto,Auto,0,10Mbps,10,100Mbps,100"
"Threshold"="0,7,Transmit Threshold,0,2,16,0,200,1"
"TxDmaCount"="0,4,TxDmaCount,0,2,0,0,63,1"
"TxFifo"="0,4,Transmit Fifo Depth,0,2,8,0,15,1"
"Txmitwait"="0,7,Txmitwait,0,2,1,0,255,1"
"UcodeSW"="0,7,UcodeSW,0,2,1,0,1,1"
"UnderrunRetry"="0,4,UnderrunRetry,0,2,1,0,3,1"
"NetworkAddress"="1,7,Locally Administered
Address,0,5,0,0,1,1"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\CpqSMRT2]
"Type"=dword:00000001
"Start"=dword:00000000
"ErrorControl"=dword:00000001
"Tag"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,64,72,6
9,76,65,72,73,5c,43,50,51,\
  53,4d,52,54,32,2e,53,59,53,00
"Group"="Primary disk"
"PlugPlayServiceType"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\CpqSMRT2\Parameters]
"InterruptMask"=dword:ffffff7f
"DynamicMask"=dword:ffffffff
"DTR"=dword:0000000f

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\CpqSMRT2\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,01,00,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,6c,03,00,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,6c,03,00
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,6c,03,00,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\CpqSMRT2\Enum]
"0"="Root\\SCSIADAPTER\\OEM0.INF&CPQSMRT2"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\dac960nt]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000063
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5
c,53,79,73,74,65,6d,33,32,\

5c,64,72,69,76,65,72,73,5c,64,61,63,39,36,30,6e,74,2e
,73,79,73,00
"PlugPlayServiceType"=dword:00000001
"RequestedSystemResources"=hex(a):28,01,00,00,05,00,0
0,00,0d,00,00,00,08,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,01,00,00,00,01
,00,01,00,08,00,00,00,\

00,02,03,00,00,00,00,00,14,00,00,00,14,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,01,03,01,00,00,00,00,00,80,00,00
,00,80,00,00,00,00,00,\

ff,f7,00,00,00,00,7f,00,ff,f7,00,00,00,00,08,03,01,00
,00,00,00,00,80,00,00,\

00,80,00,00,00,00,00,ff,f7,00,00,00,00,7f,00,ff,f7,00
,00,00,00,01,01,01,00,\

01,00,00,00,80,00,00,00,80,00,00,00,00,d0,00,00,00,00
,00,00,7f,d0,00,00,00,\

00,00,00,08,01,01,00,01,00,00,00,80,00,00,00,80,00,00
,00,00,d0,00,00,00,00,\

00,00,7f,d0,00,00,00,00,00,00,01,03,01,00,04,00,00,00
,00,00,00,02,00,00,00,\

02,00,00,00,f4,00,00,00,00,ff,ff,ff,f5,00,00,00,00,09
,03,01,00,04,00,00,00,\

00,00,00,02,00,00,00,02,00,00,00,f4,00,00,00,00,ff,ff
,ff,f5,00,00,00,00,08,\

03,01,00,04,00,00,00,00,00,00,02,00,00,00,02,00,00,ff
,f7,00,00,00,00,7f,00,\
  ff,f7,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\dce376nt]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000016

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Delldsa]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000101

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Dell_DGX]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Dell_DGX\Device0]
"VgaCompatible"=dword:00000000
"InstalledDisplayDrivers"=hex(7):66,72,61,6d,65,62,75
,66,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DHCP]
"Type"=dword:00000020
"Start"=dword:00000004
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="DHCP Client"
"Group"="TDI"
"DependOnService"=hex(7):54,63,70,69,70,00,41,66,64,0
0,4e,65,74,42,54,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DHCP\Linkage]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DHCP\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DHCP\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DHCP\Parameters\Options]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DHCP\Parameters\Options\1]
"RegLocation"="System\\CurrentControlSet\\Services\\?
\\Parameters\\Tcpip\\DhcpSubnetMaskOpt"
"KeyType"=dword:00000007
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DHCP\Parameters\Options\15]
"RegLocation"="System\\CurrentControlSet\\Services\\T
cpip\\Parameters\\DhcpDomain"
"KeyType"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DHCP\Parameters\Options\3]
"RegLocation"="System\\CurrentControlSet\\Services\\?
\\Parameters\\Tcpip\\DhcpDefaultGateway"
"KeyType"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DHCP\Parameters\Options\44]
"RegLocation"="System\\CurrentControlSet\\Services\\N
etBT\\Adapters\\?\\DhcpNameServer"
"KeyType"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DHCP\Parameters\Options\46]
"RegLocation"="System\\CurrentControlSet\\Services\\N
etBT\\Parameters\\DhcpNodeType"
"KeyType"=dword:00000004

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DHCP\Parameters\Options\47]
"RegLocation"="System\\CurrentControlSet\\Services\\N
etBT\\Parameters\\DhcpScopeID"
"KeyType"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DHCP\Parameters\Options\6]
"RegLocation"="System\\CurrentControlSet\\Services\\T
cpip\\Parameters\\DhcpNameServer"
"KeyType"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DHCP\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,20,02,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,69,00,64,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,69,00,64
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,69,00,64,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Disk]
"Type"=dword:00000001
"Start"=dword:00000000
"Group"="SCSI Class"
"ErrorControl"=dword:00000000
"Tag"=dword:00000002
"DependOnGroup"=hex(7):53,43,53,49,20,6d,69,6e,69,70,
6f,72,74,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Disk\Enum]
"0"="Root\\LEGACY_DISK\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Diskperf]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Filter"
"ErrorControl"=dword:00000001
"Tag"=dword:00000004

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Diskperf\Enum]
"0"="Root\\LEGACY_DISKPERF\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DptScsi]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000009

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DsPciCfg]
"Type"=dword:00000001
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):5c,3f,3f,5c,43,3a,5c,57,49,4e,4e,5
4,34,30,5c,73,79,73,74,65,\

6d,33,32,5c,64,72,69,76,65,72,73,5c,44,73,50,63,69,43
,66,67,2e,53,59,53,00
"DisplayName"="DsPciCfg"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DsPciCfg\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,74,00,73,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,76,00,63,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,76,00,63
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,76,00,63,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\DsPciCfg\Enum]
"0"="Root\\LEGACY_DSPCICFG\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\dtc329x]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000017

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\et4000]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\et4000\Device0]
"VgaCompatible"=dword:00000001
"InstalledDisplayDrivers"=hex(7):76,67,61,00,77,33,32
,00,76,67,61,32,35,36,00,\
  76,67,61,36,34,4b,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog]
"Type"=dword:00000020
"Start"=dword:00000002
"Group"="Event log"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,73,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"ObjectName"="LocalSystem"
"PlugPlayServiceType"=dword:00000003
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application]
"File"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,
73,79,73,74,65,6d,33,32,\

5c,63,6f,6e,66,69,67,5c,41,70,70,45,76,65,6e,74,2e,45
,76,74,00
"MaxSize"=dword:00080000
"Retention"=dword:00000000
"Sources"=hex(7):57,69,6e,6c,6f,67,6f,6e,00,57,69,6e,
64,6f,77,73,20,33,2e,31,\

20,4d,69,67,72,61,74,69,6f,6e,00,56,42,52,75,6e,74,69
,6d,65,00,55,73,65,72,\

65,6e,76,00,54,79,70,65,20,31,20,49,6e,73,74,61,6c,6c
,65,72,00,54,61,72,64,\

69,73,00,53,51,4c,53,65,72,76,65,72,50,72,6f,66,69,6c
,65,72,00,53,51,4c,53,\

65,72,76,65,72,41,67,65,6e,74,00,53,51,4c,43,54,52,37
,30,00,53,4d,41,52,54,\

63,74,72,00,52,65,70,6c,69,63,61,74,6f,72,00,50,53,45
,33,36,50,72,66,00,50,\

65,72,66,6d,6f,6e,00,50,65,72,66,6c,69,62,00,50,65,72
,66,63,74,72,73,00,4e,\

78,50,65,72,66,00,6e,74,62,61,63,6b,75,70,00,4d,53,53
,51,4c,53,65,72,76,65,\

72,00,4d,53,44,54,43,00,4c,69,63,65,6e,73,65,53,65,72
,76,69,63,65,00,4a,61,\

76,61,20,56,4d,00,68,70,6d,6f,6e,00,44,72,57,61,74,73
,6f,6e,00,44,61,74,61,\

54,72,61,6e,73,66,6f,72,6d,61,74,69,6f,6e,53,65,72,76
,69,63,65,73,00,64,61,\

74,61,6c,6f,67,00,41,75,74,6f,63,68,6b,00,41,74,61,6d
,61,6e,20,54,65,6c,6e,\

65,74,64,20,53,65,72,76,65,72,00,41,74,61,6d,61,6e,20
,52,6c,6f,67,69,6e,64,\

20,53,65,72,76,65,72,00,41,74,61,6d,61,6e,20,52,65,78
,65,63,64,20,53,65,72,\

76,65,72,00,41,74,61,6d,61,6e,20,43,6f,6d,6d,61,6e,64
,20,54,72,61,6e,73,6c,\

61,74,6f,72,00,41,74,61,6d,61,6e,20,43,6f,6d,6d,61,6e
,64,20,53,74,61,72,74,\
  65,72,00,41,70,70,6c,69,63,61,74,69,6f,6e,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Ataman Command Starter]
"TypesSupported"=dword:00000007
"EventMessageFile"=hex(2):43,3a,5c,61,74,61,6d,61,6e,
5c,73,74,61,72,74,63,6d,\
  64,2e,65,78,65,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Ataman Command Translator]
"TypesSupported"=dword:00000007
"EventMessageFile"=hex(2):43,3a,5c,61,74,61,6d,61,6e,
5c,74,72,61,6e,73,63,6d,\
  64,2e,65,78,65,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Ataman Rexecd Server]
"TypesSupported"=dword:00000007
"EventMessageFile"=hex(2):43,3a,5c,61,74,61,6d,61,6e,
5c,72,65,78,65,63,64,2e,\
  65,78,65,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Ataman Rlogind Server]
"TypesSupported"=dword:00000007
"EventMessageFile"=hex(2):43,3a,5c,61,74,61,6d,61,6e,
5c,72,6c,6f,67,69,6e,64,\
  2e,65,78,65,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Ataman Telnetd Server]
"TypesSupported"=dword:00000007
"EventMessageFile"=hex(2):43,3a,5c,61,74,61,6d,61,6e,
5c,74,65,6c,6e,65,74,64,\
  2e,65,78,65,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Autochk]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,77,69,6e,6c,6f,67,6f,6e,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\datalog]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,73,79,73,74,\
  65,6d,33,32,5c,64,61,74,61,6c,6f,67,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\DataTransformationServices]
"EventMessageFile"="C:\\MSSQL7\\Binn\\dtspkg.dll"
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\DrWatson]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,64,72,77,74,73,6e,33,32,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\hpmon]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,68,70,6d,6f,6e,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Java VM]
"EventMessageFile"="C:\\WINNT40\\System32\\vmhelper.d
ll"
"TypesSupported"=hex:07,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\LicenseService]
"TypesSupported"=dword:00000007
"EventMessageFile"="%SystemRoot%\\System32\\llsrpc.dl
l"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\MSDTC]
"EventMessageFile"=hex(2):43,3a,5c,57,49,4e,4e,54,34,
30,5c,53,79,73,74,65,6d,\
  33,32,5c,45,4e,55,64,74,63,2e,64,6c,6c,00
"TypesSupported"=dword:00000007
"CategoryMessageFile"=hex(2):43,3a,5c,57,49,4e,4e,54,
34,30,5c,53,79,73,74,65,\
  6d,33,32,5c,45,4e,55,64,74,63,2e,64,6c,6c,00
"CategoryCount"=dword:00000006

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\MSSQLServer]
"TypesSupported"=dword:000000ff
"CategoryCount"=dword:00000008
"EventMessageFile"="C:\\MSSQL7\\BINN\\SQLEVN70.DLL"
"CategoryMessageFile"="C:\\MSSQL7\\BINN\\SQLEVN70.DLL
"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\ntbackup]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,6e,74,62,61,63,6b,75,70,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\NxPerf]
"EventMessageFile"="%SystemRoot%\\system32\\NxCtrs.dl
l"
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Perfctrs]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,70,65,72,66,63,74,72,73,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Perflib]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,70,72,66,6c,62,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Perfmon]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,70,65,72,66,6d,6f,6e,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\PSE36Prf]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,70,73,65,33,36,70,72,66,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Replicator]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,4e,65,74,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\SMARTctr]
"EventMessageFile"="%SystemRoot%\\system32\\SMARTctr.
dll"
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\SQLCTR70]
"TypesSupported"=dword:00000007
"EventMessageFile"="C:\\MSSQL7\\BINN\\SQLEVN70.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\SQLServerAgent]
"TypesSupported"=dword:00000007
"CategoryCount"=dword:00000004
"EventMessageFile"="C:\\MSSQL7\\BINN\\SQLAGENT.DLL"
"CategoryMessageFile"="C:\\MSSQL7\\BINN\\SQLAGENT.DLL
"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\SQLServerProfiler]
"EventMessageFile"=hex(2):43,3a,5c,4d,53,53,51,4c,37,
5c,42,49,4e,4e,5c,43,4f,\
  4d,4e,45,56,4e,54,2e,44,4c,4c,00
"CategoryMessageFile"=hex(2):43,3a,5c,4d,53,53,51,4c,
37,5c,42,49,4e,4e,5c,43,\
  4f,4d,4e,45,56,4e,54,2e,44,4c,4c,00
"TypesSupported"=dword:00000007
"CategoryCount"=dword:00000008

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Tardis]
"EventMessageFile"="%SystemRoot%\\System32\\tardis.cp
l"
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Type 1 Installer]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,74,31,69,6e,73,74,61,6c,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Userenv]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,75,73,65,72,65,6e,76,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\VBRuntime]
"EventMessageFile"="C:\\WINNT40\\System32\\MSVBVM60.D
LL"
"TypesSupported"=dword:00000004

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Windows 3.1 Migration]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,61,64,76,61,70,69,33,32,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Application\Winlogon]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,77,69,6e,6c,6f,67,6f,6e,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Security]
"File"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,
53,79,73,74,65,6d,33,32,\

5c,63,6f,6e,66,69,67,5c,53,65,63,45,76,65,6e,74,2e,45
,76,74,00
"MaxSize"=dword:00080000
"Retention"=dword:00093a80
"PrimaryModule"="Security"
"Sources"=hex(7):53,70,6f,6f,6c,65,72,00,53,65,63,75,
72,69,74,79,20,41,63,63,\

6f,75,6e,74,20,4d,61,6e,61,67,65,72,00,53,43,20,4d,61
,6e,61,67,65,72,00,4e,\

65,74,44,44,45,20,4f,62,6a,65,63,74,00,4c,53,41,00,53
,65,63,75,72,69,74,79,\
  00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Security\LSA]
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52
,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,4d,73,4f,62,6a,73,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Security\LSA\ObjectNames]
"PolicyObject"=dword:00001600
"SecretObject"=dword:00001610
"TrustedDomainObject"=dword:00001620
"UserAccountObject"=dword:00001630

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Security\NetDDE Object]
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52
,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,4d,73,4f,62,6a,73,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Security\NetDDE Object\ObjectNames]
"DDE Share"=dword:00001d00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Security\SC Manager]
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52
,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,4d,73,4f,62,6a,73,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Security\SC Manager\ObjectNames]
"SC_MANAGER Object"=dword:00001c00
"SERVICE Object"=dword:00001c10

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Security\Security]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,4d,73,41,75,64,69,74,45,2e,64,6c,6c,00
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52
,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,4d,73,4f,62,6a,73,2e,64,6c,6c,00
"CategoryMessageFile"=hex(2):25,53,79,73,74,65,6d,52,
6f,6f,74,25,5c,53,79,73,\

74,65,6d,33,32,5c,4d,73,41,75,64,69,74,45,2e,64,6c,6c
,00
"CategoryCount"=dword:00000007
"TypesSupported"=dword:0000001c

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Security\Security\ObjectNames]
"Device"=dword:00001100
"Directory"=dword:00001110
"Event"=dword:00001120
"EventPair"=dword:00001130
"File"=dword:00001140
"Key"=dword:00001150
"Mutant"=dword:00001160
"Port"=dword:00001170
"Process"=dword:00001180
"Profile"=dword:00001190
"Section"=dword:000011a0
"Semaphore"=dword:000011b0
"SymbolicLink"=dword:000011c0
"Thread"=dword:000011d0
"Timer"=dword:000011e0
"Token"=dword:000011f0
"Type"=dword:00001200
"IoCompletion"=dword:00001300
"MailSlot"=dword:00001140
"NamedPipe"=dword:00001140
```

```
"Channel"=dword:00001400
"WindowStation"=dword:00001a00
"Desktop"=dword:00001a10

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Security\Security Account Manager]
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52
,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,4d,73,4f,62,6a,73,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Security\Security Account
Manager\ObjectNames]
"SAM_SERVER"=dword:00001500
"SAM_DOMAIN"=dword:00001510
"SAM_GROUP"=dword:00001520
"SAM_ALIAS"=dword:00001530
"SAM_USER"=dword:00001540

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Security\Spooler]
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52
,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,4d,73,4f,62,6a,73,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\Security\Spooler\ObjectNames]
"Server"=dword:00001b00
"Printer"=dword:00001b10
"Document"=dword:00001b20

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System]
"File"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,
73,79,73,74,65,6d,33,32,\

5c,63,6f,6e,66,69,67,5c,53,79,73,45,76,65,6e,74,2e,45
,76,74,00
"MaxSize"=dword:00080000
"Retention"=dword:00000000
"Sources"=hex(7):78,67,61,00,77,6f,72,6b,73,74,61,74,
69,6f,6e,00,77,65,69,74,\

65,6b,70,39,00,77,64,76,67,61,00,77,64,39,30,63,32,34
,61,00,77,64,33,33,63,\

39,33,00,56,67,61,53,74,61,72,74,00,56,67,61,53,61,76
,65,00,76,37,76,72,61,\

6d,00,55,50,53,00,75,6c,74,72,61,32,34,66,00,75,6c,74
,72,61,31,34,66,00,75,\

6c,74,72,61,31,32,34,00,74,6d,76,31,00,74,64,69,00,54
,63,70,69,70,00,74,31,\

33,62,00,74,31,32,38,00,53,79,73,4d,67,6d,74,00,73,79
,6d,63,38,31,30,00,53,\

72,76,00,73,70,6f,63,6b,00,73,70,61,72,72,6f,77,00,73
,6e,64,62,6c,73,74,00,\

73,6c,63,64,33,32,00,53,69,6d,62,61,64,00,73,66,6c,6f
,70,70,79,00,53,65,72,\

76,69,63,65,20,43,6f,6e,74,72,6f,6c,20,4d,61,6e,61,67
,65,72,00,73,65,72,76,\

65,72,00,73,65,72,6d,6f,75,73,65,00,73,65,72,69,61,6c
,00,73,63,73,69,73,63,\

61,6e,00,73,63,73,69,70,72,6e,74,00,73,63,73,69,70,6f
,72,74,00,53,63,68,65,\

64,75,6c,65,00,53,43,61,63,68,65,00,53,61,76,65,20,44
,75,6d,70,00,53,41,4d,\

00,73,33,00,52,64,72,00,71,76,00,71,6c,31,30,77,6e,74
,00,70,73,69,64,69,73,\

70,00,50,53,45,33,36,00,50,72,69,6e,74,00,70,6e,70,69
,73,61,00,70,63,6d,63,\

69,61,00,70,61,72,76,64,6d,00,70,61,72,70,6f,72,74,00
,70,61,72,61,6c,6c,65,\

6c,00,6f,6c,69,73,63,73,69,00,6e,75,6c,6c,00,4e,74,53
,65,72,76,69,63,65,50,\

61,63,6b,00,6e,74,66,73,00,6e,70,66,73,00,4e,65,74,4c
,6f,67,6f,6e,00,4e,65,\

74,44,44,45,00,4e,65,74,42,54,00,4e,65,74,42,49,4f,53
,00,6e,64,69,73,00,6e,\

63,72,63,37,31,30,00,6e,63,72,63,37,30,30,00,6e,63,72
,37,37,63,32,32,00,6e,\

63,72,35,33,63,39,78,00,4e,62,66,00,4e,31,30,30,00,4d
,75,70,00,6d,73,66,73,\

00,6d,73,61,64,6c,69,62,00,6d,6f,75,63,6c,61,73,73,00
,4d,6f,64,65,6d,00,6d,\

6b,65,63,72,35,78,78,00,6d,69,74,73,75,6d,69,00,6d,67
,61,5f,6d,69,6c,00,6d,\

67,61,00,6d,61,63,64,69,73,6b,00,4c,6d,48,6f,73,74,73
,00,6b,62,64,63,6c,61,\

73,73,00,4a,7a,76,78,6c,34,38,34,00,4a,61,7a,7a,53,6e
,64,00,4a,61,7a,7a,47,\

33,36,34,00,4a,61,7a,7a,47,33,30,30,00,69,6e,70,6f,72
,74,00,69,38,30,34,32,\

70,72,74,00,66,74,64,69,73,6b,00,66,73,5f,72,65,63,00
,66,6c,6f,70,70,79,00,\

66,6c,61,73,68,70,6e,74,00,66,64,38,58,58,00,66,64,37
,30,30,30,65,78,00,66,\

64,31,36,5f,37,30,30,00,66,61,73,74,66,61,74,00,65,76
,65,6e,74,6c,6f,67,00,\

65,74,34,30,30,30,00,64,74,63,33,32,39,78,00,64,70,74
,73,63,73,69,00,64,69,\

73,6b,70,65,72,66,00,64,69,73,6b,00,44,48,43,50,00,64
,65,6c,6c,5f,64,67,78,\

00,64,65,6c,6c,64,73,61,00,44,43,4f,4d,00,64,63,65,33
,37,36,6e,74,00,64,61,\

63,39,36,30,6e,74,00,43,70,71,53,4d,52,54,32,00,43,50
,51,53,45,54,55,50,00,\

43,70,71,4e,46,33,00,63,70,71,66,77,73,32,65,00,63,70
,71,61,72,72,79,32,00,\

63,70,71,61,72,72,61,79,00,63,70,71,33,32,66,73,32,00
,63,69,72,72,75,73,00,\

63,68,61,6e,67,65,72,00,63,64,72,6f,6d,00,63,64,66,73
,00,63,64,61,75,64,69,\

6f,00,62,75,73,6d,6f,75,73,65,00,62,75,73,6c,6f,67,69
,63,00,42,72,6f,77,73,\

65,72,00,62,65,65,70,00,61,74,69,00,61,74,64,69,73,6b
,00,61,74,61,70,69,00,\

61,72,72,6f,77,00,41,70,70,6c,69,63,61,74,69,6f,6e,20
,50,6f,70,75,70,00,61,\

6d,73,69,6e,74,00,61,6d,69,30,6e,74,00,61,6c,77,61,79
,73,00,41,6c,65,72,74,\

65,72,00,61,69,63,37,38,78,78,00,61,68,61,31,37,34,78
,00,61,68,61,31,35,34,\

78,00,41,66,64,00,61,62,69,6f,73,64,73,6b,00,53,79,73
,74,65,6d,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\abiosdsk]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Afd]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\aha154x]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\aha174x]
```

```
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\aic78xx]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Alerter]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,4e,65,74,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\always]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\ami0nt]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\amsint]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Application Popup]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,74,64,6c,6c,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\arrow]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\atapi]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\atdisk]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\ati]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,64,72,69
,76,65,72,73,5c,61,74,\
  69,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\beep]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Browser]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\buslogic]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\busmouse]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,42,75,\
  73,4d,6f,75,73,65,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\cdaudio]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\cdfs]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\cdrom]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\changer]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\cirrus]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\cpq32fs2]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\cpqarray]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\cpqarry2]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
```

```
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\cpqfws2e]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\CpqNF3]
"TypesSupported"=dword:00000007
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,4e,65,74,65,76,65,6e,74,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,4e,65,\
  74,46,6c,78,33,2e,53,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\CPQSETUP]
"TypesSupported"=dword:00000000
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,44,4c,4c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\CpqSMRT2]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,44,72,69,76,65,72,73,5c,43,50,51,53,4d
,52,54,32,2e,53,59,53,\

3b,25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74
,65,6d,33,32,5c,49,6f,\
  4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\dac960nt]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\dce376nt]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\DCOM]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52
,6f,6f,74,25,5c,53,79,73,\

74,65,6d,33,32,5c,6b,65,72,6e,65,6c,33,32,2e,64,6c,6c
,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\delldsa]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\dell_dgx]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\DHCP]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,64,68,63,70,63,73,76,63,2e,64,6c,6c,00
"TypesSupported"=dword:00000007
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52
,6f,6f,74,25,5c,53,79,73,\

74,65,6d,33,32,5c,6b,65,72,6e,65,6c,33,32,2e,64,6c,6c
,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\disk]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\diskperf]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\dptscsi]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\dtc329x]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\et4000]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\eventlog]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\fastfat]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\fd16_700]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\fd7000ex]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\fd8XX]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\flashpnt]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
```

```
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\floppy]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\fs_rec]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\ftdisk]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,46,74,\
  44,69,73,6b,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\i8042prt]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,64,72,69
,76,65,72,73,5c,69,38,\
  30,34,32,70,72,74,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\inport]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,49,6e,\
  50,6f,72,74,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\JazzG300]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\JazzG364]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\JazzSnd]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Jzvxl484]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\kbdclass]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,64,72,69
,76,65,72,73,5c,6b,62,\
  64,63,6c,61,73,73,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\LmHosts]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\macdisk]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\mga]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,6d,67,\
  61,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\mga_mil]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,6d,67,\
  61,5f,6d,69,6c,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\mitsumi]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\mkecr5xx]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Modem]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,4d,6f,\
  64,65,6d,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\mouclass]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,64,72,69
,76,65,72,73,5c,6d,6f,\
  75,63,6c,61,73,73,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\msadlib]
```

```
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\msfs]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Mup]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\N100]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,4e,65,74,65,76,65,6e,74,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,4e,31,\
  30,30,4e,54,2e,53,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Nbf]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\ncr53c9x]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\ncr77c22]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\ncrc700]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\ncrc710]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\ndis]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\NetBIOS]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\NetBT]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\NetDDE]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,64,64,65,2e,65,78,65,00
"TypesSupported"=dword:0000001f

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\NetLogon]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,4e,65,74,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52
,6f,6f,74,25,5c,53,79,73,\

74,65,6d,33,32,5c,4b,65,72,6e,65,6c,33,32,2e,64,6c,6c
,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\npfs]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\ntfs]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\NtServicePack]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,73,70,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\null]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\oliscsi]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\parallel]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,50,61,\
  72,61,6c,6c,65,6c,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\parport]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,50,61,\
  72,50,6f,72,74,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\parvdm]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\
65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,50,61,\
  72,56,64,6d,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\pcmcia]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,50,63,\
  6d,63,69,61,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\pnpisa]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,70,6e,\
  70,69,73,61,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Print]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,4c,6f,63,61,6c,53,70,6c,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\PSE36]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,44,72,69,76,65,72,73,5c,70,73,65,33,36
,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\psidisp]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\ql10wnt]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\qv]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,71,76,\
  2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Rdr]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\s3]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,73,33,\
  2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\SAM]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,73,61,6d,73,72,76,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Save Dump]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,53,61,76,65,44,75,6d,70,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\SCache]
"TypesSupported"=dword:00000007
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,44,72,69,76,65,72,73,5c,53,43,61,63,68
,65,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Schedule]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52
,6f,6f,74,25,5c,53,79,73,\

74,65,6d,33,32,5c,6b,65,72,6e,65,6c,33,32,2e,64,6c,6c
,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\scsiport]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\scsiprnt]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\scsiscan]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\serial]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,53,65,\
  72,69,61,6c,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\sermouse]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,53,65,\
  72,4d,6f,75,73,65,2e,73,79,73,00
"TypesSupported"=dword:00000007
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\server]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Service Control Manager]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52
,6f,6f,74,25,5c,53,79,73,\

74,65,6d,33,32,5c,6b,65,72,6e,65,6c,33,32,2e,64,6c,6c
,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\sfloppy]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Simbad]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\slcd32]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\sndblst]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\sparrow]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\spock]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Srv]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\symc810]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\SysMgmt]
"TypesSupported"=dword:00000007
"EventMessageFile"=hex(2):43,3a,5c,57,49,4e,4e,54,34,
30,5c,53,59,53,54,45,4d,\

33,32,5c,44,52,49,56,45,52,53,5c,53,59,53,4d,47,4d,54
,2e,53,59,53,3b,43,3a,\

5c,57,49,4e,4e,54,34,30,5c,53,59,53,54,45,4d,33,32,5c
,49,4f,4c,4f,47,4d,53,\
  47,2e,44,4c,4c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\t128]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\t13b]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\Tcpip]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\tdi]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\tmv1]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\ultra124]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\ultra14f]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\ultra24f]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\UPS]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\v7vram]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\VgaSave]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,76,67,\
  61,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\VgaStart]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,76,67,\
  61,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\wd33c93]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\wd90c24a]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\wdvga]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\weitekp9]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,77,65,\
  69,74,65,6b,70,39,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\workstation]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,4e,65,74,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\EventLog\System\xga]
```

```
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,
6f,74,25,5c,53,79,73,74,\

65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b
,25,53,79,73,74,65,6d,\

52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69
,76,65,72,73,5c,78,67,\
  61,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Fastfat]
"Type"=dword:00000002
"Start"=dword:00000004
"Group"="Boot file system"
"ErrorControl"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Fastfat\Enum]
"0"="Root\\LEGACY_FASTFAT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Fd16_700]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000010

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Fd7000ex]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:0000000e

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Fd8xx]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:0000000f

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Fd8xx\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Fd8xx\Parameters\Device0]
"DriverParameter"="IRQ=5"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\flashpnt]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000025
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Floppy]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="Primary disk"
"ErrorControl"=dword:00000000
"Tag"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Floppy\Enum]
"0"="Root\\LEGACY_FLOPPY\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Fs_Rec]
"Type"=dword:00000008
"Start"=dword:00000001
"Group"="Boot file system"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Fs_Rec\Enum]
"0"="Root\\LEGACY_FS_REC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ftdisk]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Filter"
"ErrorControl"=dword:00000000
"Tag"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\i8042prt]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="Keyboard Port"
"ErrorControl"=dword:00000001
"Tag"=dword:00000002
"DisplayName"="i8042 Keyboard and PS/2 Mouse Port
Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,4
9,56,45,52,53,5c,69,38,30,\
  34,32,70,72,74,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\i8042prt\Parameters]
"ResendIterations"=dword:00000003
"PollingIterations"=dword:00002ee0
"PollingIterationsMaximum"=dword:00002ee0
"PollStatusIterations"=dword:00000001
"KeyboardDataQueueSize"=dword:00000064
"KeyboardDeviceBaseName"="KeyboardPort"
"MouseDataQueueSize"=dword:00000064
"NumberOfButtons"=dword:00000002
"SampleRate"=dword:0000003c
"MouseResolution"=dword:00000003
"PointerDeviceBaseName"="PointerPort"
"MouseSynchIn100ns"=dword:01312d00
"EnableWheelDetection"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\i8042prt\Description]
"\\Device\\KeyboardPort0"=hex:01,00,00,00,00,00,00,00
,16,00,00,00,00,00,00,00,\
  20,00,00,00,00,00,00,00
"\\Device\\PointerPort0"=hex:01,00,00,00,00,00,00,00,
15,00,00,00,00,00,00,00,\
  1f,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\i8042prt\Enum]
"0"="Root\\*PNP030b\\1_0_22_0_32_0"
"Count"=dword:00000005
"NextInstance"=dword:00000005
"1"="Root\\*PNP030b\\1_15_22_0_32_0"
"2"="Root\\*PNP0F03\\1_0_21_0_31_0"
"3"="Root\\*PNP0F03\\1_15_21_0_31_0"
"4"="Root\\*PNP0F12\\1_0_21_0_31_0"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\inetaccs]
@=""

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\inetaccs\Parameters]
@=""

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Inport]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Pointer Port"
"ErrorControl"=dword:00000000
"Tag"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Inport\Parameters]
"MouseDataQueueSize"=dword:00000064
"NumberOfButtons"=dword:00000002
"SampleRate"=dword:00000032
"HzMode"=dword:00000002
"PointerDeviceBaseName"="PointerPort"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Jazzg300]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Jazzg300\Device0]
"VgaCompatible"=dword:00000000
"Monitor"="drivers\\jazz.bvp"
"InstalledDisplayDrivers"=hex(7):66,72,61,6d,65,62,75
,66,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Jazzg364]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Jazzg364\Device0]
"VgaCompatible"=dword:00000000
"Monitor"="drivers\\jazz.bvp"
"InstalledDisplayDrivers"=hex(7):66,72,61,6d,65,62,75
,66,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Jzvxl484]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Jzvxl484\Device0]
"VgaCompatible"=dword:00000000
"Monitor"="drivers\\jazz.bvp"
"InstalledDisplayDrivers"=hex(7):6a,7a,76,78,6c,34,38
,34,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Kbdclass]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="Keyboard Class"
"ErrorControl"=dword:00000001
"Tag"=dword:00000001
"DisplayName"="Keyboard Class Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,4
9,56,45,52,53,5c,6b,62,64,\
  63,6c,61,73,73,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Kbdclass\Parameters]
"KeyboardDataQueueSize"=dword:00000064
"MaximumPortsServiced"=dword:00000003
"ConnectMultiplePorts"=dword:00000000
"KeyboardDeviceBaseName"="KeyboardClass"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Kbdclass\Enum]
"0"="Root\\LEGACY_KBDCLASS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\KSecDD]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="Base"
"ErrorControl"=dword:00000001
"Tag"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\KSecDD\Enum]
"0"="Root\\LEGACY_KSECDD\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanServer]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="Server"
"ObjectName"="LocalSystem"
"DependOnService"=hex(7):00
"DependOnGroup"=hex(7):54,44,49,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanServer\AutotunedParameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanServer\Linkage]
"Bind"=hex(7):5c,44,65,76,69,63,65,5c,4e,65,74,42,54,
5f,4e,31,30,30,31,00,00
"Export"=hex(7):5c,44,65,76,69,63,65,5c,4c,61,6e,6d,6
1,6e,53,65,72,76,65,72,5c,\
  4e,65,74,42,54,5f,4e,31,30,30,31,00,00
"Route"=hex(7):22,4e,65,74,42,54,22,20,22,4e,31,30,30
,22,20,22,4e,31,30,30,31,\
  22,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanServer\Linkage\Disabled]
"Bind"=hex(7):00
"Export"=hex(7):00
"Route"=hex(7):00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanServer\Parameters]
"NullSessionPipes"=hex(7):43,4f,4d,4e,41,50,00,43,4f,
4d,4e,4f,44,45,00,53,51,\

4c,5c,51,55,45,52,59,00,53,50,4f,4f,4c,53,53,00,4c,4c
,53,52,50,43,00,45,50,\
  4d,41,50,50,45,52,00,4c,4f,43,41,54,4f,52,00,00
"NullSessionShares"=hex(7):43,4f,4d,43,46,47,00,44,46
,53,24,00,00
"Size"=dword:00000003
"Lmannounce"=dword:00000000
"MaxMpxCt"=dword:000001ff

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanServer\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,00,00,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,c8,00,14,00,00,00
,1c,00,ff,01,0f,00,01,\
02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,c8,00,14
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,c8,00,14,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanServer\Shares]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanServer\Shares\Security]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanServer\Enum]
"0"="Root\\LEGACY_LANMANSERVER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanWorkstation]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="Workstation"
"Group"="NetworkProvider"
"ObjectName"="LocalSystem"
"DependOnService"=hex(7):00
"DependOnGroup"=hex(7):54,44,49,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanWorkstation\Linkage]
"Bind"=hex(7):5c,44,65,76,69,63,65,5c,4e,65,74,42,54,
5f,4e,31,30,30,31,00,00
"Export"=hex(7):5c,44,65,76,69,63,65,5c,4c,61,6e,6d,6
1,6e,57,6f,72,6b,73,74,61,\

74,69,6f,6e,5c,4e,65,74,42,54,5f,4e,31,30,30,31,00,00
"Route"=hex(7):22,4e,65,74,42,54,22,20,22,4e,31,30,30
,22,20,22,4e,31,30,30,31,\
  22,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanWorkstation\Linkage\Disabled]
"Bind"=hex(7):00
"Export"=hex(7):00
"Route"=hex(7):00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanWorkstation\networkprovider]
"Devicename"="\\Device\\LanmanRedirector"
"ProviderPath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,7
4,25,5c,53,79,73,74,65,6d,\
  33,32,5c,6e,74,6c,61,6e,6d,61,6e,2e,64,6c,6c,00
"Name"="Microsoft Windows Network"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanWorkstation\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanWorkstation\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,25,02,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,c8,00,14,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,c8,00,14
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,c8,00,14,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LanmanWorkstation\Enum]
"0"="Root\\LEGACY_LANMANWORKSTATION\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LicenseInfo]
"ErrorControl"=dword:00000001
"Start"=dword:00000003
"Type"=dword:00000004

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LicenseInfo\FilePrint]
"Mode"=dword:00000001
"ConcurrentLimit"=dword:000003e8
"FlipAllow"=dword:00000001
"FamilyDisplayName"="Windows NT Server"
"DisplayName"="Windows NT Server"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LicenseInfo\MSSQL7.00]
"DisplayName"="Microsoft SQL Server 7.0"
"ConcurrentLimit"=dword:000003e8
"FlipAllow"=dword:00000000
"FamilyDisplayName"="Microsoft SQL Server"
"Mode"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LicenseService]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,53,79,73,74,65,6d,33,\
  32,5c,6c,6c,73,73,72,76,2e,65,78,65,00
"DisplayName"="License Logging Service"
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LicenseService\FilePrint]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LicenseService\FilePrint\KSecDD]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LicenseService\FilePrint\MSAfpSrv]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LicenseService\FilePrint\SMBServer]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LicenseService\FilePrint\TCP/IP Print Server]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LicenseService\Parameters]
"UseEnterprise"=dword:00000000
"ReplicationType"=dword:00000000
"ReplicationTime"=dword:00015180
"EnterpriseServer"=""

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LicenseService\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,6f,00,72,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,74,00,00,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,74,00,00
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,74,00,00,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LicenseService\Enum]
"0"="Root\\LEGACY_LICENSESERVICE\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LmHosts]
"Type"=dword:00000020
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="TCP/IP NetBIOS Helper"
"DependOnService"=hex(7):00
"DependOnGroup"=hex(7):4e,65,74,77,6f,72,6b,50,72,6f,
76,69,64,65,72,00,00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LmHosts\Linkage]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LmHosts\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LmHosts\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LmHosts\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,30,30,30,30,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,30,30,30,30,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,30,30,30
,30,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,30,30,30,30,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\LmHosts\Enum]
"0"="Root\\LEGACY_LMHOSTS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\macdisk]
"Type"=dword:00000001
"Start"=dword:00000004
"ErrorControl"=dword:00000001
"Tag"=dword:00000001
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5
c,53,79,73,74,65,6d,33,32,\

5c,64,72,69,76,65,72,73,5c,6d,61,63,64,69,73,6b,2e,73
,79,73,00
"Group"="Filter"
"PlugPlayServiceType"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\macdisk\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,74,00,6c,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,00,00,15,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,00,00,15
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,00,00,15,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Messenger]
"Type"=dword:00000020
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="Messenger"
"DependOnService"=hex(7):4c,61,6e,6d,61,6e,57,6f,72,6
b,73,74,61,74,69,6f,6e,00,\
  4e,65,74,42,69,6f,73,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Messenger\Linkage]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Messenger\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Messenger\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Messenger\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,25,02,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,c8,00,14,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,c8,00,14
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,c8,00,14,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Messenger\Enum]
"0"="Root\\LEGACY_MESSENGER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\mga]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\mga\Device0]
"VgaCompatible"=dword:00000000
"InstalledDisplayDrivers"=hex(7):6d,67,61,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\mga_mil]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\mga_mil\Device0]
"VgaCompatible"=dword:00000000
```

```
"InstalledDisplayDrivers"=hex(7):6d,67,61,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\mitsumi]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:0000001f

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\mkecr5xx]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:0000001d

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Modem]
"Type"=dword:00000001
"Start"=dword:00000003
"Group"="Extended base"
"ErrorControl"=dword:00000000
"Tag"=dword:00000004
"PlugPlayServiceType"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Modem\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Monitor]
"Type"=dword:00000010
"Start"=dword:00000004
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,73,79,73,74,65,6d,33,\
  32,5c,64,61,74,61,6c,6f,67,2e,65,78,65,00
"DisplayName"="Monitor Service"
"ObjectName"="LocalSystem"
"DataTimeOut"=dword:00004e20
"SettingFile"="TIGGERPERF"
"MonitorDuplicateInstances"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Monitor\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,74,00,6f,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,72,00,00,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,72,00,00
,00,00,00,1c,00,ff,01,\
```

```
0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,72,00,00,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Monitor\Enum]
"0"="Root\\LEGACY_MONITOR\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Mouclass]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="Pointer Class"
"ErrorControl"=dword:00000001
"Tag"=dword:00000001
"DisplayName"="Mouse Class Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,4
9,56,45,52,53,5c,6d,6f,75,\
  63,6c,61,73,73,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Mouclass\Parameters]
"MouseDataQueueSize"=dword:00000064
"MaximumPortsServiced"=dword:00000003
"ConnectMultiplePorts"=dword:00000001
"PointerDeviceBaseName"="PointerClass"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Mouclass\Enum]
"0"="Root\\LEGACY_MOUCLASS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\MSDTC]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,57,49,4e,4e,54,34,30,5c,5
3,79,73,74,65,6d,33,32,5c,\
  6d,73,64,74,63,2e,65,78,65,00
"DisplayName"="MSDTC"
"Group"="MS Transactions"
"DependOnService"=hex(7):52,50,43,53,53,00,4e,54,4c,4
d,53,53,50,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Tag"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\MSDTC\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\
```

```
00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,00,00,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,00,00,00,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,00,00,00
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,00,00,00,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Msfs]
"Type"=dword:00000002
"Start"=dword:00000001
"Group"="File system"
"ErrorControl"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Msfs\Enum]
"0"="Root\\LEGACY_MSFS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\MSSQLServer]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,4d,53,53,51,4c,37,5c,62,6
9,6e,6e,5c,73,71,6c,73,65,\
  72,76,72,2e,65,78,65,00
"DisplayName"="MSSQLServer"
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\MSSQLServer\Performance]
"Collect"="CollectSQLPerformanceData"
"Library"="SQLCTR70.DLL"
"Close"="CloseSQLPerformanceData"
"Open"="OpenSQLPerformanceData"
"Last Counter"=dword:0000083a
"Last Help"=dword:0000083b
"First Counter"=dword:00000738
"First Help"=dword:00000739

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\MSSQLServer\Security]
```

```
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,00,00,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,00,00,00,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,00,00,00
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,00,00,00,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\MSSQLServer\Enum]
"0"="Root\\LEGACY_MSSQLSERVER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Mup]
"Type"=dword:00000002
"Start"=dword:00000003
"DisplayName"="Mup"
"Group"="Network"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5
c,53,79,73,74,65,6d,33,32,\
  5c,64,72,69,76,65,72,73,5c,6d,75,70,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Mup\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Mup\Enum]
"0"="Root\\LEGACY_MUP\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\N100]
"Type"=dword:00000001
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5
c,53,79,73,74,65,6d,33,32,\

5c,44,72,69,76,65,72,73,5c,4e,31,30,30,4e,54,2e,73,79
,73,00
```

```
"DisplayName"="Compaq Ethernet or Fast Ethernet NIC
Driver"
"Group"="NDIS"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\N100\Linkage]
"Bind"=hex(7):5c,44,65,76,69,63,65,5c,4e,31,30,30,31,
00,00
"Export"=hex(7):5c,44,65,76,69,63,65,5c,4e,31,30,30,3
1,00,00
"Route"=hex(7):22,4e,31,30,30,31,22,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\N100\Linkage\Disabled]
"Bind"=hex(7):00
"Export"=hex(7):00
"Route"=hex(7):00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\N100\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\N100\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,51,00,4b,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,44,00,44,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,44,00,44
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,44,00,44,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\N100\Enum]
"0"="Root\\LEGACY_N100\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\N1001]
"Type"=dword:00000004
"Start"=dword:00000003
"ErrorControl"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\N1001\Linkage]
"Bind"=hex(7):5c,44,65,76,69,63,65,5c,4e,31,30,30,31,
00,00
"Export"=hex(7):5c,44,65,76,69,63,65,5c,4e,31,30,30,3
1,00,00
"Route"=hex(7):22,4e,31,30,30,31,22,00,00,00,00,00,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\N1001\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\N1001\Parameters]
"NetworkAddress"=""
"Adaptive_IFS"=dword:00000001
"NumCoalesce"=dword:00000008
"NumRfd"=dword:00000020
"NumTbdPerTcb"=dword:0000000c
"NumTcb"=dword:00000010
"Threshold"=dword:00000010
"UnderrunRetry"=dword:00000001
"LogErrorMessages"=dword:00000001
"Txmitwait"=dword:00000001
"RxDmaCount"=dword:00000000
"RxFifo"=dword:0000000c
"TxDmaCount"=dword:00000000
"TxFifo"=dword:00000000
"UseManualPciAssign"=dword:00000001
"BoardHasBridge"=dword:00000000
"BusTypeLocal"=dword:00000005
"UcodeSW"=dword:00000001
"MsPciScan"=dword:00000001
"MediaType"=dword:00000001
"BusType"=dword:00000005
"PerfOptims"=dword:00000000
"MapRegisters"=dword:00000040
"ForceDpx"=dword:00000001
"Speed"=dword:00000064
"PhySlotNumber"=dword:0000ffff
"BusNumber"=dword:00000008
"SlotNumber"=dword:00000004
"IRQ"=dword:00000010
"IoBaseAddress"=dword:00005c00
"MemoryMappedBaseAddress"=dword:f7cf0000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\N1001\Parameters\Tcpip]
"EnableDHCP"=dword:00000000
"UseZeroBroadcast"=dword:00000000
"LLInterface"=""
"PPTPFiltering"=dword:00000000
"IPAddress"=hex(7):31,33,30,2e,31,36,38,2e,32,30,31,2
e,31,30,39,00,00
"SubnetMask"=hex(7):32,35,35,2e,32,35,35,2e,30,2e,30,
00,00
"TCPAllowedPorts"=hex(7):30,00,00
"UDPAllowedPorts"=hex(7):30,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00
"DefaultGateway"=hex(7):00,00
"IPInterfaceContext"=dword:00000001
"IPInterfaceContextMax"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ncr53c9x]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ncr77c22]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ncr77c22\Device0]
"VgaCompatible"=dword:00000001
"InstalledDisplayDrivers"=hex(7):76,67,61,00,76,67,61
,32,35,36,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ncrc700]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000004

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ncrc710]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000005

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ncrc710\Enum]
"0"="Root\\LEGACY_NCRC710\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="NDIS"
"DisplayName"="Microsoft NDIS System Driver"
"ErrorControl"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\MediaTypes]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\3C592]
"Id"=dword:20596d50
"Mask"=dword:f0ffffff
"token"="3C592"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\3C597]
"Id"=dword:70596d50
"Mask"=dword:f0ffffff
"token"="3C597"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\BONSAI]
"Id"=dword:0062110e
"Mask"=dword:00ffffff
"token"="BONSAI"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\C320TNT]
"Id"=dword:0032530e
"Mask"=dword:00ffffff
"token"="C320TNT"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\DE425]
"Id"=dword:5042a310
"Mask"=dword:f0ffffff
"token"="DE425"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\DEC300]
"Id"=dword:0230a310
"Mask"=dword:0fffffff
"token"="DEC300"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\DEC422]
"Id"=dword:2042a310
"Mask"=dword:f0ffffff
"token"="DEC422"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\DURANGO]
"Id"=dword:0260110e
"Mask"=dword:0fffffff
"token"="DURANGO"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\ELNK3EISA]
"Id"=dword:90506d50
"Mask"=dword:f0ffffff
"token"="ELNK3EISA"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\ES3210]
"Id"=dword:00012949
"Mask"=dword:00ffffff
"token"="ES3210"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\F70XX]
"Id"=dword:0006690e
"Mask"=dword:00ffffff
"token"="F70XX"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\FL32]
"Id"=dword:1010d425
"Mask"=dword:ffffffff
"token"="FL32"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\FLNK]
"Id"=dword:00776d50
"Mask"=dword:00ffffff
"token"="FLNK"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\J2577A]
"Id"=dword:4019f022
"Mask"=dword:f0ffffff
"token"="J2577A"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\MAPLE]
"Id"=dword:0160110e
"Mask"=dword:0fffffff
"token"="MAPLE"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\NE3200]
"Id"=dword:0007cc3a
"Mask"=dword:00ffffff
"token"="NE3200"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\NETFLEX3]
"Id"=dword:20f1110e
"Mask"=dword:f0ffffff
"token"="NETFLEX3"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\NETFLEX3.1]
"Id"=dword:40f1110e
"Mask"=dword:f0ffffff
"token"="NETFLEX3"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\NETFLX]
"Id"=dword:0061110e
"Mask"=dword:00ffffff
"token"="NETFLX"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\NF3500]
"Id"=dword:0084633a
"Mask"=dword:00ffffff
"token"="NF3500"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\NPEISA.1]
"id"=dword:0002093a
"Mask"=dword:00ffffff
"token"="NPEISA"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\NPEISA.2]
"id"=dword:0003093a
```

```
"Mask"=dword:00ffffff
"token"="NPEISA"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\P1990]
"Id"=dword:00604f42
"Mask"=dword:00ffffff
"token"="P1990"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\RODAN]
"Id"=dword:0063110e
"Mask"=dword:00ffffff
"token"="RODAN"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\SKETHNT]
"Id"=dword:0002644d
"Mask"=dword:000fffff
"token"="SKETHNT"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\SKFENT]
"Id"=dword:0001644d
"Mask"=dword:000fffff
"token"="SKFENT"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\SMC8232]
"Id"=dword:0080a34d
"Mask"=dword:00ffffff
"token"="SMC8232"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\TLNK3E]
"Id"=dword:9c616d50
"Mask"=dword:f0ffffff
"token"="TLNK3E"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\EISA\TLNK3EISA]
"Id"=dword:90616d50
"Mask"=dword:f0ffffff
"token"="TLNK3EISA"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\AT1700]
"Id"=dword:00006413
"token"="AT1700"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\EE16MC]
"token"="EE16MC"
"Id"=dword:0000628b

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\ElINK527]
"Id"=dword:00000041
"token"="ElINK527"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\ELNK3MCA.1]
"token"="ELNK3MCA"
"Id"=dword:0000627c

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\ELNK3MCA.2]
"token"="ELNK3MCA"
"Id"=dword:0000627d

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\ELNK3MCA.3]
"token"="ELNK3MCA"
"Id"=dword:000061db

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\ELNK3MCA.4]
"token"="ELNK3MCA"
"Id"=dword:000062f6

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\ELNK3MCA.5]
"token"="ELNK3MCA"
"Id"=dword:000062f7

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\ELNKMC]
"token"="ELNKMC"
"Id"=dword:00006042

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\F30XX]
"Id"=dword:00000070
"token"="F30XX"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\HPMCA]
"Id"=dword:000063ca
"token"="HPMCA"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\IBMENIIN]
"Id"=dword:0000ffe0
"token"="IBMENIIN"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\IBMTOKA]
"token"="IBMTOKA"
"Id"=dword:0000e000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\IBMTOKMC]
"token"="IBMTOKMC"
"Id"=dword:0000e001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\IRMAtrac.1]
"Id"=dword:00005c1c
"token"="IRMAtrac"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\IRMAtrac.2]
"Id"=dword:00005c1d
"token"="IRMAtrac"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\NCRTOK]
"Id"=dword:00000100
"token"="NCRTOK"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\NPMCA]
"token"="NPMCA"
"Id"=dword:00000069

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\OCTK16.1]
"Id"=dword:00000a84
"token"="OCTK16"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\OCTK16.2]
"Id"=dword:00000a85
"token"="OCTK16"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\OCTK16.3]
"Id"=dword:00000a86
"token"="OCTK16"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\QUADENET.1]
"Id"=dword:00008f6d
"token"="QUADENET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\QUADENET.2]
"Id"=dword:00008f6a
"token"="QUADENET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\SKFMNT.1]
"Id"=dword:00000083
"token"="SKFMNT"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\SKFMNT.2]
"Id"=dword:000000ab
"token"="SKFMNT"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\STREAMER.1]
"Id"=dword:00008fa0
"token"="STREAMER"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\STREAMER.2]
"Id"=dword:00008fa2
"token"="STREAMER"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\STREAMER.3]
"Id"=dword:00008fa8
"token"="STREAMER"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\STREAMER.4]
"Id"=dword:00008faa
"token"="STREAMER"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\TC$4046E]
"Id"=dword:00000051
"token"="TC$4046E"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\UBPS]
"token"="UBPS"
"Id"=dword:00007012

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\WAVELAN_MCA]
"Id"=dword:00006a14
"token"="WAVELAN_MCA"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\WD8003EA]
"token"="WD8003EA"
"Id"=dword:000067c0

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\WD8003WA]
"token"="WD8003WA"
"Id"=dword:000067c2

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\WD8013EPA]
"token"="WD8013EPA"
"Id"=dword:000061c8

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\MCA\WD8013WPA]
"token"="WD8013WPA"
"Id"=dword:000061c9

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\3C590]
"Id"=dword:590010b7
"token"="3C590"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\3C595]
"Id"=dword:595010b7
"token"="3C595"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\3C905]
"Id"=dword:905010b7
"token"="3C905"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\ALANE0]
"Id"=dword:59009004
"token"="ALANE0"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\AMDPCI]
"token"="AMDPCI"
"Id"=dword:20001022

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\DC21040]
"token"="DC21040"
"Id"=dword:00021011

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\DC21041]
"token"="DC21041"
"Id"=dword:00141011

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\DC21140]
"token"="DC21140"
"Id"=dword:00091011

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\DC21142]
"token"="DC21142"
"Id"=dword:00191011

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\DEFPA]
"token"="DEFPA"
"Id"=dword:000f1011

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\E100BPCI]
"Id"=dword:12298086
"token"="E100BPCI"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\E10PCI]
"Id"=dword:12268086
"token"="E10PCI"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\LEC]
"Id"=dword:100110b6
"token"="LEC"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\NCPF]
"Id"=dword:000111bc
"token"="NCPF"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\NETFLEX3.1]
"Id"=dword:f1300e11
"token"="NETFLEX3"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\NETFLEX3.2]
"Id"=dword:ae320e11
"token"="NETFLEX3"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\NETFLEX3.3]
"Id"=dword:ae340e11
"token"="NETFLEX3"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\NETFLEX3.4]
"Id"=dword:ae350e11
"token"="NETFLEX3"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\NETFLEX3.5]
"Id"=dword:ae430e11
"token"="NETFLEX3"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\NETFLEX3.6]
"Id"=dword:ae400e11
"token"="NETFLEX3"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\NETFLEX3.7]
"Id"=dword:f1500e11
"token"="NETFLEX3"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\O100PCI]
"Id"=dword:0011108d
"token"="O100PCI"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\OCE4XMP]
"Id"=dword:0013108d
"token"="OCE4XMP"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\OCTK16]
"Id"=dword:0001108d
"token"="OCTK16"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\RNSFDDI]
"Id"=dword:22001112
"token"="RNSFDDI"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\RTL8029]
"Id"=dword:802910ec
"token"="RTL8029"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\SKFPNT]
"Id"=dword:40001148
"token"="SKFPNT"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\SKTOKNT_PCI]
"Id"=dword:42001148
"token"="SKTOKNT_PCI"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\NetDetect\PCI\STREAMER]
"Id"=dword:00181014
"token"="STREAMER"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\Parameters]
"ProcessorAffinityMask"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NDIS\Enum]
"0"="Root\\LEGACY_NDIS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBIOS]
"Type"=dword:00000002
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5
c,53,79,73,74,65,6d,33,32,\

5c,64,72,69,76,65,72,73,5c,6e,65,74,62,69,6f,73,2e,73
,79,73,00
"DisplayName"="NetBIOS Interface"
"Group"="NetBIOSGroup"
"DependOnService"=hex(7):00
"DependOnGroup"=hex(7):54,44,49,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBIOS\Linkage]
"LanaMap"=hex:01,00
"Bind"=hex(7):5c,44,65,76,69,63,65,5c,4e,65,74,42,54,
5f,4e,31,30,30,31,00,00
"Export"=hex(7):5c,44,65,76,69,63,65,5c,4e,65,74,62,6
9,6f,73,5c,4e,65,74,42,54,\
  5f,4e,31,30,30,31,00,00
"Route"=hex(7):22,4e,65,74,42,54,22,20,22,4e,31,30,30
,22,20,22,4e,31,30,30,31,\
  22,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBIOS\Linkage\Disabled]
"Bind"=hex(7):00
"Export"=hex(7):00
"Route"=hex(7):00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBIOS\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBIOS\Parameters\Winsock]
"HelperDllName"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,
74,25,5c,73,79,73,74,65,\
  6d,33,32,5c,77,73,68,6e,65,74,62,73,2e,64,6c,6c,00
"MaxSockAddrLength"=dword:00000014
"MinSockAddrLength"=dword:00000014
"Mapping"=hex:02,00,00,00,03,00,00,00,11,00,00,00,05,
00,00,00,00,00,00,00,11,\
  00,00,00,02,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBIOS\Security]
"Security"=hex:01,00,14,80,cc,00,00,00,d8,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,98,00,06,00,00,00,00,03,18,00
,00,00,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,00,00,00,00,00,03,18,00,ff
,01,0f,00,01,02,00,00,\

00,00,00,05,20,00,00,00,20,02,00,00,00,03,18,00,ff,01
,0f,00,01,01,00,00,00,\

00,00,05,12,00,00,00,20,02,00,00,00,03,18,00,00,00,02
,00,01,02,00,00,00,00,\

00,05,20,00,00,00,23,02,00,00,00,03,18,00,9d,00,00,00
,01,01,00,00,00,00,00,\

05,04,00,00,00,23,02,00,00,00,03,18,00,9d,00,00,00,01
,02,00,00,00,00,00,05,\

20,00,00,00,21,02,00,00,01,01,00,00,00,00,00,05,12,00
,00,00,01,01,00,00,00,\
  00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBIOS\Enum]
"0"="Root\\LEGACY_NETBIOS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBIOSInformation]
"Type"=dword:00000004
"Start"=dword:00000003
"ErrorControl"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBIOSInformation\Linkage]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBIOSInformation\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBIOSInformation\Parameters]
"Route"=hex(7):22,4e,65,74,42,54,22,20,22,4e,31,30,30
,22,20,22,4e,31,30,30,31,\
  22,00,00
"MaxLana"=dword:00000000
"LanaNum1"=dword:00000000
"EnumExport1"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBT]
"Type"=dword:00000001
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5
c,53,79,73,74,65,6d,33,32,\

5c,64,72,69,76,65,72,73,5c,6e,65,74,62,74,2e,73,79,73
,00
"DisplayName"="WINS Client(TCP/IP)"
"Group"="PNP_TDI"
"DependOnService"=hex(7):54,63,70,69,70,00,00
"DependOnGroup"=hex(7):00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBT\Adapters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBT\Adapters\N1001]
"NameServer"=""
"NameServerBackup"=""

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBT\Linkage]
"OtherDependencies"=hex(7):54,63,70,69,70,00,00
"Bind"=hex(7):5c,44,65,76,69,63,65,5c,4e,31,30,30,31,
00,00
"Export"=hex(7):5c,44,65,76,69,63,65,5c,4e,65,74,42,5
4,5f,4e,31,30,30,31,00,00
"Route"=hex(7):22,4e,31,30,30,22,20,22,4e,31,30,30,31
,22,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBT\Linkage\Disabled]
"Bind"=hex(7):00
"Export"=hex(7):00
"Route"=hex(7):00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBT\Parameters]
"NbProvider"="_tcp"
"NameServerPort"=dword:00000089
"EnableLMHOSTS"=dword:00000001
"EnableProxy"=dword:00000000
"EnableDNS"=dword:00000000
"CacheTimeout"=dword:000927c0
"BcastNameQueryCount"=dword:00000003
"BcastQueryTimeout"=dword:000002ee
"NameSrvQueryCount"=dword:00000003
"NameSrvQueryTimeout"=dword:000005dc
"Size/Small/Medium/Large"=dword:00000001
"ScopeID"=""
"TransportBindName"="\\Device\\"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBT\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,25,02,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,00,00,00,00,00,00
,1c,00,ff,01,0f,00,01,\
02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,00,00,00
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,00,00,00,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetBT\Enum]
"0"="Root\\LEGACY_NETBT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetDDE]
"Type"=dword:00000020
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,73,79,73,74,65,6d,33,\
  32,5c,6e,65,74,64,64,65,2e,65,78,65,00
"DependOnService"=hex(7):4e,65,74,44,44,45,44,53,44,4
d,00,00
"DisplayName"="Network DDE"
"ObjectName"="LocalSystem"
"Group"="NetDDEGroup"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetDDE\Security]
"Security"=hex:01,00,14,80,9c,00,00,00,a8,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,68,00,04,00,00,00,00,03,18,00
,8d,00,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,00,00,00,00,00,03,18,00,ff
,01,0f,00,01,02,00,00,\

00,00,00,05,20,00,00,00,20,02,00,00,00,03,18,00,8f,00
,02,00,01,02,00,00,00,\

00,00,05,20,00,00,00,23,02,00,00,00,03,18,00,9d,00,00
,00,01,01,00,00,00,00,\

00,05,04,00,00,00,23,02,00,00,01,01,00,00,00,00,00,05
,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetDDEdsdm]
"Type"=dword:00000020
"Start"=dword:00000003
"ErrorControl"=dword:00000001
```

```
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,73,79,73,74,65,6d,33,\
  32,5c,6e,65,74,64,64,65,2e,65,78,65,00
"DependOnService"=hex(7):00,00
"DisplayName"="Network DDE DSDM"
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetDDEdsdm\Security]
"Security"=hex:01,00,14,80,9c,00,00,00,a8,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,68,00,04,00,00,00,00,03,18,00
,8d,00,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,00,00,00,00,00,03,18,00,ff
,01,0f,00,01,02,00,00,\

00,00,00,05,20,00,00,00,20,02,00,00,00,03,18,00,8f,00
,02,00,01,02,00,00,00,\

00,00,05,20,00,00,00,23,02,00,00,00,03,18,00,9d,00,00
,00,01,01,00,00,00,00,\

00,05,04,00,00,00,23,02,00,00,01,01,00,00,00,00,00,05
,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetDetect]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"DisplayName"="NetDetect"
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5
c,73,79,73,74,65,6d,33,32,\

5c,64,72,69,76,65,72,73,5c,6e,65,74,64,74,65,63,74,2e
,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NetDetect\Enum]
"0"="Root\\LEGACY_NETDETECT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Netlogon]
"Type"=dword:00000020
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,53,79,73,74,65,6d,33,\
  32,5c,6c,73,61,73,73,2e,65,78,65,00
"DisplayName"="Net Logon"
"Group"="RemoteValidation"
"DependOnService"=hex(7):4c,61,6e,6d,61,6e,57,6f,72,6
b,73,74,61,74,69,6f,6e,00,\
  4c,6d,48,6f,73,74,73,00,00
"DependOnGroup"=hex(7):00
```

```
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Netlogon\Linkage]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Netlogon\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Netlogon\Parameters]
"DisablePasswordChange"=dword:00000000
"Update"="no"
"Scripts"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,
5c,73,79,73,74,65,6d,33,\

32,5c,72,65,70,6c,5c,69,6d,70,6f,72,74,5c,73,63,72,69
,70,74,73,00
"DBFlag"="0"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Netlogon\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,25,02,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,c8,00,14,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,c8,00,14
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,c8,00,14,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Npfs]
"Type"=dword:00000002
"Start"=dword:00000001
"Group"="File system"
"ErrorControl"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Npfs\Aliases]
"lsass"=hex(7):6e,65,74,6c,6f,67,6f,6e,00,6c,73,61,72
,70,63,00,73,61,6d,72,00,\
  00
"ntsvcs"=hex(7):73,72,76,73,76,63,00,77,6b,73,73,76,6
3,00,65,76,65,6e,74,6c,6f,\
67,00,62,72,6f,77,73,65,72,00,6d,73,67,73,76,63,00,73
,76,63,63,74,6c,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Npfs\Enum]
"0"="Root\\LEGACY_NPFS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ntfs]
"Type"=dword:00000002
"Start"=dword:00000004
"Group"="File system"
"ErrorControl"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ntfs\Enum]
"0"="Root\\LEGACY_NTFS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NtLmSsp]
"Type"=dword:00000020
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,53,79,73,74,65,6d,33,\
  32,5c,53,45,52,56,49,43,45,53,2e,45,58,45,00
"DisplayName"="NT LM Security Support Provider"
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NtLmSsp\Linkage]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NtLmSsp\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NtLmSsp\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NtLmSsp\Security]
"Security"=hex:01,00,14,80,cc,00,00,00,d8,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,98,00,06,00,00,00,00,03,18,00
,00,00,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,00,00,00,00,00,03,18,00,ff
,01,0f,00,01,02,00,00,\

00,00,00,05,20,00,00,00,20,02,00,00,00,03,18,00,ff,01
,0f,00,01,01,00,00,00,\

00,00,05,12,00,00,00,20,02,00,00,00,03,18,00,00,00,02
,00,01,02,00,00,00,00,\
```

```
00,05,20,00,00,00,23,02,00,00,00,03,18,00,9d,00,00,00
,01,01,00,00,00,00,00,\

05,04,00,00,00,23,02,00,00,00,03,18,00,9d,00,00,00,01
,02,00,00,00,00,00,05,\

20,00,00,00,21,02,00,00,01,01,00,00,00,00,00,05,12,00
,00,00,01,01,00,00,00,\
  00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NtLmSsp\Enum]
"0"="Root\\LEGACY_NTLMSSP\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Null]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="Base"
"ErrorControl"=dword:00000001
"Tag"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Null\Enum]
"0"="Root\\LEGACY_NULL\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NxPerf]
"Type"=dword:00000001
"Start"=dword:00000004
"ErrorControl"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NxPerf\Performance]
"Close"="CloseNxPerfData"
"Collect"="CollectNxPerfData"
"Library"="NxCtrs.dll"
"Open"="OpenNxPerfData"
"Last Counter"=dword:00000950
"Last Help"=dword:00000951
"First Counter"=dword:000008ee
"First Help"=dword:000008ef

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\NxPerf\Enum]
"0"="Root\\LEGACY_NXPERF\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Oliscsi]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000003
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Parallel]
"DependOnGroup"=hex(7):50,61,72,61,6c,6c,65,6c,20,61,
72,62,69,74,72,61,74,6f,\
  72,00,00
"DependOnService"=hex(7):50,61,72,70,6f,72,74,00,00
"Type"=dword:00000001
"Start"=dword:00000002
"Group"="Extended base"
"ErrorControl"=dword:00000000
"Tag"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Parallel\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Parallel\Enum]
"0"="Root\\LEGACY_PARALLEL\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Parport]
"Type"=dword:00000001
"Start"=dword:00000002
"Group"="Parallel arbitrator"
"ErrorControl"=dword:00000000
"Tag"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Parport\Enum]
"0"="Root\\LEGACY_PARPORT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ParVdm]
"DependOnGroup"=hex(7):50,61,72,61,6c,6c,65,6c,20,61,
72,62,69,74,72,61,74,6f,\
  72,00,00
"DependOnService"=hex(7):50,61,72,70,6f,72,74,00,00
"Type"=dword:00000001
"Start"=dword:00000002
"Group"="Extended base"
"ErrorControl"=dword:00000000
"Tag"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ParVdm\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ParVdm\Enum]
"0"="Root\\LEGACY_PARVDM\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\PCIDump]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="PCI Configuration"
"ErrorControl"=dword:00000000
"Tag"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia]
"ErrorControl"=dword:00000001
"Group"="System Bus Extender"
"Start"=dword:00000004
"Tag"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\3Com Corporation]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\3Com Corporation\3C589]
"Driver"="elnk3"
"Option"="ELNK3ISA509"
"InfFileName"="oemnade3.inf"
"DeviceType"="NET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Accton]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Accton\EN2212]
"DeviceType"="NET"
"Driver"="ne2000"
"InfFileName"="oemnadzz.inf"
"Option"="ACCTONEN2216"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Accton\EN2216-PCMCIA-ETHERNET]
"DeviceType"="NET"
"Driver"="ne2000"
"InfFileName"="oemnadzz.inf"
"Option"="ACCTONEN2216"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Adaptec, Inc.]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Adaptec, Inc.\APA-1460 SCSI Host
Adapter]
"Driver"="sparrow"
"DeviceType"="SCSI"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Compaq]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Compaq\Ethernet LAN Card]
"AttributeMemorySize"=dword:00001000
"DeviceType"="NET"
"Driver"="cpqndis"
"InfFileName"="oemnadzz.inf"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\D-Link]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\D-Link\DE-650]
"Driver"="ne2000"
"DeviceType"="NET"
"Option"="DLINKDE650"
"InfFileName"="oemnadzz.inf"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Diehl ISDN]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Diehl ISDN\ISDN-DIVA/PCM]
"DeviceType"="NET"
"Driver"="Diehl_DIVA_TED5ES"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Future Domain Corporation]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Future Domain Corporation\SCSI
PCMCIA Credit Card Controller]
"Driver"="fd16_700"
"DeviceType"="SCSI"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\IBM]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\IBM\TOKEN RING]
"Driver"="ibmtok"
"DeviceType"="NET"
"InfFileName"="oemnadtk.inf"
"Option"="IBMTOK"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\IBM Corp.]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\IBM Corp.\Ethernet]
"AttributeMemorySize"=dword:00001000
"CardMemorySize"=dword:00000000
"Driver"="ne2000"
"Option"="NE2000IBMCOMPAT"
"InfFileName"="oemnadni.inf"
"DeviceType"="NET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Kingston Technology Corp.]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Kingston Technology Corp.\EtheRx]
"AttributeMemorySize"=dword:00001000
"CardMemorySize"=dword:00000000
"Driver"="ne2000"
"Option"="NE2000IBMCOMPAT"
"InfFileName"="oemnadni.inf"
"DeviceType"="NET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\KME]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\KME\KXLC002]
"Driver"="qlmini"
"DeviceType"="SCSI"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\MADGE]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\MADGE\SMART 16/4 PCMCIA RINGNODE]
"Driver"="madgemport"
"Option"="MSMDGMPPCMCIA"
"InfFileName"="oemnadma.inf"
"DeviceType"="NET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Microdyne]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Microdyne\NE4200]
"AttributeMemorySize"=dword:00001000
"CardMemorySize"=dword:00000000
"Driver"="ne2000"
"Option"="NE2000IBMCOMPAT"
"InfFileName"="oemnadni.inf"
"DeviceType"="NET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\National Semiconductor]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\National Semiconductor\InfoMover
NE4100]
"AttributeMemorySize"=dword:00001000
"CardMemorySize"=dword:00000000
"Driver"="ne2000"
"Option"="NE2000IBMCOMPAT"
"InfFileName"="oemnadni.inf"
"DeviceType"="NET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\OLICOM]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\OLICOM\ETHERCOM]
"DeviceType"="NET"
"Driver"="oce2xm"
"Option"="GOCARD"
"InfFileName"="oemnadzz.inf"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\OLICOM\TR 16/4 II]
"DeviceType"="NET"
"Driver"="octk16"
"Option"="PCMCIA"
"InfFileName"="oemnadzz.inf"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Socket Communications Inc]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Socket Communications Inc\Socket EA
PCMCIA LAN Adapter Revision D]
"AttributeMemorySize"=dword:00000000
"CardMemorySize"=dword:00000000
"Driver"="ne2000"
"Option"="NE2000SOCKETEA"
```

```
"InfFileName"="oemnadn2.inf"
"DeviceType"="NET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Socket Communications Inc\Socket EA
PCMCIA LAN Adapter Revision E]
"AttributeMemorySize"=dword:00000000
"CardMemorySize"=dword:00000000
"Driver"="ne2000"
"Option"="NE2000SOCKETEA"
"InfFileName"="oemnadn2.inf"
"DeviceType"="NET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Xircom]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Xircom\CreditCard 10Base-T]
"Option"=""
"InfFileName"="oemnadzz.inf"
"DeviceType"="NET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Xircom\CreditCard 10Base-T\5331]
"Driver"="ce2xps"
"AttributeMemorySize"=dword:00001000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Xircom\CreditCard 10Base-T\5FFA]
"Driver"="cendis3"
"AttributeMemorySize"=dword:00001000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Xircom\CreditCard 10Base-T\64BA]
"AttributeMemorySize"=dword:00001000
"Driver"="ce2xps"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Xircom\CreditCard 10Base-T\9C9B]
"Driver"="cendis3"
"AttributeMemorySize"=dword:00001000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Xircom\CreditCard 10Base-T\9F5B]
"Driver"="ce2ndis3"
"AttributeMemorySize"=dword:00001000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Xircom\CreditCard 10Base-T\9FAB]
"Driver"="cendis3"
"AttributeMemorySize"=dword:00001000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Xircom\CreditCard 10Base-T\B00D]
"Driver"="cendis3"
"AttributeMemorySize"=dword:00001000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Xircom\CreditCard Ethernet+Modem
28.8]
"AttributeMemorySize"=dword:00001000
"DeviceType"="NET"
"Driver"="cem28xps"
```

```
"InfFileName"="oemnadzz.inf"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Xircom\CreditCard Ethernet+Modem II]
"Driver"="cm2ndis3"
"AttributeMemorySize"=dword:00001000
"Option"=""
"InfFileName"="oemnadzz.inf"
"DeviceType"="NET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\DataBase\Xircom\CreditCard Token Ring]
"Driver"="ctndnt"
"AttributeMemorySize"=dword:00001000
"Option"=""
"InfFileName"="oemnadzz.inf"
"DeviceType"="NET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Pcmcia\Parameters]
"Interrupt"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\PlugPlay]
"Type"=dword:00000020
"Start"=dword:00000003
"Group"="PlugPlay"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,73,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"ObjectName"="LocalSystem"
"DisplayName"="Plug and Play"
"PlugPlayServiceType"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\pnpisa]
"DisplayName"="PnP ISA Enabler Driver"
"ErrorControl"=dword:00000000
"Group"="Base"
"Start"=dword:00000004
"Tag"=dword:0000000e
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ProtectedStorage]
"Type"=dword:00000110
"Start"=dword:00000004
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):63,3a,5c,77,69,6e,6e,74,34,30,5c,7
3,79,73,74,65,6d,33,32,5c,\
  70,73,74,6f,72,65,73,2e,65,78,65,00
"DisplayName"="Protected Storage"
"DependOnService"=hex(7):52,70,63,53,73,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ProtectedStorage\Parameters]
"AuthCodeCfg"=dword:00000001
"Configuration"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ProtectedStorage\Parameters\S2]
"pstores.exe"=hex:85,88,4d,da,80,23,97,51,7d,ed,3f,28
,64,e7,34,cc,1b,ff,d9,8e,\

95,1b,60,d4,72,df,64,a3,e2,a3,7d,af,4b,1a,f3,19,cb,f3
,6d,f2,3d,bb,30,b6,fe,\

35,fe,66,1e,29,f5,a0,bc,d9,b3,07,9e,02,b9,ac,f2,37,70
,8b,4f,87,41,b8,a9,66,\

ab,cf,4d,1e,54,50,68,d0,51,a0,48,af,44,1c,23,b7,27,f8
,4a,7b,86,54,5a,42,86,\

e2,a3,c2,07,0f,ef,13,a2,72,b4,8a,e7,31,d4,4a,d3,ef,3a
,11,9d,51,2f,52,ec,c6,\
  6c,d3,28,43,45,30,e2,72,00,00,00,00,00,00,00,00
"psbase.dll"=hex:e6,ba,57,e2,1b,88,fc,20,01,ed,35,32,
7e,8f,36,97,41,4c,e1,76,\

97,ef,60,09,38,ea,86,6b,43,d6,30,4e,10,6b,c1,75,61,84
,d2,1b,c7,eb,b2,18,f1,\

13,cb,a0,6c,8f,31,65,8e,00,93,5e,72,78,97,61,63,d1,74
,9f,19,e6,b4,08,fc,ba,\

ba,df,b1,9e,da,17,19,5d,95,22,f9,f6,60,40,ae,c4,72,0b
,d1,74,7c,86,a8,8a,f8,\

d2,ba,14,63,f2,a5,25,b9,a2,b7,86,58,1e,4e,92,28,cf,41
,6a,b7,57,4c,44,67,51,\
  48,47,85,f5,6e,00,99,af,00,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ProtectedStorage\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,00,00,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,00,00,00,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,00,00,00
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,00,00,00,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ProtectedStorage\Enum]
"0"="Root\\LEGACY_PROTECTEDSTORAGE\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\PSE36]
"Type"=dword:00000001
"Start"=dword:00000004
"ErrorControl"=dword:00000001
"Group"="Primary disk"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\PSE36\Parameters]
"BreakOnEntry"=dword:00000000
"DebugLevel"=dword:00000000
"UseOnlyLowMemory"=dword:00000000
"DiskSpace"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\PSE36\Performance]
"Library"="pse36prf.dll"
"Open"="OpenPSE36PerformanceData"
"Collect"="CollectPSE36PerformanceData"
"Close"="ClosePSE36PerformanceData"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\PSE36\Enum]
"0"="Root\\LEGACY_PSE36\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\psidisp]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\psidisp\Device0]
"VgaCompatible"=dword:00000000
"InstalledDisplayDrivers"=hex(7):70,73,69,64,69,73,70
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ql10wnt]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000023

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\qv]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\qv\Device0]
```

```
"VgaCompatible"=dword:00000001
"InstalledDisplayDrivers"=hex(7):76,67,61,00,71,76,00
,00
"HardwareInformation.AdapterString"=hex:51,00,56,00,4
7,00,41,00,20,00,4e,00,65,\
  00,77,00,00,00
"HardwareInformation.ChipType"=hex:51,00,56,00,47,00,
41,00,20,00,4e,00,65,00,\
  77,00,00,00
"HardwareInformation.DacType"=hex:55,00,6e,00,6b,00,6
e,00,6f,00,77,00,6e,00,00,\
  00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Rdr]
"Type"=dword:00000002
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5
c,53,79,73,74,65,6d,33,32,\
  5c,64,72,69,76,65,72,73,5c,72,64,72,2e,73,79,73,00
"DisplayName"="Rdr"
"Group"="Network"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Rdr\Linkage]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Rdr\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Rdr\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Rdr\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,25,02,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,c8,00,14,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,c8,00,14
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,c8,00,14,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Rdr\Enum]
"0"="Root\\LEGACY_RDR\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Replicator]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,53,79,73,74,65,6d,33,\
  32,5c,6c,6d,72,65,70,6c,2e,65,78,65,00
"DisplayName"="Directory Replicator"
"DependOnService"=hex(7):4c,61,6e,6d,61,6e,57,6f,72,6
b,73,74,61,74,69,6f,6e,00,\
  4c,61,6e,6d,61,6e,53,65,72,76,65,72,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Replicator\Exports]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Replicator\Imports]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Replicator\Linkage]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Replicator\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Replicator\Parameters]
"Replicate"=dword:00000002
"ImportPath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,
25,5c,53,79,73,74,65,6d,\
  33,32,5c,52,65,70,6c,5c,49,6d,70,6f,72,74,00
"ExportPath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,
25,5c,53,79,73,74,65,6d,\
  33,32,5c,52,65,70,6c,5c,45,78,70,6f,72,74,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Replicator\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,25,02,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,c8,00,14,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,c8,00,14
,00,00,00,1c,00,ff,01,\
0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,c8,00,14,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\RPCLOCATOR]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,53,79,73,74,65,6d,33,\
  32,5c,4c,4f,43,41,54,4f,52,2e,45,58,45,00
"DisplayName"="Remote Procedure Call (RPC) Locator"
"ObjectName"="LocalSystem"
"DependOnService"=hex(7):4c,61,6e,6d,61,6e,57,6f,72,6
b,73,74,61,74,69,6f,6e,00,\
  52,64,72,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\RPCLOCATOR\Linkage]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\RPCLOCATOR\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\RPCLOCATOR\Parameters]
"ExpirationAge"=dword:00000e10

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\RPCLOCATOR\Security]
"Security"=hex:01,00,14,80,cc,00,00,00,d8,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,98,00,06,00,00,00,00,03,18,00
,00,00,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,00,00,00,00,00,03,18,00,ff
,01,0f,00,01,02,00,00,\

00,00,00,05,20,00,00,00,20,02,00,00,00,03,18,00,ff,01
,0f,00,01,01,00,00,00,\

00,00,05,12,00,00,00,20,02,00,00,00,03,18,00,00,00,02
,00,01,02,00,00,00,00,\

00,05,20,00,00,00,23,02,00,00,00,03,18,00,9d,00,00,00
,01,01,00,00,00,00,00,\

05,04,00,00,00,23,02,00,00,00,03,18,00,9d,00,00,00,01
,02,00,00,00,00,00,05,\

20,00,00,00,21,02,00,00,01,01,00,00,00,00,00,05,12,00
,00,00,01,01,00,00,00,\
  00,00,05,12,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\RpcSs]
"Type"=dword:00000010
"DisplayName"="Remote Procedure Call (RPC) Service"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,73,79,73,74,65,6d,33,\
  32,5c,52,70,63,53,73,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\RpcSs\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\RpcSs\Security]
"Security"=hex:01,00,14,80,cc,00,00,00,d8,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,98,00,06,00,00,00,00,03,18,00
,00,00,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,00,00,00,00,00,03,18,00,ff
,01,0f,00,01,02,00,00,\

00,00,00,05,20,00,00,00,20,02,00,00,00,03,18,00,ff,01
,0f,00,01,01,00,00,00,\

00,00,05,12,00,00,00,20,02,00,00,00,03,18,00,00,00,02
,00,01,02,00,00,00,00,\

00,05,20,00,00,00,23,02,00,00,00,03,18,00,9d,00,00,00
,01,01,00,00,00,00,00,\

05,04,00,00,00,23,02,00,00,00,03,18,00,9d,00,00,00,01
,02,00,00,00,00,00,05,\

20,00,00,00,21,02,00,00,01,01,00,00,00,00,00,05,12,00
,00,00,01,01,00,00,00,\
  00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\RpcSs\Enum]
"0"="Root\\LEGACY_RPCSS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\s3]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\s3\Device0]
"VgaCompatible"=dword:00000000
"InstalledDisplayDrivers"=hex(7):73,33,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\SCache]
"ErrorControl"=dword:00000001
"Start"=dword:00000003
"Type"=dword:00000001
"Group"="Base"
"KeyCode"="B5J6775-199-159-150-135-1FCX"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\SCache\Parameters]
"BreakOnEntry"=dword:00000000
"DebugLevel"=dword:00000000
"Drive"=dword:00000000
"Partition"=dword:00000001
"PageSize"=dword:00008000
"MinLRU"=dword:00000020
"LazyWrite"=dword:00000000
"PendingLimit"=dword:00000010
"SystemDevice"=dword:00000000
"devChar"=dword:00000911
"NpParm"=dword:00036775
"DosDevice"="G:"
"PartitionList"=hex:00,10,00,00,13,00,00,00,d0,04,00,
00,00,00,00,00,00,00,00,\

00,17,00,00,00,02,00,00,00,00,00,00,00,00,00,f1,7f,00
,00,00,00,43,3a,00,00,\

00,00,00,00,46,41,54,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,17,00,00
,00,03,00,00,00,00,00,\

00,00,00,f8,81,7d,00,00,00,00,44,3a,00,00,00,00,00,00
,4e,54,46,53,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,16,00,00,00,01,00,00,00,00,00,00,00,00,c0
,ef,34,0c,00,00,00,46,\

3a,00,00,00,00,00,00,52,41,57,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,01,00,00,00,01,00,00,\

00,00,00,00,00,00,c0,0f,e4,04,00,00,00,47,3a,00,00,00
,00,00,00,52,41,57,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,04,00,00,00,01,00,00,00,00,00,00
,00,00,c0,8f,e4,04,00,\

00,00,48,3a,00,00,00,00,00,00,52,41,57,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,06,00,00,00,\
01,00,00,00,00,00,00,00,00,c0,4f,e5,04,00,00,00,49,3a
,00,00,00,00,00,00,52,\

41,57,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,08,00,00,00,01,00,00,00
,00,00,00,00,00,c0,0f,\

e6,04,00,00,00,4a,3a,00,00,00,00,00,00,52,41,57,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,10,\

00,00,00,01,00,00,00,00,00,00,00,00,c0,cf,e6,04,00,00
,00,4b,3a,00,00,00,00,\

00,00,52,41,57,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,12,00,00,00,01
,00,00,00,00,00,00,00,\

00,c0,4f,e7,04,00,00,00,4c,3a,00,00,00,00,00,00,52,41
,57,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,14,00,00,00,01,00,00,00,00,00,00,00,00,c0,0f,e8
,04,00,00,00,4d,3a,00,\

00,00,00,00,00,52,41,57,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,02,00
,00,00,01,00,00,00,00,\

00,00,00,00,c0,ff,ed,02,00,00,00,4e,3a,00,00,00,00,00
,00,52,41,57,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,05,00,00,00,01,00,00,00,00,00,00,00,00
,c0,ff,ed,02,00,00,00,\

4f,3a,00,00,00,00,00,00,52,41,57,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,07,00,00,00,01,00,\

00,00,00,00,00,00,00,c0,ff,ed,02,00,00,00,50,3a,00,00
,00,00,00,00,52,41,57,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,01,00,00,00,09,00,00,00,01,00,00,00,00,00
,00,00,00,c0,ff,ed,02,\
```

```
00,00,00,51,3a,00,00,00,00,00,00,52,41,57,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,11,00,00,\

00,01,00,00,00,00,00,00,00,00,c0,ff,ed,02,00,00,00,52
,3a,00,00,00,00,00,00,\

52,41,57,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,13,00,00,00,01,00,00
,00,00,00,00,00,00,c0,\

ff,ed,02,00,00,00,53,3a,00,00,00,00,00,00,52,41,57,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

15,00,00,00,01,00,00,00,00,00,00,00,00,c0,ff,ed,02,00
,00,00,54,3a,00,00,00,\

00,00,00,52,41,57,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,01,00,00,00,00,00,00,\

00,00,b0,ef,69,18,00,00,00,59,3a,00,00,00,00,00,00,4e
,54,46,53,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,03,00,00,00,01,00,00,00,00,00,00,00,00,b0,ef
,69,18,00,00,00,5a,3a,\

00,00,00,00,00,00,4e,54,46,53,00,00,00,00,00,00,00,00
,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
\SCache\Enum]
"0"="Root\\LEGACY_SCACHE\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Schedule]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,53,79,73,74,65,6d,33,\
  32,5c,41,74,53,76,63,2e,45,78,65,00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ScMonitor]
"Type"=dword:00000010
```

```
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,53,79,73,74,65,6d,33,\
  32,5c,53,63,4d,6f,6e,69,74,6f,72,2e,65,78,65,00
"DisplayName"="SuperCache-NT Monitor"
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ScMonitor\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,74,00,69,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,76,00,65,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,76,00,65
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,76,00,65,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\ScMonitor\Enum]
"0"="Root\\LEGACY_SCMONITOR\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Scsiprnt]
"Type"=dword:00000001
"Start"=dword:00000002
"Group"="Extended base"
"ErrorControl"=dword:00000000
"DependOnGroup"=hex(7):53,43,53,49,20,6d,69,6e,69,70,
6f,72,74,00,00
"Tag"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Scsiprnt\Enum]
"Count"=dword:00000000
"NextInstance"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Scsiscan]
"Type"=dword:00000001
"Start"=dword:00000003
"Group"="SCSI Class"
```

```
"ErrorControl"=dword:00000000
"Tag"=dword:00000003
"DependOnGroup"=hex(7):53,43,53,49,20,6d,69,6e,69,70,
6f,72,74,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Scsiscan\Enum]
"0"="Root\\LEGACY_SCSISCAN\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Serial]
"Type"=dword:00000001
"Start"=dword:00000002
"Group"="Extended base"
"ErrorControl"=dword:00000000
"Tag"=dword:00000001
"ForceFifoEnable"=dword:00000001
"RxFIFO"=dword:00000008
"TxFIFO"=dword:00000001
"PermitShare"=dword:00000000
"LogFifo"=dword:00000000
"ShareInterrupt"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Serial\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Serial\Enum]
"0"="Root\\LEGACY_SERIAL\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Sermouse]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Pointer Port"
"ErrorControl"=dword:00000000
"Tag"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Sermouse\Parameters]
"MouseDataQueueSize"=dword:00000064
"NumberOfButtons"=dword:00000002
"SampleRate"=dword:00000028
"PointerDeviceBaseName"="PointerPort"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Sfloppy]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="Primary disk"
"ErrorControl"=dword:00000000
"Tag"=dword:00000004
"DependOnGroup"=hex(7):53,43,53,49,20,6d,69,6e,69,70,
6f,72,74,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Sfloppy\Enum]
"Count"=dword:00000000
"NextInstance"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Simbad]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Filter"
"ErrorControl"=dword:00000001
"Tag"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\slcd32]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:0000001c

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\SMARTperf]
"Type"=dword:00000001
"Start"=dword:00000004
"ErrorControl"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\SMARTperf\Performance]
"Close"="CloseSMARTPerformanceData"
"Collect"="CollectSMARTPerformanceData"
"Library"="SMARTctr.dll"
"Open"="OpenSMARTPerformanceData"
"Last Counter"=dword:000008ec
"Last Help"=dword:000008ed
"First Counter"=dword:0000083c
"First Help"=dword:0000083d

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Sparrow]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Spock]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Spooler]
"Type"=dword:00000110
"Start"=dword:00000003
"Group"="SpoolerGroup"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,73,79,73,74,65,6d,33,\
  32,5c,73,70,6f,6f,6c,73,73,2e,65,78,65,00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Spooler\Parameters]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Spooler\Enum]
"0"="Root\\LEGACY_SPOOLER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\SQLServerAgent]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,4d,53,53,51,4c,37,5c,62,6
9,6e,6e,5c,73,71,6c,61,67,\
  65,6e,74,2e,65,78,65,00
"DisplayName"="SQLServerAgent"
"DependOnService"=hex(7):4d,53,53,51,4c,53,65,72,76,6
5,72,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\SQLServerAgent\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,74,00,69,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,6f,00,6e,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,6f,00,6e
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,6f,00,6e,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Srv]
"Type"=dword:00000002
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5
c,53,79,73,74,65,6d,33,32,\
  5c,64,72,69,76,65,72,73,5c,73,72,76,2e,73,79,73,00
"DisplayName"="Srv"
"Group"="Network"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Srv\Linkage]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Srv\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Srv\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Srv\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,25,02,00,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,c8,00,14,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,c8,00,14
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,c8,00,14,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Srv\Enum]
"0"="Root\\LEGACY_SRV\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\symc810]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:0000001a
"RequestedSystemResources"=hex(a):48,01,00,00,05,00,0
0,00,04,00,00,00,29,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,01,00,00,00,01
,00,01,00,09,00,00,00,\

00,02,03,00,00,00,00,00,25,00,00,00,25,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,01,01,01,00,01,00,00,00,00,01,00
,00,00,01,00,00,00,64,\

00,00,00,00,00,00,ff,64,00,00,00,00,00,00,08,01,01,00
,01,00,00,00,00,01,00,\

00,00,01,00,00,00,64,00,00,00,00,00,00,ff,64,00,00,00
,00,00,00,01,03,01,00,\

00,00,00,00,00,01,00,00,00,01,00,00,00,00,8d,f7,00,00
,00,00,ff,00,8d,f7,00,\

00,00,00,08,03,01,00,00,00,00,00,00,01,00,00,00,01,00
,00,00,00,8d,f7,00,00,\

00,00,ff,00,8d,f7,00,00,00,00,08,03,01,00,00,00,00,00
,00,01,00,00,00,01,00,\

00,00,00,8c,f7,00,00,00,00,ff,0f,8c,f7,00,00,00,00,01
,03,01,00,00,00,00,00,\

00,10,00,00,00,10,00,00,00,00,8c,f7,00,00,00,00,ff,0f
,8c,f7,00,00,00,00,08,\

03,01,00,00,00,00,00,00,10,00,00,00,10,00,00,00,00,8d
,f7,00,00,00,00,ff,00,\

8d,f7,00,00,00,00,08,03,01,00,00,00,00,00,00,10,00,00
,00,10,00,00,00,00,8c,\
  f7,00,00,00,00,ff,0f,8c,f7,00,00,00,00
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,4
9,56,45,52,53,5c,73,79,6d,\
  63,38,31,30,2e,73,79,73,00
"PlugPlayServiceType"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\SysMgmt]
"Type"=dword:00000001
"Start"=dword:00000004
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,59,53,54,45,4d,33,32,5c,44,52,4
9,56,45,52,53,5c,53,59,53,\
  4d,47,4d,54,2e,53,59,53,00
"DisplayName"="Compaq System Management Driver"
"Group"="System Management"
"InstalledBySetup"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\SysMgmt\Linkage]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\SysMgmt\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\SysMgmt\Parameters]
"DriverMode"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\SysMgmt\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,48,63,16,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,00,ff,14,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,00,ff,14
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,00,ff,14,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\SysMgmt\Enum]
"0"="Root\\LEGACY_SYSMGMT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\T128]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000012

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\T128\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\T128\Parameters\Device0]
"DriverParameter"="IRQ=5"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\T13B]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000013

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\T13B\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\T13B\Parameters\Device0]
"DriverParameter"="IRQ=5"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\TapiSrv]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,73,79,73,74,65,6d,33,\
  32,5c,74,61,70,69,73,72,76,2e,65,78,65,00
```

```
"ObjectName"="LocalSystem"
"DisplayName"="Telephony Service"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\TapiSrv\Performance]
"Close"="CloseTapiPerformanceData"
"Collect"="CollectTapiPerformanceData"
"Library"="tapiperf.dll"
"Open"="OpenTapiPerformanceData"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\TapiSrv\Security]
"Security"=hex:01,00,14,80,b4,00,00,00,c0,00,00,00,14
,00,00,00,34,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,80,00,05,00,00,00,00,03,18,00
,8d,00,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,00,00,00,00,00,03,18,00,ff
,01,0f,00,01,02,00,00,\

00,00,00,05,20,00,00,00,20,02,00,00,00,03,18,00,8f,00
,02,00,01,02,00,00,00,\

00,00,05,20,00,00,00,23,02,00,00,00,03,18,00,9d,00,00
,00,01,01,00,00,00,00,\

00,05,04,00,00,00,23,02,00,00,00,03,18,00,9d,00,00,00
,01,02,00,00,00,00,00,\

05,20,00,00,00,21,02,00,00,01,01,00,00,00,00,00,05,12
,00,00,00,01,01,00,00,\
  00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Tardis]
"Type"=dword:00000010
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,57,49,4e,4e,54,34,30,5c,5
3,79,73,74,65,6d,33,32,5c,\
  54,61,72,64,69,73,4e,54,2e,65,78,65,00
"DisplayName"="Tardis time service"
"DependOnService"=hex(7):54,63,70,69,70,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Tardis\Parameters]
"Protocol"=dword:000003e9
"Server"="archive"
"Max"="0"
"Frequency"="720"
"Within"="100"
"SNTPServer"=dword:00000001
"TCPServer"=dword:00000001
"UDPServer"=dword:00000001
"NTPBroadcastServer"=dword:00000000
"Dialup"=dword:00000000
"LogDuplicates"=dword:00000001
"LogWarn"=dword:00000001
"LogError"=dword:00000001
"Logtime"=dword:00000000
"LogServerChange"=dword:00000000
"LogPollChange"=dword:00000000
"AutoAdjust"=dword:00000000
"AutoChange"=dword:00000001
"Servers"="archive,ashe.cs.tcd.ie,bernina.ethz.ch,bla
ck-
ice.cc.vt.edu,clepsydra.dec.com,clock.psu.edu,clock.t
ricity.wsu.edu,constellation.ecn.ou.edu,delphi.cs.ucl
a.edu,eagle.tamu.edu,esavax.esa.lanl.gov,fartein.ifi.
uio.no,fuzz.psc.edu,gazette.bcm.tmc.edu,gilbreth.ecn.
purdue.edu,harbor.ecn.purdue.edu,heechee.esa.lanl.gov
,jane.jpl.nasa.gov,kuhub.cc.ukans.edu,lane.cc.ukans.e
du,lib1.tcd.ie,libra.rice.edu,louie.udel.edu,molecule
.ecn.purdue.edu,nic.near.net,noc.near.net,ntp.adelaid
e.edu.au,ntp.cox.smu.edu,ntp.css.gov,ntp.olivetti.com
,ntp.univ-lyon1.fr,ntp0.cornell.edu,ntp-
0.cso.uiuc.edu,ntp0.strath.ac.uk,ntp-
1.cso.uiuc.edu,ntp1.strath.ac.uk,ntp-
2.cso.uiuc.edu,ntp2.strath.ac.uk,ntp3.strath.ac.uk,nt
p4.strath.ac.uk,salmon.maths.tcd.ie,sundial.columbia.
edu,timeserver.cs.umb.edu,tmc.edu,vtserf.cc.vt.edu,wu
archive.wustl.edu,yoyo.aarnet.edu.au,zeus.tamu.edu,"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Tardis\Security]
"Security"=hex:01,00,14,80,34,00,00,00,40,00,00,00,14
,00,00,00,00,00,00,00,02,\

00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,01,01,00,00,00,00,00,05,12,00,00,00
,01,01,00,00,00,00,00,\
  05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Tardis\Enum]
"0"="Root\\LEGACY_TARDIS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Tcpip]
"Type"=dword:00000001
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5
c,53,79,73,74,65,6d,33,32,\

5c,64,72,69,76,65,72,73,5c,74,63,70,69,70,2e,73,79,73
,00
"DisplayName"="TCP/IP Service"
"Group"="PNP_TDI"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Tcpip\Linkage]
"Bind"=hex(7):5c,44,65,76,69,63,65,5c,4e,31,30,30,31,
00,00
"Export"=hex(7):5c,44,65,76,69,63,65,5c,54,63,70,69,7
0,5c,4e,31,30,30,31,00,00
"Route"=hex(7):22,4e,31,30,30,22,20,22,4e,31,30,30,31
,22,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Tcpip\Linkage\Disabled]
"Bind"=hex(7):00
"Export"=hex(7):00
"Route"=hex(7):00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Tcpip\Parameters]
"DataBasePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,7
4,25,5c,53,79,73,74,65,6d,\
  33,32,5c,64,72,69,76,65,72,73,5c,65,74,63,00
"Domain"=""
"Hostname"="tigger2"
"NameServer"=""
"ForwardBroadcasts"=dword:00000000
"IPEnableRouter"=dword:00000000
"SearchList"=""
"EnableSecurityFilters"=dword:00000000
"KeepAliveTime"=dword:000493e0

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Tcpip\Parameters\PersistentRoutes]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Tcpip\Parameters\Winsock]
"HelperDllName"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,
74,25,5c,53,79,73,74,65,\
  6d,33,32,5c,77,73,68,74,63,70,69,70,2e,64,6c,6c,00
"MaxSockAddrLength"=dword:00000010
"MinSockAddrLength"=dword:00000010
"Mapping"=hex:0b,00,00,00,03,00,00,00,02,00,00,00,01,
00,00,00,06,00,00,00,02,\

00,00,00,01,00,00,00,00,00,00,00,02,00,00,00,00,00,00
,00,06,00,00,00,00,00,\

00,00,00,00,00,00,06,00,00,00,00,00,00,00,01,00,00,00
,06,00,00,00,02,00,00,\

00,02,00,00,00,11,00,00,00,02,00,00,00,02,00,00,00,00
,00,00,00,02,00,00,00,\

00,00,00,00,11,00,00,00,00,00,00,00,00,00,00,00,11,00
,00,00,00,00,00,00,02,\

00,00,00,11,00,00,00,02,00,00,00,03,00,00,00,00,00,00
,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Tcpip\Performance]
"Library"="Perfctrs.dll"
"Open"="OpenTcpIpPerformanceData"
"Collect"="CollectTcpIpPerformanceData"
"Close"="CloseTcpIpPerformanceData"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Tcpip\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14
,00,00,00,34,00,00,00,02,\
00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00
,00,00,00,00,01,00,00,\

00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00
,8d,01,02,00,01,01,00,\

00,00,00,00,01,00,00,00,00,72,00,76,00,00,00,1c,00,fd
,01,02,00,01,02,00,00,\

00,00,00,05,20,00,00,00,23,02,00,00,69,00,63,00,00,00
,1c,00,ff,01,0f,00,01,\

02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,69,00,63
,00,00,00,1c,00,ff,01,\

0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00
,69,00,63,00,00,00,18,\

00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25
,02,00,00,01,01,00,00,\

00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00
,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Tcpip\ServiceProvider]
"Class"=dword:00000008
"DnsPriority"=dword:000007d0
"HostsPriority"=dword:000001f4
"LocalPriority"=dword:000001f3
"ProviderPath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,7
4,25,5c,53,79,73,74,65,6d,\
  33,32,5c,77,73,6f,63,6b,33,32,2e,64,6c,6c,00
"NetbtPriority"=dword:000007d1
"Name"="TCP/IP"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Tcpip\Enum]
"0"="Root\\LEGACY_TCPIP\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\tga]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\tga\Device0]
"VgaCompatible"=dword:00000000
"InstalledDisplayDrivers"=hex(7):74,67,61,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\tmv1]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000018
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\tmv1\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\tmv1\Parameters\Device0]
"DriverParameter"="IRQ=15"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ultra124]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:0000000b

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ultra14f]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:0000000a

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Ultra24f]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:0000000c

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\update]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="Base"
"ErrorControl"=dword:00000000
"Tag"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\update\Enum]
"Count"=dword:00000000
"NextInstance"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\UPS]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,2
5,5c,53,79,73,74,65,6d,33,\
  32,5c,75,70,73,2e,65,78,65,00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\v7vram]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\v7vram\Device0]
```

```
"VgaCompatible"=dword:00000001
"InstalledDisplayDrivers"=hex(7):76,67,61,00,76,67,61
,32,35,36,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\VgaSave]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="Video Save"
"ErrorControl"=dword:00000000
"Tag"=dword:00000001
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5
c,53,79,73,74,65,6d,33,32,\
  5c,64,72,69,76,65,72,73,5c,76,67,61,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\VgaSave\Device0]
"VgaCompatible"=dword:00000001
"InstalledDisplayDrivers"=hex(7):76,67,61,00,00
"HardwareInformation.Crc32"=hex:57,d7,f0,ab

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\VgaSave\Enum]
"0"="Root\\LEGACY_VGASAVE\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\VgaStart]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="Video Init"
"ErrorControl"=dword:00000000
"Tag"=dword:00000001
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5
c,53,79,73,74,65,6d,33,32,\
  5c,64,72,69,76,65,72,73,5c,76,67,61,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\VgaStart\Device0]
"VgaCompatible"=dword:00000001
"InstalledDisplayDrivers"=hex(7):76,67,61,00,00
"ForceVga"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\VgaStart\Enum]
"Count"=dword:00000000
"NextInstance"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\VXD]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\VXD\JAVASUP]
"Start"=hex:00
"StaticVxD"="JAVASUP.VXD"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\w3svc]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\w3svc\parameters]
```

```
[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\Virtual Roots]
"/MSADC"="C:\\Program Files\\Common
Files\\System\\MSADC,,5"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Wd33c93]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="SCSI miniport"
"ErrorControl"=dword:00000001
"Tag"=dword:00000011

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Wd33c93\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Wd33c93\Parameters\Device0]
"DriverParameter"="IRQ=10;DMA=6"
"DisableDisconnect"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\wd90c24a]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\wd90c24a\Device0]
"VgaCompatible"=dword:00000001
"InstalledDisplayDrivers"=hex(7):76,67,61,00,76,67,61
,32,35,36,00,77,64,39,30,\
  63,32,34,61,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\wdvga]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\wdvga\Device0]
"VgaCompatible"=dword:00000001
"InstalledDisplayDrivers"=hex(7):76,67,61,00,76,67,61
,32,35,36,00,76,67,61,36,\
  34,6b,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\weitekp9]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\weitekp9\Device0]
"VgaCompatible"=dword:00000000
"InstalledDisplayDrivers"=hex(7):77,65,69,74,65,6b,70
,39,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock]
"Type"=dword:00000004
"Start"=dword:00000003
"ErrorControl"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock\Autodial]
"AutodialDllName32"="wininet.dll"
"AutodialFcnName32"="InternetAutodialCallback"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock\Linkage]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock\Parameters]
"Transports"=hex(7):54,63,70,69,70,00,4e,65,74,42,49,
4f,53,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock\Setup Migration]
"Setup Version"=dword:00001009
"Provider
List"=hex(7):54,63,70,69,70,00,4e,65,74,42,49,4f,53,0
0,00
"Known Static
Providers"=hex(7):54,63,70,69,70,00,4e,77,6c,6e,6b,49
,70,78,00,\

4e,77,6c,6e,6b,53,70,78,00,41,70,70,6c,65,54,61,6c,6b
,00,49,73,6f,54,70,00,\
  00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock\Setup Migration\Providers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock\Setup Migration\Providers\NetBIOS]
"WinSock 1.1 Provider
Data"=hex:0e,10,00,00,11,00,00,00,14,00,00,00,14,00,0
0,\

00,05,00,00,00,00,00,00,80,00,fa,00,00,68,00,00,00,09
,12,00,00,11,00,00,00,\

14,00,00,00,14,00,00,00,02,00,00,00,00,00,00,80,00,fa
,00,00,40,00,00,00,5c,\

00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,4e,00,65
,00,74,00,42,00,54,00,\

5f,00,4e,00,31,00,30,00,30,00,31,00,00,00,5c,00,44,00
,65,00,76,00,69,00,63,\

00,65,00,5c,00,4e,00,65,00,74,00,42,00,54,00,5f,00,4e
,00,31,00,30,00,30,00,\
  31,00,00,00
"WinSock 2.0 Provider
ID"=hex:30,18,5f,8d,73,c2,cf,11,95,c8,00,80,5f,48,a1,
92

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock\Setup Migration\Providers\Tcpip]
"WinSock 2.0 Provider
ID"=hex:a0,1a,0f,e7,8b,ab,cf,11,8c,a3,00,80,5f,48,a1,
92

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock\Setup Migration\Well Known Guids]
"IsoTp"=hex:b0,cb,e4,89,c1,b9,cf,11,95,c8,00,80,5f,48
,a1,92
"McsXns"=hex:b1,cb,e4,89,c1,b9,cf,11,95,c8,00,80,5f,4
8,a1,92
"AppleTalk"=hex:a0,17,3b,2c,df,c6,cf,11,95,c8,00,80,5
f,48,a1,92

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock2]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock2\Parameters]
"WinSock_Registry_Version"="2.0"
"Current_NameSpace_Catalog"="NameSpace_Catalog5"
"Current_Protocol_Catalog"="Protocol_Catalog9"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock2\Parameters\NameSpace_Catalog5]
"Num_Catalog_Entries"=dword:00000001
"Next_Provider_ID"=dword:000007d0

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock2\Parameters\NameSpace_Catalog5\Catalog_Entri
es]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock2\Parameters\NameSpace_Catalog5\Catalog_Entri
es\000000000001]
"LibraryPath"="%SystemRoot%\\System32\\rnr20.dll"
"DisplayString"="TCP/IP"
"ProviderId"=hex:40,9d,05,22,9e,7e,cf,11,ae,5a,00,aa,
00,a7,11,2b
"SupportedNameSpace"=dword:0000000c
"Enabled"=dword:00000001
"Version"=dword:00000000
"StoresServiceClassInfo"=dword:000016c8

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock2\Parameters\Protocol_Catalog9]
"Num_Catalog_Entries"=dword:00000005
"Next_Provider_ID"=dword:00000001
"Next_Catalog_Entry_ID"=dword:0000040c
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock2\Parameters\Protocol_Catalog9\Catalog_Entrie
s]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinSock2\Parameters\Protocol_Catalog9\Catalog_Entrie
s\000000000001]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f
,74,25,5c,73,79,73,74,65,\

6d,33,32,5c,6d,73,61,66,64,2e,64,6c,6c,00,28,69,66,70
,72,65,73,65,6e,74,20,\

5f,20,44,65,76,20,5f,20,5f,20,4f,62,6a,65,63,74,6e,61
,6d,65,29,0d,0a,20,20,\

20,20,28,63,75,74,29,0d,0a,20,20,20,20,28,73,74,72,69
,6e,67,5f,63,6f,6e,63,\

61,74,20,4f,6c,64,73,74,72,20,4f,62,6a,65,63,74,6e,61
,6d,65,20,4e,65,77,73,\

74,72,29,0d,0a,29,0d,0a,0d,0a,20,20,0d,0a,20,20,0d,0a
,0d,0a,28,20,28,6d,61,\

6b,65,64,65,76,73,74,72,69,6e,67,20,65,78,70,6f,72,74
,20,28,44,65,76,20,44,\

65,76,6e,65,78,74,7c,54,61,69,6c,29,20,4f,6c,64,73,74
,72,20,4e,65,77,73,74,\

72,29,0d,0a,20,20,20,20,28,69,73,65,6e,64,70,6f,69,6e
,74,20,44,65,76,29,0d,\

0a,20,20,20,20,28,69,73,73,74,72,65,61,6d,20,44,65,76
,6e,65,78,74,29,0d,0a,\

20,20,20,20,28,63,75,74,29,0d,0a,20,20,20,20,28,6d,66
,00,02,00,00,00,00,00,\

00,00,00,00,00,00,00,00,08,00,00,00,a0,1a,0f,e7,8b,ab
,cf,11,8c,a3,00,80,5f,\

48,a1,92,e9,03,00,00,01,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,02,00,00,00
,02,00,00,00,10,00,00,\

00,10,00,00,00,01,00,00,00,06,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,4d,00,53,00,41,00,46,00,44,00
,20,00,54,00,63,00,70,\

00,69,00,70,00,20,00,5b,00,54,00,43,00,50,00,2f,00,49
,00,50,00,5d,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

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
\WinSock2\Parameters\Protocol_Catalog9\Catalog_Entrie
s\000000000002]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f
,74,25,5c,73,79,73,74,65,\

6d,33,32,5c,6d,73,61,66,64,2e,64,6c,6c,00,28,69,66,70
,72,65,73,65,6e,74,20,\

5f,20,44,65,76,20,5f,20,5f,20,4f,62,6a,65,63,74,6e,61
,6d,65,29,0d,0a,20,20,\

20,20,28,63,75,74,29,0d,0a,20,20,20,20,28,73,74,72,69
,6e,67,5f,63,6f,6e,63,\

61,74,20,4f,6c,64,73,74,72,20,4f,62,6a,65,63,74,6e,61
,6d,65,20,4e,65,77,73,\

74,72,29,0d,0a,29,0d,0a,0d,0a,20,20,0d,0a,20,20,0d,0a
,0d,0a,28,20,28,6d,61,\

6b,65,64,65,76,73,74,72,69,6e,67,20,65,78,70,6f,72,74
,20,28,44,65,76,20,44,\

65,76,6e,65,78,74,7c,54,61,69,6c,29,20,4f,6c,64,73,74
,72,20,4e,65,77,73,74,\

72,29,0d,0a,20,20,20,20,28,69,73,65,6e,64,70,6f,69,6e
,74,20,44,65,76,29,0d,\

0a,20,20,20,20,28,69,73,73,74,72,65,61,6d,20,44,65,76
,6e,65,78,74,29,0d,0a,\

20,20,20,20,28,63,75,74,29,0d,0a,20,20,20,20,28,6d,09
,06,02,00,00,00,00,00,\

00,00,00,00,00,00,00,00,08,00,00,00,a0,1a,0f,e7,8b,ab
,cf,11,8c,a3,00,80,5f,\

48,a1,92,ea,03,00,00,01,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,02,00,00,00
,02,00,00,00,10,00,00,\

00,10,00,00,00,02,00,00,00,11,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

bb,ff,00,00,00,00,00,00,4d,00,53,00,41,00,46,00,44,00
,20,00,54,00,63,00,70,\

00,69,00,70,00,20,00,5b,00,55,00,44,00,50,00,2f,00,49
,00,50,00,5d,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

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
\WinSock2\Parameters\Protocol_Catalog9\Catalog_Entrie
s\000000000003]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f
,74,25,5c,73,79,73,74,65,\

6d,33,32,5c,6d,73,61,66,64,2e,64,6c,6c,00,28,69,66,70
,72,65,73,65,6e,74,20,\

5f,20,44,65,76,20,5f,20,5f,20,4f,62,6a,65,63,74,6e,61
,6d,65,29,0d,0a,20,20,\

20,20,28,63,75,74,29,0d,0a,20,20,20,20,28,73,74,72,69
,6e,67,5f,63,6f,6e,63,\

61,74,20,4f,6c,64,73,74,72,20,4f,62,6a,65,63,74,6e,61
,6d,65,20,4e,65,77,73,\

74,72,29,0d,0a,29,0d,0a,0d,0a,20,20,0d,0a,20,20,0d,0a
,0d,0a,28,20,28,6d,61,\

6b,65,64,65,76,73,74,72,69,6e,67,20,65,78,70,6f,72,74
,20,28,44,65,76,20,44,\

65,76,6e,65,78,74,7c,54,61,69,6c,29,20,4f,6c,64,73,74
,72,20,4e,65,77,73,74,\

72,29,0d,0a,20,20,20,20,28,69,73,65,6e,64,70,6f,69,6e
,74,20,44,65,76,29,0d,\

0a,20,20,20,20,28,69,73,73,74,72,65,61,6d,20,44,65,76
,6e,65,78,74,29,0d,0a,\

20,20,20,20,28,63,75,74,29,0d,0a,20,20,20,20,28,6d,09
,06,02,00,00,00,00,00,\

00,00,00,00,00,00,00,00,0c,00,00,00,a0,1a,0f,e7,8b,ab
,cf,11,8c,a3,00,80,5f,\

48,a1,92,eb,03,00,00,01,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,02,00,00,00
,02,00,00,00,10,00,00,\

00,10,00,00,00,03,00,00,00,00,00,00,00,ff,00,00,00,00
,00,00,00,00,00,00,00,\

bb,ff,00,00,00,00,00,00,4d,00,53,00,41,00,46,00,44,00
,20,00,54,00,63,00,70,\

00,69,00,70,00,20,00,5b,00,52,00,41,00,57,00,2f,00,49
,00,50,00,5d,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

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
\WinSock2\Parameters\Protocol_Catalog9\Catalog_Entrie
s\000000000004]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f
,74,25,5c,73,79,73,74,65,\

6d,33,32,5c,6d,73,61,66,64,2e,64,6c,6c,00,28,69,66,70
,72,65,73,65,6e,74,20,\

5f,20,44,65,76,20,5f,20,5f,20,4f,62,6a,65,63,74,6e,61
,6d,65,29,0d,0a,20,20,\

20,20,28,63,75,74,29,0d,0a,20,20,20,20,28,73,74,72,69
,6e,67,5f,63,6f,6e,63,\

61,74,20,4f,6c,64,73,74,72,20,4f,62,6a,65,63,74,6e,61
,6d,65,20,4e,65,77,73,\

74,72,29,0d,0a,29,0d,0a,0d,0a,20,20,0d,0a,20,20,0d,0a
,0d,0a,28,20,28,6d,61,\

6b,65,64,65,76,73,74,72,69,6e,67,20,65,78,70,6f,72,74
,20,28,44,65,76,20,44,\

65,76,6e,65,78,74,7c,54,61,69,6c,29,20,4f,6c,64,73,74
,72,20,4e,65,77,73,74,\

72,29,0d,0a,20,20,20,20,28,69,73,65,6e,64,70,6f,69,6e
,74,20,44,65,76,29,0d,\

0a,20,20,20,20,28,69,73,73,74,72,65,61,6d,20,44,65,76
,6e,65,78,74,29,0d,0a,\

20,20,20,20,28,63,75,74,29,0d,0a,20,20,20,20,28,6d,0e
,00,02,00,00,00,00,00,\

00,00,00,00,00,00,00,00,08,00,00,00,30,18,5f,8d,73,c2
,cf,11,95,c8,00,80,5f,\

48,a1,92,0a,04,00,00,01,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\
00,00,00,00,00,00,00,00,00,00,00,00,00,00,02,00,00,00
,11,00,00,00,14,00,00,\

00,14,00,00,00,05,00,00,00,00,00,00,80,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,fa,00,00,00,00,00,00,4d,00,53,00,41,00,46,00,44,00
,20,00,4e,00,65,00,74,\

00,42,00,49,00,4f,00,53,00,20,00,5b,00,5c,00,44,00,65
,00,76,00,69,00,63,00,\

65,00,5c,00,4e,00,65,00,74,00,42,00,54,00,5f,00,4e,00
,31,00,30,00,30,00,31,\

00,5d,00,20,00,53,00,45,00,51,00,50,00,41,00,43,00,4b
,00,45,00,54,00,20,00,\

30,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

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
\WinSock2\Parameters\Protocol_Catalog9\Catalog_Entrie
s\000000000005]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f
,74,25,5c,73,79,73,74,65,\

6d,33,32,5c,6d,73,61,66,64,2e,64,6c,6c,00,28,69,66,70
,72,65,73,65,6e,74,20,\

5f,20,44,65,76,20,5f,20,5f,20,4f,62,6a,65,63,74,6e,61
,6d,65,29,0d,0a,20,20,\

20,20,28,63,75,74,29,0d,0a,20,20,20,20,28,73,74,72,69
,6e,67,5f,63,6f,6e,63,\

61,74,20,4f,6c,64,73,74,72,20,4f,62,6a,65,63,74,6e,61
,6d,65,20,4e,65,77,73,\

74,72,29,0d,0a,29,0d,0a,0d,0a,20,20,0d,0a,20,20,0d,0a
,0d,0a,28,20,28,6d,61,\

6b,65,64,65,76,73,74,72,69,6e,67,20,65,78,70,6f,72,74
,20,28,44,65,76,20,44,\

65,76,6e,65,78,74,7c,54,61,69,6c,29,20,4f,6c,64,73,74
,72,20,4e,65,77,73,74,\

72,29,0d,0a,20,20,20,20,28,69,73,65,6e,64,70,6f,69,6e
,74,20,44,65,76,29,0d,\

0a,20,20,20,20,28,69,73,73,74,72,65,61,6d,20,44,65,76
,6e,65,78,74,29,0d,0a,\

20,20,20,20,28,63,75,74,29,0d,0a,20,20,20,20,28,6d,09
,02,02,00,00,00,00,00,\

00,00,00,00,00,00,00,00,08,00,00,00,30,18,5f,8d,73,c2
,cf,11,95,c8,00,80,5f,\

48,a1,92,0b,04,00,00,01,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,02,00,00,00
,11,00,00,00,14,00,00,\

00,14,00,00,00,02,00,00,00,00,00,00,80,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,fa,00,00,00,00,00,00,4d,00,53,00,41,00,46,00,44,00
,20,00,4e,00,65,00,74,\

00,42,00,49,00,4f,00,53,00,20,00,5b,00,5c,00,44,00,65
,00,76,00,69,00,63,00,\

65,00,5c,00,4e,00,65,00,74,00,42,00,54,00,5f,00,4e,00
,31,00,30,00,30,00,31,\

00,5d,00,20,00,44,00,41,00,54,00,41,00,47,00,52,00,41
,00,4d,00,20,00,30,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
,00,00,00,00,00,00,00,\

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
\WinTrust]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinTrust\SubjectPackages]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinTrust\SubjectPackages\MS Subjects 1]
```

```
"$DLL"="C:\\WINNT40\\System32\\MsSip1.dll"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinTrust\SubjectPackages\MS Subjects 2]
"$DLL"="C:\\WINNT40\\System32\\MsSip2.dll"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinTrust\SubjectPackages\MS Subjects 3]
"$DLL"="C:\\WINNT40\\System32\\MsSip3.dll"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinTrust\TrustProviders]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\WinTrust\TrustProviders\Software Publisher]
"$DLL"="C:\\WINNT40\\System32\\SoftPub.dll"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Xga]
"Type"=dword:00000001
"Start"=dword:00000004
"Group"="Video"
"ErrorControl"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
\Xga\Device0]
"VgaCompatible"=dword:00000000
"Monitor"="drivers\\vga.bvp"
"InstalledDisplayDrivers"=hex(7):78,67,61,00,76,67,61
,32,35,36,00,00
```

TPCC Application Registry Parameters

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPCC]
"PATH"="C:\\Inetpub\\wwwroot\\"
"LOG"="OFF"
"NumberOfDeliveryThreads"="7"
"MaximumWarehouses"="3000"
"BackoffDelay"="500"
"DeadlockRetry"="3"
"MaxConnections"="8000"
"LastInstalledVersion"="MTS"
"ConnectionPooling"="OFF"
"ConnectionPoolRetryTime"="500"
"QueueSLots"="330"
```

TUXEDO UBB File

```
*RESOURCES
IPCKEY 133133

MAXACCESSERS 2700
MAXSERVERS 180
MAXSERVICES 1800
MODEL SHM
MASTER C1
LDBAL Y
SCANUNIT 60
BLOCKTIME 60
BBLQUERY 60

*MACHINES
DEFAULT:

"C1" LMID=C1
        TUXDIR="C:\tuxedo"
        APPDIR="C:\InetPub\wwwroot"
        TUXCONFIG="C:\InetPub\wwwroot\tuxconfig"
        ULOGPFX="C:\InetPub\wwwroot\ULOG"
        TYPE="WinNT"
        UID= 0
        GID= 0

*GROUPS
Q1
        LMID=C1 GRPNO=1 OPENINFO=NONE

Q2
        LMID=C1 GRPNO=2 OPENINFO=NONE

Q3
        LMID=C1 GRPNO=3 OPENINFO=NONE

Q4
        LMID=C1 GRPNO=4 OPENINFO=NONE

Q5
        LMID=C1 GRPNO=5 OPENINFO=NONE

Q6
        LMID=C1 GRPNO=6 OPENINFO=NONE

Q7
        LMID=C1 GRPNO=7 OPENINFO=NONE

Q8
        LMID=C1 GRPNO=8 OPENINFO=NONE

Q9
        LMID=C1 GRPNO=9 OPENINFO=NONE

GROUPDEL
        LMID=C1 GRPNO=10 OPENINFO=NONE

*SERVERS
DEFAULT:

Q         SRVGRP=Q1
        SRVID=60
        MIN=4 MAX=5
        CLOPT="-A -- -Stigger2_ip -DTPCC"
        RQADDR=q1q REPLYQ=Y

Q         SRVGRP=Q2
        SRVID=200
        MIN=4 MAX=5
        CLOPT="-A -- -Stigger2_ip -DTPCC"
        RQADDR=q2q REPLYQ=Y

Q         SRVGRP=Q3
        SRVID=300
        MIN=4 MAX=5
        CLOPT="-A -- -Stigger2_ip -DTPCC"
        RQADDR=q3q REPLYQ=Y

Q         SRVGRP=Q4
        SRVID=400
        MIN=4 MAX=5
        CLOPT="-A -- -Stigger2_ip -DTPCC"
        RQADDR=q4q REPLYQ=Y

Q         SRVGRP=Q5
        SRVID=500
        MIN=4 MAX=5
        CLOPT="-A -- -Stigger2_ip -DTPCC"
        RQADDR=q5q REPLYQ=Y

Q         SRVGRP=Q6
        SRVID=600
        MIN=4 MAX=5
        CLOPT="-A -- -Stigger2_ip -DTPCC"
        RQADDR=q6q REPLYQ=Y

Q         SRVGRP=Q7
        SRVID=700
        MIN=4 MAX=5
        CLOPT="-A -- -Stigger2_ip -DTPCC"
        RQADDR=q7q REPLYQ=Y

Q         SRVGRP=Q8
        SRVID=800
        MIN=4 MAX=5
        CLOPT="-A -- -Stigger2_ip -DTPCC"
        RQADDR=q8q REPLYQ=Y

Q         SRVGRP=Q9
        SRVID=900
        MIN=4 MAX=5
        CLOPT="-A -- -Stigger2_ip -DTPCC"
        RQADDR=q9q REPLYQ=Y

delivery SRVGRP=GROUPDEL
        SRVID=1000
        MIN=4 MAX=6
        CLOPT="-A -- -Stigger2_ip -DTPCC"
        RQADDR=delq REPLYQ=N


*SERVICES
```

Client Systems Configuration

```
Date . . . . . . . . . . . . . .  06/07/1999
Time . . . . . . . . . . . . . .  10:10:57
```

```
Product  . . . . . . . . . . . .  ProLiant 1600

Machine ID
  From System Board  . . . . . .  CPQ0689

Processor  . . . . . . . . . . .  Pentium II(R) at
450 MHz
  Slot . . . . . . . . . . . . .  2
  Secondary Cache  . . . . . . .  512K
  CPU ID . . . . . . . . . . . .  0652

Processor  . . . . . . . . . . .  Pentium II(R) at
450 MHz
  Slot . . . . . . . . . . . . .  1
  Secondary Cache  . . . . . . .  512K
  CPU ID . . . . . . . . . . . .  0652

Processor(s) Mapped Out  . . . .  None

Numeric Coprocessor  . . . . . .  Integrated 387-
Compatible

Expansion Bus  . . . . . . . . .  ISA, PCI

System Identification Number . .  D826BWV10060

CPU Mode . . . . . . . . . . . .  Real Mode

System ROM
    Revision . . . . . . . . . .  01/29/1999
      Family . . . . . . . . . .  P8
    Flashable  . . . . . . . . .  Yes
    Supports F10 partition . . .  Yes

Video Controller ROM
    Revision . . . . . . . . . .  1.31 (Cirrus)

Option ROMs
  Address Range  . . . . . . . .  C0000 - C7FFF
    Data Dump  . . . . . . . . .  (CL-GD5446 PCI VGA
BIOS Version 1.31      Copyright 1992-1996...)

  Address Range  . . . . . . . .  E8000 - EDFFF
    Data Dump  . . . . . . . . .  ( CPQSCSI d)

Bootblock ROM  . . . . . . . . .  04/07/1998

Standby Recovery Server
  Status . . . . . . . . . . . .  Disabled
  COM Port . . . . . . . . . . .  COM1
  Server Configuration . . . . .  Recovery
  Timeout Value  . . . . . . . .  1 minutes

Memory Boards Identified:
  System Board
    DIMM Slot  1 (SDRAM) . . . .   256 Megabytes
    DIMM Slot  2 (SDRAM) . . . .    64 Megabytes
    DIMM Slot  3 (SDRAM) . . . .    64 Megabytes
    DIMM Slot  4 (SDRAM) . . . .   128 Megabytes
Total Compaq Memory  . . . . . .   512 Megabytes

Keyboard . . . . . . . . . . . .  Enhanced
```

```
LPT Ports  . . . . . . . . . . .  LPT1 (Address 3BC)

COM Ports  . . . . . . . . . . .  COM1 (Address 3F8)
                                  COM2 (Address 2F8)

Diskette Drive A . . . . . . . .  1.44 Megabyte (3.5
inch)

Graphics Mode  . . . . . . . . .  03 (80-Column Text)


Primary Monitor attached to  . .  Cirrus CL-GD5446
Graphics Controller
                                  with Video Graphics
Color Monitor

Total Video Memory . . . . . . .  1024 Kbytes

Base Memory
  System Total . . . . . . . . .    640 Kbytes
  Amount Free  . . . . . . . . .    597 Kbytes
(611888 Bytes)

Extended Memory
  System Total . . . . . . . . .  523264 Kbytes

Expanded Memory
  LIM Driver Support . . . . . .  LIM driver not
loaded

Operating System . . . . . . . .  MS-DOS version 7.00
(from diskette)

Status of file: C:\IO.SYS
  Date and Time  . . . . . . . .  07/07/1997
17:16:26
  Size . . . . . . . . . . . . .  0 Bytes

Status of file: C:\MSDOS.SYS
  Date and Time  . . . . . . . .  07/07/1997
17:16:26
  Size . . . . . . . . . . . . .  0 Bytes

-----------------------------------------------------
-----------------------
** Dump of C:\CONFIG.SYS
-----------------------------------------------------
-----------------------
** End of file

Status of file: C:\COMMAND.COM
*** File does not exist

-----------------------------------------------------
-----------------------
** Dump of C:\AUTOEXEC.BAT
-----------------------------------------------------
-----------------------
** End of file

Environment variables
  PATH=
  PROMPT=$P$G
```

```
  COMSPEC=A:\COMMAND.COM
  CMDLINE=inspect /u
End of environment

Revisions Table

Previous Revisions
  I/O Board Revision . . . . . .  02
    Assembly Version . . . . . .  1
    Functional Revision Level  .  B

Current Revisions
  I/O Board Revision . . . . . .  02
    Assembly Version . . . . . .  1
    Functional Revision Level  .  B

Memory Allocation (including INSPECT)
  PSP   SIZE    NAME          TRAPPED INTERRUPTS
  ----  ------  ------------  -----------------------
---------------
  08D8  007200  COMMAND.COM   F7h  FDh  2Fh  2Eh  24h
23h  22h
  0AA3  206336  INSPECT.EXE   FBh  F8h  F1h  EFh  EBh
E7h  3Fh  00h

System Configuration Memory
  00 - 0F :    11 00 11 00    10 00 01 07    06 99 26
82    50 80 00 00
  10 - 1F :    40 00 00 00    03 80 02 00    3C 00 00
00    00 00 00 02
  20 - 2F :    00 00 00 00    7F 20 20 40    00 92 00
00    00 10 02 A4
  30 - 3F :    00 3C 19 80    00 00 XX XX    XX XX XX
XX    XX XX XX XX

BIOS Data Area
  40:0000 :    F8 03 F8 02    00 00 00 00    BC 03 00
00    00 00 08 02
  40:0010 :    27 44 00 80    02 00 00 00    00 00 1E
00    1E 00 00 00
  40:0020 :    00 00 00 00    00 00 00 00    00 00 00
00    00 00 00 00
  40:0030 :    00 00 00 00    00 00 00 00    00 00 00
00    00 00 01 01
  40:0040 :    21 00 04 00    00 2C 01 08    02 03 50
00    00 10 00 00
  40:0050 :    00 14 00 00    00 00 00 00    00 00 00
00    00 00 00 00
  40:0060 :    0E 0D 00 D4    03 29 30 C2    11 45 77
00    FA 2E 0A 00
  40:0070 :    00 00 00 12    00 01 00 00    14 14 14
14    01 01 01 01
  40:0080 :    1E 00 3E 00    18 10 00 60    F9 11 0B
01    00 00 00 05
  40:0090 :    17 00 00 00    2C 00 10 00    00 00 00
00    00 00 00 00
  40:00A0 :    00 00 00 00    00 00 00 00    42 54 00
C0    00 00 00 00
  40:00B0 :    00 00 00 00    00 00 00 00    00 00 00
00    00 00 00 00
  40:00C0 :    00 00 00 00    00 00 00 00    00 00 00
00    00 00 00 00
```

```
  40:00D0 :    00 00 00 00    00 00 00 00    00 00 00
00    00 00 00 00
  40:00E0 :    00 00 00 00    00 00 00 00    00 00 00
00    00 00 00 00
  40:00F0 :    00 00 00 00    00 00 00 00    00 00 00
00    00 00 00 00


Interrupt Vector Table (including INSPECT)
  00 - 03 :    0AB3:0555      0070:0465
080F:0016      0070:0465
  04 - 07 :    0070:0465      F000:FF54
F000:93CC      F000:9BD0
  08 - 0B :    080F:001F      080F:0028
F000:9BD0      F000:9BD0
  0C - 0F :    F000:9BD0      F000:9BD0
080F:009A      0070:0465
  10 - 13 :    C000:3342      F000:F84D
F000:F841      0070:03EE
  14 - 17 :    F000:F71E      0249:0240
0070:042D      F000:EFD2
  18 - 1B :    F000:C2E8      08D0:002F
F000:FE6E      0070:045F
  1C - 1F :    F000:FF53      C000:1E2B
0000:0522      C000:6032
  20 - 23 :    00C9:0FA8      00C9:0FB2
08D8:0314      08D8:016D
  24 - 27 :    08D8:0178      00C9:0FBC
00C9:0FC6      00C9:0FD0
  28 - 2B :    00C9:106F      0070:0466
00C9:106F      00C9:106F
  2C - 2F :    00C9:106F      00C9:106F
08D8:0162      08D9:01CC
  30 - 33 :    C90F:E4EA      F000:9B00
00C9:106F      00C9:106F
  34 - 37 :    00C9:106F      00C9:106F
00C9:106F      00C9:106F
  38 - 3B :    00C9:106F      00C9:106F
00C9:106F      00C9:106F
  3C - 3F :    00C9:106F      00C9:106F
00C9:106F      1BC1:04FD
  40 - 43 :    F000:EC59      0000:0000
F000:F065      C000:5C32
  44 - 47 :    F000:9BD0      F000:9BD0
0000:0000      F000:9BD0
  48 - 4B :    F000:9BD0      F000:9BD0
F000:9BD0      F000:9BD0
  4C - 4F :    F000:9BD0      F000:9BD0
F000:9BD0      0070:04FC
  50 - 53 :    F000:9BD0      F000:9BD0
F000:9BD0      F000:9BD0
  54 - 57 :    F000:9BD0      F000:9BD0
F000:9BD0      F000:9BD0
  58 - 5B :    F000:9BD0      F000:9BD0
F000:9BD0      F000:9BD0
  5C - 5F :    F000:9BD0      F000:9BD0
F000:9BD0      F000:9BD0
  60 - 63 :    0000:0000      0000:0000
0000:0000      0000:0000
  64 - 67 :    0000:0000      0000:0000
0000:0000      0000:0000
  68 - 6B :    F000:9BD0      F000:9BD0
F000:9BD0      F000:9BD0
```

```
  6C - 6F :    F000:9BD0      C000:3342
F000:9BD0      F000:9BD0
  70 - 73 :    080F:0035      F000:9C1F
080F:00B2      080F:00CA
  74 - 77 :    080F:00E2      F000:9C28
080F:00FA      F000:9BD0
  78 - 7B :    0000:0000      0000:0000
0000:0000      0000:0000
  7C - 7F :    0000:0000      0000:0000
0000:0000      0000:0000
  80 - 83 :    0000:0000      0000:0000
0000:0000      0000:0000
  84 - 87 :    0000:0000      0000:0000
0000:0000      0000:0000
  88 - 8B :    0000:0000      0000:0000
0000:0000      0000:0000
  8C - 8F :    0000:0000      0000:0000
0000:0000      0000:0000
  90 - 93 :    0000:0000      0000:0000
0000:0000      0000:0000
  94 - 97 :    0000:0000      0000:0000
0000:0000      0000:0000
  98 - 9B :    0000:0000      0000:0000
0000:0000      0000:0000
  9C - 9F :    0000:0000      0000:0000
0000:0000      0000:0000
  A0 - A3 :    0000:0000      0000:0000
0000:0000      0000:0000
  A4 - A7 :    0000:0000      0000:0000
0000:0000      0000:0000
  A8 - AB :    0000:0000      0000:0000
0000:0000      0000:0000
  AC - AF :    0000:0000      0000:0000
0000:0000      0000:0000
  B0 - B3 :    0000:0000      0000:0000
0000:0000      0000:0000
  B4 - B7 :    0000:0000      0000:0000
0000:0000      0000:0000
  B8 - BB :    0000:0000      0000:0000
0000:0000      0000:0000
  BC - BF :    0000:0000      0000:0000
0000:0000      0000:0000
  C0 - C3 :    0000:0000      0000:0000
0000:0000      0000:0000
  C4 - C7 :    0000:0000      0000:0000
0000:0000      0000:0000
  C8 - CB :    0000:0000      0000:0000
0000:0000      0000:0000
  CC - CF :    0000:0000      0000:0000
0000:0000      0000:0000
  D0 - D3 :    0000:0000      0000:0000
0000:0000      0000:0000
  D4 - D7 :    0000:0000      0000:0000
0000:0000      0000:0000
  D8 - DB :    0000:0000      0000:0000
0000:0000      0000:0000
  DC - DF :    0000:0000      0000:0000
0000:0000      0000:0000
  E0 - E3 :    0000:0000      0000:0000
0000:0000      0000:0000
  E4 - E7 :    0000:0000      0000:0000
0000:0000      0F10:0CFD
```

```
  E8 - EB :    0083:0009      0006:1FB9
0006:1FB7      0046:C300
  EC - EF :    0000:1DD3      0020:0203
0000:031F      0202:C000
  F0 - F3 :    C000:0008      0BE9:0E06
03CE:03DA      0008:5050
  F4 - F7 :    00E8:655F      0203:0000
0000:03C4      0A2E:03C4
  F8 - FB :    3394:0000      FF01:000F
0000:0000      00F4:D3E5
  FC - FF :    0246:0A00      0A00:0000
E0FB:0049      0003:0A1F

PCI Devices Information
  Signature  . . . . . . . . . .  PCI
  Config Mechanism #1  . . . . .  Supported
  Config Mechanism #2  . . . . .  Not Supported
  Spec Cycle for Config #1 . . .  Supported
  Spec Cycle for Config #2 . . .  Not Supported
  BIOS Interface Version . . . .  2.10
  Last PCI Bus Number  . . . . .  1
  Number of PCI Devices  . . . .  5

  Bus Number . . . . . . . . . .  0
  Device Number  . . . . . . . .  11
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  0
  Vendor ID  . . . . . . . . . .  1013h
  Device ID  . . . . . . . . . .  00B8h
  Revision ID  . . . . . . . . .  45h
  Device Type  . . . . . . . . .  VGA Compatible
Controller
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFFF0000h
  IRQ Line . . . . . . . . . . .  255
  IRQ Pin  . . . . . . . . . . .  Not Used
  Memory Address Base  . . . . .  C4000000h
  Memory Address Length  . . . .  2000000h
  Memory Address Base  . . . . .  C6DFF000h
  Memory Address Length  . . . .  1000h

  Bus Number . . . . . . . . . .  0
  Device Number  . . . . . . . .  18
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  6
  Vendor ID  . . . . . . . . . .  0E11h
  Device ID  . . . . . . . . . .  1229h
  Revision ID  . . . . . . . . .  05h
  Device Type  . . . . . . . . .  Ethernet Controller
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFF00000h
  IRQ Line . . . . . . . . . . .  5
  IRQ Pin  . . . . . . . . . . .  INTA#
  Memory Address Base  . . . . .  C3FFF000h
  Memory Address Length  . . . .  1000h
  IO Address Base  . . . . . . .  2000h
  IO Address Length  . . . . . .  20h
  Memory Address Base  . . . . .  C6E00000h
  Memory Address Length  . . . .  100000h

  Bus Number . . . . . . . . . .  1
  Device Number  . . . . . . . .  7
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  0
  Vendor ID  . . . . . . . . . .  0E11h
  Device ID  . . . . . . . . . .  AE43h
  Revision ID  . . . . . . . . .  10h
  Device Type  . . . . . . . . .  Other Network
Controller
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFFF0000h
  IRQ Line . . . . . . . . . . .  9
  IRQ Pin  . . . . . . . . . . .  INTA#
  IO Address Base  . . . . . . .  3800h
  IO Address Length  . . . . . .  10h
  Memory Address Base  . . . . .  C6FFDDF0h
  Memory Address Length  . . . .  10h

  Bus Number . . . . . . . . . .  1
  Device Number  . . . . . . . .  9
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  0
  Vendor ID  . . . . . . . . . .  0E11h
  Device ID  . . . . . . . . . .  000Fh
  Revision ID  . . . . . . . . .  14h
  Device Type  . . . . . . . . .  SCSI Bus Controller
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  0h
  IRQ Line . . . . . . . . . . .  10
  IRQ Pin  . . . . . . . . . . .  INTA#
  IO Address Base  . . . . . . .  3000h
  IO Address Length  . . . . . .  100h
  Memory Address Base  . . . . .  C6FFDF00h
  Memory Address Length  . . . .  100h
  Memory Address Base  . . . . .  C6FFF000h
  Memory Address Length  . . . .  1000h

  Bus Number . . . . . . . . . .  1
  Device Number  . . . . . . . .  9
  Function Number  . . . . . . .  01h
  Slot Number  . . . . . . . . .  0
  Vendor ID  . . . . . . . . . .  0E11h
  Device ID  . . . . . . . . . .  000Fh
  Revision ID  . . . . . . . . .  14h
  Device Type  . . . . . . . . .  SCSI Bus Controller
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  0h
  IRQ Line . . . . . . . . . . .  11
  IRQ Pin  . . . . . . . . . . .  INTB#
  IO Address Base  . . . . . . .  3400h
  IO Address Length  . . . . . .  100h
  Memory Address Base  . . . . .  C6FFDE00h
  Memory Address Length  . . . .  100h
  Memory Address Base  . . . . .  C6FFE000h
  Memory Address Length  . . . .  1000h
```

ProLiant 1600 is a trademark of Compaq Computer Corporation.

SUT System Configuration

```
Date . . . . . . . . . . . . . .  06/07/1999
Time . . . . . . . . . . . . . .  09:58:51

Product  . . . . . . . . . . . .  PROLIANT 7000

Machine ID
  From System Board  . . . . . .  CPQ1601

Processor  . . . . . . . . . . .  Pentium III(R) Xeon
at 500 MHz
  Slot . . . . . . . . . . . . .  1
  Secondary Cache  . . . . . . .  2048K
  CPU ID . . . . . . . . . . . .  0672

Processor  . . . . . . . . . . .  Pentium III(R) Xeon
at 500 MHz
  Slot . . . . . . . . . . . . .  2
  Secondary Cache  . . . . . . .  2048K
  CPU ID . . . . . . . . . . . .  0672

Processor  . . . . . . . . . . .  Pentium III(R) Xeon
at 500 MHz
  Slot . . . . . . . . . . . . .  3
  Secondary Cache  . . . . . . .  2048K
  CPU ID . . . . . . . . . . . .  0672

Processor  . . . . . . . . . . .  Pentium III(R) Xeon
at 500 MHz
  Slot . . . . . . . . . . . . .  4
  Secondary Cache  . . . . . . .  2048K
  CPU ID . . . . . . . . . . . .  0672

Processor(s) Mapped Out  . . . .  None

Numeric Coprocessor  . . . . . .  Integrated 387-
Compatible

Expansion Bus  . . . . . . . . .  ISA, PCI

System Identification Number . .  None

CPU Mode . . . . . . . . . . . .  Real Mode

System ROM
    Revision . . . . . . . . . .  02/26/1999
      Family . . . . . . . . . .  P43
    Flashable  . . . . . . . . .  Yes
    Supports F10 partition . . .  Yes

Video Controller ROM
    Revision . . . . . . . . . .  3.95

Option ROMs
  Address Range  . . . . . . . .  C0000 - C7FFF
    Data Dump  . . . . . . . . .  (1998/08/26 10:10)

  Address Range  . . . . . . . .  C8000 - CBFFF
    Data Dump  . . . . . . . . .  (04/22/98
SurefireSmart Array Option ROM/BIOS (C)Copyright
CO...)

  Address Range  . . . . . . . .  E8000 - EDFFF
    Data Dump  . . . . . . . . .  ( CPQSCSI d)

Bootblock ROM  . . . . . . . . .  07/17/1998

Standby Recovery Server
  Status . . . . . . . . . . . .  Disabled
  COM Port . . . . . . . . . . .  COM1
  Server Configuration . . . . .  Recovery
  Timeout Value  . . . . . . . .  1 minutes

Memory Boards Identified:
  Expansion Board
    Bank  1 (EDO)  . . . . . . .   512 Megabytes
    Bank  2 (EDO)  . . . . . . .   512 Megabytes
    Bank  3 (EDO)  . . . . . . .   512 Megabytes
    Bank  4 (EDO)  . . . . . . .   512 Megabytes
    Bank  5 (EDO)  . . . . . . .   512 Megabytes
    Bank  6 (EDO)  . . . . . . .   512 Megabytes
    Bank  7 (EDO)  . . . . . . .   512 Megabytes
    Bank  8 (EDO)  . . . . . . .   512 Megabytes
Total Compaq Memory  . . . . . .  4096 Megabytes

Keyboard . . . . . . . . . . . .  Enhanced

LPT Ports  . . . . . . . . . . .  LPT1 (Address 3BC)

COM Ports  . . . . . . . . . . .  COM1 (Address 3F8)
                                  COM2 (Address 2F8)

Diskette Drive A . . . . . . . .  1.44 Megabyte (3.5
inch)

Graphics Mode  . . . . . . . . .  03 (80-Column Text)


Primary Monitor attached to  . .  ATI RAGE IIC PCI
Graphics Controller
with Video Graphics Color Monitor

Base Memory
  System Total . . . . . . . . .    638 Kbytes
  Amount Free  . . . . . . . . .    596 Kbytes
(611008 Bytes)

Extended Memory
  System Total . . . . . . . . .  4193280 Kbytes

Expanded Memory
  LIM Driver Support . . . . . .  LIM driver not
loaded

Operating System . . . . . . . .  MS-DOS version 7.00
(from diskette)

Status of file: C:\IO.SYS
  Date and Time  . . . . . . . .  11/18/1998
13:54:42
  Size . . . . . . . . . . . . .  0 Bytes
```

```
Status of file: C:\MSDOS.SYS
  Date and Time  . . . . . . . .  11/18/1998
13:54:42
  Size . . . . . . . . . . . . .  0 Bytes

----------------------------------------------------
-----------------------
** Dump of C:\CONFIG.SYS
----------------------------------------------------
-----------------------
** End of file

Status of file: C:\COMMAND.COM
*** File does not exist

----------------------------------------------------
-----------------------
** Dump of C:\AUTOEXEC.BAT
----------------------------------------------------
-----------------------
** End of file

Environment variables
  PATH=
  PROMPT=$P$G
  COMSPEC=A:\COMMAND.COM
  CMDLINE=inspect /u
End of environment

Revisions Table

Previous Revisions
  Power backplane revision . . .  01
    Assembly Version . . . . . .  1
    Functional Revision Level  .  A

Current Revisions
  I/O Board Revision . . . . . .  01
    Assembly Version . . . . . .  1
    Functional Revision Level  .  A
  Memory Board  1 Revision . . .  01
    Assembly Version . . . . . .  1
    Functional Revision Level  .  A
  Power backplane revision . . .  01
    Assembly Version . . . . . .  1
    Functional Revision Level  .  A

System Configuration Utility . .  Version 2.45

Extended Non-volatile Memory
-------- ----------- ------
D:\DIRECT.OUT 0
  Slot Type  . . . . . . . . . .  Embedded
  Board ID . . . . . . . . . . .  CPQ1601
  CFG File Extension
    Revision Level . . . . . . .  2.21
  Type Entry(s)  . . . . . . . .  MSD,FPYCTL
    IRQ Entry(s):
      IRQ  6, Not Shared, Edge Triggered
    DMA Channel(s):
      Channel 2, Not Shared
      Timing: Type B
```

```
      Transfer Size: 8-bit (byte)
    Port Range(s):
      03F0h - 03F5h, Not Shared
      03F6h - 03F7h, Shared

  Type Entry(s)  . . . . . . . .
MSD,UNIT0,FPYDRV;TYP=4
  Type Entry(s)  . . . . . . . .
MSD,UNIT1,FPYDRV;TYP=0
  Type Entry(s)  . . . . . . . .  MSD
  Type Entry(s)  . . . . . . . .  MSD,DSKCTL;CTL2
    IRQ Entry(s):
      IRQ 14, Not Shared, Edge Triggered
    Port Range(s):
      01F0h - 01F7h, Not Shared
      03F6h, Shared
      11F1h, Not Shared

  Type Entry(s)  . . . . . . . .  MSD,UNIT0,DSKDRV;CD
  Type Entry(s)  . . . . . . . .  MSD,UNIT1,DSKDRV
  Memory Entry(s):
                                          Range
Size
                                          -----
----
    ROM: Other,    Cacheable         896K -        1M
128K

  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Memory Entry(s):
                                          Range
Size
                                          -----
----
    RAM: System,   Cacheable           0K -      640K
640K

  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Memory Entry(s):
                                          Range
Size
                                          -----
----
    RAM: System,   Cacheable           1M -       16M
15M

  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Memory Entry(s):
                                          Range
Size
                                          -----
----
    RAM: System,   Cacheable          16M -       64M
48M
    RAM: System,   Cacheable          64M -      128M
64M
    RAM: System,   Cacheable         128M -      192M
64M
    RAM: System,   Cacheable         192M -      256M
64M
    RAM: System,   Cacheable         256M -      320M
64M
```

```
    RAM: System,   Cacheable         320M -      384M
64M
    RAM: System,   Cacheable         384M -      448M
64M
    RAM: System,   Cacheable         448M -      512M
64M
    RAM: System,   Cacheable         512M -      576M
64M

  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Memory Entry(s):
                                          Range
Size
                                          -----
----
    RAM: System,   Cacheable         576M -      640M
64M
    RAM: System,   Cacheable         640M -      704M
64M
    RAM: System,   Cacheable         704M -      768M
64M
    RAM: System,   Cacheable         768M -      832M
64M
    RAM: System,   Cacheable         832M -      896M
64M
    RAM: System,   Cacheable         896M -      960M
64M
    RAM: System,   Cacheable         960M -     1024M
64M
    RAM: System,   Cacheable        1024M -     1088M
64M
    RAM: System,   Cacheable        1088M -     1152M
64M

  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Memory Entry(s):
                                          Range
Size
                                          -----
----
    RAM: System,   Cacheable        1152M -     1216M
64M
    RAM: System,   Cacheable        1216M -     1280M
64M
    RAM: System,   Cacheable        1280M -     1344M
64M
    RAM: System,   Cacheable        1344M -     1408M
64M
    RAM: System,   Cacheable        1408M -     1472M
64M
    RAM: System,   Cacheable        1472M -     1536M
64M
    RAM: System,   Cacheable        1536M -     1600M
64M
    RAM: System,   Cacheable        1600M -     1664M
64M
    RAM: System,   Cacheable        1664M -     1728M
64M

  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Memory Entry(s):
                                          Range
Size
```

```
                                       -----
----
    RAM: System,   Cacheable        1728M -     1792M
64M
    RAM: System,   Cacheable        1792M -     1856M
64M
    RAM: System,   Cacheable        1856M -     1920M
64M
    RAM: System,   Cacheable        1920M -     1984M
64M
    RAM: System,   Cacheable        1984M -     2048M
64M
    RAM: System,   Cacheable        2048M -     2112M
64M
    RAM: System,   Cacheable        2112M -     2176M
64M
    RAM: System,   Cacheable        2176M -     2240M
64M
    RAM: System,   Cacheable        2240M -     2304M
64M

  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Memory Entry(s):
                                       Range
Size
                                       -----
----
    RAM: System,   Cacheable        2304M -     2368M
64M
    RAM: System,   Cacheable        2368M -     2432M
64M
    RAM: System,   Cacheable        2432M -     2496M
64M
    RAM: System,   Cacheable        2496M -     2560M
64M
    RAM: System,   Cacheable        2560M -     2624M
64M
    RAM: System,   Cacheable        2624M -     2688M
64M
    RAM: System,   Cacheable        2688M -     2752M
64M
    RAM: System,   Cacheable        2752M -     2816M
64M
    RAM: System,   Cacheable        2816M -     2880M
64M

  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Memory Entry(s):
                                       Range
Size
                                       -----
----
    RAM: System,   Cacheable        2880M -     2944M
64M
    RAM: System,   Cacheable        2944M -     3008M
64M
    RAM: System,   Cacheable        3008M -     3072M
64M
    RAM: System,   Cacheable        3072M -     3136M
64M
    RAM: System,   Cacheable        3136M -     3200M
64M
```

```
    RAM: System,   Cacheable        3200M -     3264M
64M
    RAM: System,   Cacheable        3264M -     3328M
64M
    RAM: System,   Cacheable        3328M -     3392M
64M
    RAM: System,   Cacheable        3392M -     3456M
64M

  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Memory Entry(s):
                                       Range
Size
                                       -----
----
    RAM: System,   Cacheable        3456M -     3520M
64M
    RAM: System,   Cacheable        3520M -     3584M
64M
    RAM: System,   Cacheable        3584M -     3648M
64M
    RAM: System,   Cacheable        3648M -     3712M
64M
    RAM: System,   Cacheable        3712M -     3776M
64M
    RAM: System,   Cacheable        3776M -     3840M
64M
    RAM: System,   Cacheable        3840M -     3904M
64M
    RAM: System,   Cacheable        3904M -     3936M
32M

  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Type Entry(s)  . . . . . . . .  MEM;COMPAQ
  Type Entry(s)  . . . . . . . .  COM,ASY;COM1;A
    IRQ Entry(s):
      IRQ  4, Not Shared, Edge Triggered
    Port Range(s):
      03F8h - 03FFh, Not Shared

  Type Entry(s)  . . . . . . . .  COM,ASY;COM2;B
    IRQ Entry(s):
      IRQ  3, Not Shared, Edge Triggered
    Port Range(s):
      02F8h - 02FFh, Not Shared

  Type Entry(s)  . . . . . . . .  PAR;LPT1
    IRQ Entry(s):
      IRQ  7, Not Shared, Edge Triggered
    DMA Channel(s):
      Channel 0, Not Shared
```

```
      Timing: ISA Compatible
      Transfer Size: 8-bit (byte)
    Port Range(s):
      03BCh - 03BEh, Not Shared

  Type Entry(s)  . . . . . . . .  PTR,8042
    IRQ Entry(s):
      IRQ 12, Not Shared, Edge Triggered
    IRQ Entry(s):
      IRQ 13, Shared, Edge Triggered
  Type Entry(s)  . . . . . . . .  OTH,A20
  Type Entry(s)  . . . . . . . .  OTH,SOFTNMI
  Type Entry(s)  . . . . . . . .  OTH,FLSFNMI
  Type Entry(s)  . . . . . . . .  OTH,BUSNMI
  Type Entry(s)  . . . . . . . .  OTH,DSKTDMA
  Type Entry(s)  . . . . . . . .  OTH,REFRESH
  Type Entry(s)  . . . . . . . .  OTH,PERR
  Type Entry(s)  . . . . . . . .  OTH,HPK
  Type Entry(s)  . . . . . . . .  OTH,ACPI
    IRQ Entry(s):
      IRQ  9, Not Shared, Level Triggered
  Type Entry(s)  . . . . . . . .  OTH,WARMBOOT
  Type Entry(s)  . . . . . . . .  OTH,IOQ
  Type Entry(s)  . . . . . . . .  OTH,F1BP
  Type Entry(s)  . . . . . . . .  OTH,TABLE;FULLTABLE
    IRQ Entry(s):
      IRQ 13, Shared, Edge Triggered
  Type Entry(s)  . . . . . . . .  ISA;MAP
  Free Form Text . . . . . . . .  21 82 83 84 85 86
87 88 89 8A 8B 8C E0 E0
                                  E0 00 00 00 00 00
00 00 00 00 00 00 00 00
                                  00 00
  Type Entry(s)  . . . . . . . .  ISA;PCIMAP
  Free Form Text . . . . . . . .  03 00 08 04 00 10
05 04 08 06 04 10 07 08
                                  08 08 08 10 09 08
18 0A 08 20 0B 08 28
D:\DIRECT.OUT 2
  Slot Type  . . . . . . . . . .  Expansion
  Board ID . . . . . . . . . . .  CPQ1603
  CFG File Extension
    Revision Level . . . . . . .  2.21

Memory Allocation (including INSPECT)
  PSP   SIZE    NAME          TRAPPED INTERRUPTS
  ----  ------  ------------  -----------------------
---------------
  088F  007200  COMMAND.COM   2Fh  2Eh  24h  23h  22h
  0A5A  206336  INSPECT.EXE   DDh  F9h  F4h  F3h  F2h
F1h  EFh  EBh
                              E5h  3Fh  00h

System Configuration Memory
  00 - 0F :    03 00 59 00    09 00 01 07    06 99 26
82    50 80 00 00
  10 - 1F :    40 00 00 00    03 80 02 00    3C 00 00
00    00 00 00 02
  20 - 2F :    00 00 00 00    7F 20 20 40    00 92 00
00    00 10 02 A4
  30 - 3F :    00 3C 19 80    00 00 XX XX    XX XX XX
XX    XX XX XX XX
```

```
BIOS Data Area
  40:0000 :    F8 03 F8 02    00 00 00 00    BC 03 00
00    00 00 80 9F
  40:0010 :    27 44 00 7E    02 00 00 00    00 00 1E
00    1E 00 00 00
  40:0020 :    00 00 00 00    00 00 00 00    00 00 00
00    00 00 00 00
  40:0030 :    00 00 00 00    00 00 00 00    00 00 00
00    00 00 01 01
  40:0040 :    25 00 04 00    00 2C 01 08    02 03 50
00    00 10 00 00
  40:0050 :    00 14 00 00    00 00 00 00    00 00 00
00    00 00 00 00
  40:0060 :    0E 0D 00 D4    03 29 30 C2    11 05 77
00    6D FB 09 00
  40:0070 :    00 00 00 12    00 01 00 00    14 14 14
14    01 01 01 01
  40:0080 :    1E 00 3E 00    18 10 00 60    F9 11 0B
01    00 00 00 05
  40:0090 :    17 00 00 00    2C 00 10 00    00 00 00
00    00 00 00 00
  40:00A0 :    00 00 00 00    00 00 00 00    7C 13 00
C0    00 00 00 00
  40:00B0 :    00 00 00 00    00 00 00 00    00 00 00
00    00 00 00 00
  40:00C0 :    00 00 00 00    00 00 00 00    00 00 00
00    00 00 00 00
  40:00D0 :    00 00 00 00    00 00 00 00    00 00 00
00    00 00 00 00
  40:00E0 :    00 00 00 00    00 00 00 00    00 00 00
00    00 00 00 00
  40:00F0 :    00 00 00 00    00 00 00 00    00 00 00
00    00 00 00 00


Interrupt Vector Table (including INSPECT)
  00 - 03 :    0A6A:0555      0070:0465
07C6:0016      0070:0465
  04 - 07 :    0070:0465      F000:FF54
F000:93CC      F000:9BD0
  08 - 0B :    07C6:001F      07C6:0028
F000:9BD0      F000:9BD0
  0C - 0F :    F000:9BD0      F000:9BD0
07C6:009A      0070:0465
  10 - 13 :    C000:12FE      F000:F84D
F000:F841      0070:03EE
  14 - 17 :    F000:D0A3      0206:0240
0070:042D      F000:EFD2
  18 - 1B :    F000:F06D      0887:002F
F000:FE6E      0070:045F
  1C - 1F :    F000:FF53      F000:0000
0000:0522      C000:2039
  20 - 23 :    00C9:0FA8      00C9:0FB2
088F:0314      088F:016D
  24 - 27 :    088F:0178      00C9:0FBC
00C9:0FC6      00C9:0FD0
  28 - 2B :    00C9:106F      0070:0466
00C9:106F      00C9:106F
  2C - 2F :    00C9:106F      00C9:106F
088F:0162      0890:01CC
  30 - 33 :    C90F:E4EA      F000:9B00
00C9:106F      00C9:106F
```

```
  34 - 37 :    00C9:106F      00C9:106F
00C9:106F      00C9:106F
  38 - 3B :    00C9:106F      00C9:106F
00C9:106F      00C9:106F
  3C - 3F :    00C9:106F      00C9:106F
00C9:106F      1B78:04FD
  40 - 43 :    F000:EC59      0000:0000
F000:F065      C000:244C
  44 - 47 :    F000:9BD0      F000:9BD0
0000:0000      F000:9BD0
  48 - 4B :    F000:9BD0      F000:9BD0
F000:9BD0      F000:9BD0
  4C - 4F :    F000:9BD0      F000:9BD0
F000:9BD0      0070:04FC
  50 - 53 :    F000:9BD0      F000:9BD0
F000:9BD0      F000:9BD0
  54 - 57 :    F000:9BD0      F000:9BD0
F000:9BD0      F000:9BD0
  58 - 5B :    F000:9BD0      F000:9BD0
F000:9BD0      F000:9BD0
  5C - 5F :    F000:9BD0      F000:9BD0
F000:9BD0      F000:9BD0
  60 - 63 :    0000:0000      0000:0000
0000:0000      0000:0000
  64 - 67 :    0000:0000      0000:0000
0000:0000      0000:0000
  68 - 6B :    F000:9BD0      F000:9BD0
F000:9BD0      F000:9BD0
  6C - 6F :    F000:9BD0      C000:12FE
F000:9BD0      F000:9BD0
  70 - 73 :    07C6:0035      F000:9C1F
F000:9BD0      F000:9BD0
  74 - 77 :    07C6:00E2      F000:9C28
07C6:00FA      07C6:0112
  78 - 7B :    0000:0000      0000:0000
0000:0000      0000:0000
  7C - 7F :    0000:0000      0000:0000
0000:0000      0000:0000
  80 - 83 :    0000:0000      0000:0000
0000:0000      0000:0000
  84 - 87 :    0000:0000      0000:0000
0000:0000      0000:0000
  88 - 8B :    0000:0000      0000:0000
0000:0000      0000:0000
  8C - 8F :    0000:0000      0000:0000
0000:0000      0000:0000
  90 - 93 :    0000:0000      0000:0000
0000:0000      0000:0000
  94 - 97 :    0000:0000      0000:0000
0000:0000      0000:0000
  98 - 9B :    0000:0000      0000:0000
0000:0000      0000:0000
  9C - 9F :    0000:0000      0000:0000
0000:0000      0000:0000
  A0 - A3 :    0000:0000      0000:0000
0000:0000      0000:0000
  A4 - A7 :    0000:0000      0000:0000
0000:0000      0000:0000
  A8 - AB :    0000:0000      0000:0000
0000:0000      0000:0000
  AC - AF :    0000:0000      0000:0000
0000:0000      0000:0000
```

```
  B0 - B3 :    0000:0000      0000:0000
0000:0000      0000:0000
  B4 - B7 :    0000:0000      0000:0000
0000:0000      0000:0000
  B8 - BB :    0000:0000      0000:0000
0000:0000      0000:0000
  BC - BF :    0000:0000      0000:0000
0000:0000      0000:0000
  C0 - C3 :    0000:0000      0000:0000
0000:0000      0000:0000
  C4 - C7 :    0000:0000      0000:0000
0000:0000      0000:0000
  C8 - CB :    0000:0000      0000:0000
0000:0000      0000:0000
  CC - CF :    0000:0000      0000:0000
0000:0000      0000:0000
  D0 - D3 :    0000:0000      0000:0000
0000:0000      0000:0000
  D4 - D7 :    0000:0000      0000:0000
0000:0000      0000:0000
  D8 - DB :    0000:0000      0000:0000
0000:0000      0000:0000
  DC - DF :    0000:0000      9394:0000
7083:F000      0000:0402
  E0 - E3 :    A700:0030      7202:F000
A6DC:0950      03E8:2851
  E4 - E7 :    00A4:0080      0019:A6D0
9C3C:80D8      6972:1000
  E8 - EB :    F000:A6C9      6972:0000
7292:1000      007C:F886
  EC - EF :    0005:081F      7006:5E9F
0046:7046      0800:7B00
  F0 - F3 :    7046:5FC4      1D08:13C1
13C1:5ECA      1400:0808
  F4 - F7 :    1BD0:7246      0101:7268
0000:0000      0000:601E
  F8 - FB :    601E:0020      14B7:6324
0098:0003      0000:0947
  FC - FF :    525F:72BE      0000:00F6
3E8A:0000      7246:F000

PCI Devices Information
  Signature  . . . . . . . . . .  PCI
  Config Mechanism #1  . . . . .  Supported
  Config Mechanism #2  . . . . .  Not Supported
  Spec Cycle for Config #1 . . .  Supported
  Spec Cycle for Config #2 . . .  Not Supported
  BIOS Interface Version . . . .  2.10
  Last PCI Bus Number  . . . . .  14
  Number of PCI Devices  . . . .  13

  Bus Number . . . . . . . . . .  0
  Device Number  . . . . . . . .  1
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  3
  Vendor ID  . . . . . . . . . .  0E11h
  Device ID  . . . . . . . . . .  0046h
  Revision ID  . . . . . . . . .  01h
  Device Type  . . . . . . . . .  Invalid Sub
Class(4)
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFFC0000h
  IRQ Line . . . . . . . . . . .  5
  IRQ Pin  . . . . . . . . . . .  INTA#
  Memory Address Base  . . . . .  F7AF0000h
  Memory Address Length  . . . .  1000h
  IO Address Base  . . . . . . .  2000h
  IO Address Length  . . . . . .  100h

  Bus Number . . . . . . . . . .  0
  Device Number  . . . . . . . .  2
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  4
  Vendor ID  . . . . . . . . . .  0E11h
  Device ID  . . . . . . . . . .  0046h
  Revision ID  . . . . . . . . .  01h
  Device Type  . . . . . . . . .  Invalid Sub
Class(4)
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFFC0000h
  IRQ Line . . . . . . . . . . .  15
  IRQ Pin  . . . . . . . . . . .  INTA#
  Memory Address Base  . . . . .  F7AE0000h
  Memory Address Length  . . . .  1000h
  IO Address Base  . . . . . . .  2400h
  IO Address Length  . . . . . .  100h

  Bus Number . . . . . . . . . .  0
  Device Number  . . . . . . . .  13
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  0
  Vendor ID  . . . . . . . . . .  1000h
  Device ID  . . . . . . . . . .  000Fh
  Revision ID  . . . . . . . . .  04h
  Device Type  . . . . . . . . .  SCSI Bus Controller
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  0h
  IRQ Line . . . . . . . . . . .  15
  IRQ Pin  . . . . . . . . . . .  INTA#
  IO Address Base  . . . . . . .  3000h
  IO Address Length  . . . . . .  100h
  Memory Address Base  . . . . .  F7AA0000h
  Memory Address Length  . . . .  100h
  Memory Address Base  . . . . .  F7A90000h
  Memory Address Length  . . . .  1000h

  Bus Number . . . . . . . . . .  0
  Device Number  . . . . . . . .  14
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  0
  Vendor ID  . . . . . . . . . .  1002h
  Device ID  . . . . . . . . . .  4756h
  Revision ID  . . . . . . . . .  7Ah
  Device Type  . . . . . . . . .  VGA Compatible
Controller
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFFE0000h
  IRQ Line . . . . . . . . . . .  255
  IRQ Pin  . . . . . . . . . . .  Not Used
  Memory Address Base  . . . . .  F6000000h
  Memory Address Length  . . . .  1000000h
  IO Address Base  . . . . . . .  3400h
  IO Address Length  . . . . . .  100h
  Memory Address Base  . . . . .  F7A80000h
  Memory Address Length  . . . .  1000h
```

```
  Bus Number . . . . . . . . . .  4
  Device Number  . . . . . . . .  1
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  5
  Vendor ID  . . . . . . . . . .  0E11h
  Device ID  . . . . . . . . . .  0046h
  Revision ID  . . . . . . . . .  01h
  Device Type  . . . . . . . . .  Invalid Sub
Class(4)
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFFC0000h
  IRQ Line . . . . . . . . . . .  5
  IRQ Pin  . . . . . . . . . . .  INTA#
  Memory Address Base  . . . . .  F7BF0000h
  Memory Address Length  . . . .  1000h
  IO Address Base  . . . . . . .  4000h
  IO Address Length  . . . . . .  100h

  Bus Number . . . . . . . . . .  4
  Device Number  . . . . . . . .  2
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  6
  Vendor ID  . . . . . . . . . .  0E11h
  Device ID  . . . . . . . . . .  0046h
  Revision ID  . . . . . . . . .  01h
  Device Type  . . . . . . . . .  Invalid Sub
Class(4)
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFFC0000h
  IRQ Line . . . . . . . . . . .  15
  IRQ Pin  . . . . . . . . . . .  INTA#
  Memory Address Base  . . . . .  F7BE0000h
  Memory Address Length  . . . .  1000h
  IO Address Base  . . . . . . .  4400h
  IO Address Length  . . . . . .  100h

  Bus Number . . . . . . . . . .  4
  Device Number  . . . . . . . .  9
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  0
  Vendor ID  . . . . . . . . . .  0E11h
  Device ID  . . . . . . . . . .  000Fh
  Revision ID  . . . . . . . . .  14h
  Device Type  . . . . . . . . .  SCSI Bus Controller
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  0h
  IRQ Line . . . . . . . . . . .  15
  IRQ Pin  . . . . . . . . . . .  INTA#
  IO Address Base  . . . . . . .  4800h
  IO Address Length  . . . . . .  100h
  Memory Address Base  . . . . .  F7BD0000h
  Memory Address Length  . . . .  100h
  Memory Address Base  . . . . .  F7BC0000h
  Memory Address Length  . . . .  1000h

  Bus Number . . . . . . . . . .  4
  Device Number  . . . . . . . .  9
  Function Number  . . . . . . .  01h
  Slot Number  . . . . . . . . .  0
  Vendor ID  . . . . . . . . . .  0E11h
  Device ID  . . . . . . . . . .  000Fh
  Revision ID  . . . . . . . . .  14h
  Device Type  . . . . . . . . .  SCSI Bus Controller
```

```
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  0h
  IRQ Line . . . . . . . . . . .  15
  IRQ Pin  . . . . . . . . . . .  INTB#
  IO Address Base  . . . . . . .  4C00h
  IO Address Length  . . . . . .  100h
  Memory Address Base  . . . . .  F7BB0000h
  Memory Address Length  . . . .  100h
  Memory Address Base  . . . . .  F7BA0000h
  Memory Address Length  . . . .  1000h

  Bus Number . . . . . . . . . .  8
  Device Number  . . . . . . . .  1
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  7
  Vendor ID  . . . . . . . . . .  0E11h
  Device ID  . . . . . . . . . .  0046h
  Revision ID  . . . . . . . . .  01h
  Device Type  . . . . . . . . .  Invalid Sub
Class(4)
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFFC0000h
  IRQ Line . . . . . . . . . . .  15
  IRQ Pin  . . . . . . . . . . .  INTA#
  Memory Address Base  . . . . .  F7FF0000h
  Memory Address Length  . . . .  1000h
  IO Address Base  . . . . . . .  5000h
  IO Address Length  . . . . . .  100h

  Bus Number . . . . . . . . . .  8
  Device Number  . . . . . . . .  2
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  8
  Vendor ID  . . . . . . . . . .  0E11h
  Device ID  . . . . . . . . . .  0046h
  Revision ID  . . . . . . . . .  01h
  Device Type  . . . . . . . . .  Invalid Sub
Class(4)
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFFC0000h
  IRQ Line . . . . . . . . . . .  15
  IRQ Pin  . . . . . . . . . . .  INTA#
  Memory Address Base  . . . . .  F7FE0000h
  Memory Address Length  . . . .  1000h
  IO Address Base  . . . . . . .  5400h
  IO Address Length  . . . . . .  100h

  Bus Number . . . . . . . . . .  8
  Device Number  . . . . . . . .  3
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  9
  Vendor ID  . . . . . . . . . .  0E11h
  Device ID  . . . . . . . . . .  0046h
  Revision ID  . . . . . . . . .  01h
  Device Type  . . . . . . . . .  Invalid Sub
Class(4)
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFFC0000h
  IRQ Line . . . . . . . . . . .  15
  IRQ Pin  . . . . . . . . . . .  INTA#
  Memory Address Base  . . . . .  F7FD0000h
  Memory Address Length  . . . .  1000h
  IO Address Base  . . . . . . .  5800h
  IO Address Length  . . . . . .  100h

  Bus Number . . . . . . . . . .  8
  Device Number  . . . . . . . .  4
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  10
  Vendor ID  . . . . . . . . . .  0E11h
  Device ID  . . . . . . . . . .  1229h
  Revision ID  . . . . . . . . .  05h
  Device Type  . . . . . . . . .  Ethernet Controller
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFF00000h
  IRQ Line . . . . . . . . . . .  10
  IRQ Pin  . . . . . . . . . . .  INTA#
  Memory Address Base  . . . . .  F7CF0000h
  Memory Address Length  . . . .  1000h
  IO Address Base  . . . . . . .  5C00h
  IO Address Length  . . . . . .  20h
  Memory Address Base  . . . . .  F7E00000h
  Memory Address Length  . . . .  100000h

  Bus Number . . . . . . . . . .  8
  Device Number  . . . . . . . .  5
  Function Number  . . . . . . .  00h
  Slot Number  . . . . . . . . .  11
  Vendor ID  . . . . . . . . . .  0E11h
  Device ID  . . . . . . . . . .  0046h
  Revision ID  . . . . . . . . .  01h
  Device Type  . . . . . . . . .  Invalid Sub
Class(4)
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFFC0000h
  IRQ Line . . . . . . . . . . .  15
  IRQ Pin  . . . . . . . . . . .  INTA#
  Memory Address Base  . . . . .  F7DF0000h
  Memory Address Length  . . . .  1000h
  IO Address Base  . . . . . . .  6000h
  IO Address Length  . . . . . .  100h



PROLIANT 7000 is a trademark of Compaq Computer
Corporation.
```

Appendix D: 180-Day Space

Warehouses	2030	**tpmC**	25,065.30	**tpmC/W**	12.35		
Table	**Rows**		**Data**	**Index**	**5% Space**	**8H Space**	**Total Space**
	should be	as counted	KB	KB	KB	KB	MB
Warehouse	2,050	2,050	223	48	14		0.28
District	20,500	20,500	2,280	64	117		2.40
Customer	61,500,000	61,500,000	44,727,280	2,872,032	2,379,966		48,807.89
History	61,500,000	61,500,000	3,416,736	0		668,422	3,989.41
Orders	61,500,000	61,500,000	1,885,064	1,041,224		572,474	3,416.76
New-order	18,450,000	18,450,000	291,704	808		41,000	325.70
Order-line	615,000,000	615,000,000	38,437,344	95,744		7,538,290	44,991.58
Stock	205,000,000	205,000,000	65,600,000	146,992	3,287,350		67,416.35
Item	100,000	100,000	9,528	88	481		9.86
Totals			154,370,159	4,157,000	5,667,927	8,820,186	168,960
File Groups	**Files**		**Sizes**	**In Meg**	**Needed**	**Overhead**	**Not Needed**
		kb	kb		in meg	in meg	
misc_fg	7		8,064,000	55,125	52,736	527	1,862
big_fg	7		17,203,200	117,600	116,224	1,162	214
	0		0	0	0	0	0
	0		0	0	0	0	0
	0		0	0	0	0	0
	0		0	0	0	0	0
Totals				172,725	168,960	1,690	2,075
Dynamic space	42,714	(MB) Sum of Data for Order, Order-Line and History (excluding free extents)					
Static space	119,322	(MB) Data + Index + 5% Space + Overhead - Dynamic space					
Free space	8,613	(MB) Total DB Size - Dynamic Space - Static Space - Not Needed					
Daily growth	8,356	(Dynamic space/W * 62.5)* tpmC					
Daily spread	(3,921)	Free space - 1.5 * Daily growth (zero if negative)					
180 day (MB)	1,623,440	Static space + 180 (daily growth + daily spread)					
180 day (GB)	**1,585.39**	Excludes OS, Paging and RDBMS Logs					
Log size (MB)	51999.99219	Total size of log file					
Log space before	18.095621						
Log space after	35.428162						
% Log used	17.3325	% of log file used during entire run					
Total N-O Txn	1838375	Total count of N-O transactions during entire run					
Log per N-O txn	5,141	Number of bytes per New-Order transaction					
8 Hour Log (GB)	57.60						
Mirrored=	**115.21**						
Confirgured disks							
Data disks =	# of Disks	Format'd size					
	18	3.99	72				
	234	8.68	2,030	252			
	0	17.37	0				
Totals	252		2,102				
Log disks =	# of Disks	Format'd size					
	0	3.99	0				
	14	8.68	122				
	0	17.37	0				
Totals	14		122				
Priced disks							
Data disks =	# of Disks	Format'd size					
	108	3.99	431				
	144	8.68	1,249				
	0	17.37	0				
Totals	252		1,680				
Log disks =	# of Disks	Format'd size					
	0	3.99	0				
	14	8.68	122				
	0	17.37	0				
Totals	14		122				
			115.21	used in log			
			6.31	left after log			
			1,686	space available for 180 day			

Appendix E:
Third Party Letters

JUN 07 1999 16:41 FR MICROSOFT RECP #1 425 936 7329 TO 912815148375 P.02/02

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

[illegible]
Fax 425 936 7329
http://www.microsoft.com/

Microsoft

June 7, 1999

Mr. Mike Engbrock
Compaq Computer Corporation
P.O. Box 692000
Houston, TX 77269-2000

Dear Mr. Engbrock:

Here is the information you requested regarding U.S. pricing for several Microsoft products, to be used in conjunction with TPC-C benchmark testing.

Microsoft SQL Server 7.0, Enterprise Edition (one server plus unlimited CALs)	$28,999
Microsoft Windows NT Server 4.0, Enterprise Edition (one server plus 25 CALs)	$3,999
Windows NT Server 4.0 (one server plus 5 CALs)	$809
Visual C++ Professional 6.0 (single copy)	$549
5-year maintenance for above software @ $2095/yr	$10,475

This quote is valid for the next 90 days.

If I can be of any further assistance, please contact me at 425-936-5301 or tomkr@microsoft.com.

Yours truly,

Thomas Kreyche
Product Manager
SQL Server Marketing

Microsoft Corporation is an equal opportunity employer.

** TOTAL PAGE.02 **

ALLSTAR SYSTEMS®

Item Description	Stock Number	Price Each	QTY	Total	Service		Total Price
ProLiant 7000-6/500-2S	112893-001	28,728.00	1	$28,728.00			$28,728.00
ProLiant Enterprise Server - Parts Exchange - 1 Year		$597	2		$1,194		$1,194.00
Onsite 5x8 4 Hr Service Upgrade from Compaq Standard Service		0.00	0		$23,821.06		$23,821.06
ProLiant Xeon 500 - 2MB Processor Kit	328814-B21	6,720.00	2	$13,440.00			$13,440.00
512MB EDO DIMM Memory Kit (50ns)	328582-B21	1,676.64	7	$11,736.48			$11,736.48
Compaq SMART Array Controller 4200	295636-B21	2,217.60	7	$15,523.20			$15,523.20
ProLiant Storage System -UE	348710-002	1,902.88	21	$39,960.48			$39,960.48
ProLiant Storage System - Parts Exchange - 1 Year			42		$10,710.00		$10,710.00
4.3-GB UW SCSI-3 10,000 RPM Drive	304858-B21	502.88	1	$502.88			$502.88
Compaq VGA V55 Color Monitor	308006-001	239.68	6	$1,438.08			$1,438.08
4/8GB DAT Drive	295353-B21	703.36	1	$703.36			$703.36
4.3 GB Hot-Plug SCSI-3 Drive - 10K	336356-B21	502.88	108	$54,311.04			$54,311.04
9.1 GB Hot-Plug SCSI-3 Drive - 10K	336357-B21	838.88	158	$132,543.04			$132,543.04
Netelligent 10/100 TX Controller	317600-B21	75.04	1	$75.04			$75.04
ProLiant 1600 6/450 - 64MB,100TX Nic, 4.3GB	313650-001	2,832.48	5	$14,162.40			$14,162.40
ProLiant Workgroup Server - Parts Exchange - 1 Year	FM-LOPRT-12	259.00	10		$2,590.00		$2,590.00
Onsite 5x8 4 Hr Service Upgrade from Compaq Standard Service		0.00	0		$2,445.63		$2,445.63
ProLiant 1600 6/450 Processor Option Kit	313618-B21	1,342.88	5	$6,714.40			$6,714.40
128-Megabyte SD DIMM Kit	313615-B21	327.04	20	$6,540.80			$6,540.80
4.3GB Wide Ultra2 SCSI Hard Drive	349525-B21	390.88	5	$1,954.40			$1,954.40
3Com Superstack II 24 Port 10/100 Switch		1,109.99	4	$4,440			$4,439.96
8-port 10BaseT hub	0	29.00	2816	$81,664			$81,664.00
MKS Toolkit for NT		499.00	1	$499.00			$499.00
Large Purchase and Cash discount				($37,299.38)	($3,668.46)		($40,967.84)
		Totals:		$377,637.18	$37,092.23		$414,729.41

Prices include a large volume and cash prepay discount. Prices may vary when items are purchased separately.
Service prices based upon a service contract for all items listed.
On-site 5 days/ 8 hour day / 4 hour response upgrade to Compaq standard 2nd day Service
The above quotation is valid for at least 60 days.
Compaq Parts Exchange contract in years 4 & 5 in lieu of parts warranty.