

TPC Benchmark™ C

Full Disclosure Report for

SIEMENS

Primergy 870-40

**Using Microsoft SQL Server 7.0
Enterprise Edition**

**and Microsoft Windows NT 4.0
Enterprise Edition**

March 18, 1999

Second Edition

Second Edition March 18, 1999

Siemens 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. We assume 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, we provide no warranty of the pricing information in this document.

Benchmark results are highly dependent upon workload, specific application requirements, 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. We do not warrant or represent that a user can or will achieve similar performance expressed in transactions per minute (tpmC) or normalized price/performance (\$/tpmC). No warranty of system performance or price/performance is expressed or implied in this report.

Copyright © 1999 Siemens AG 1999. All rights reserved.

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

Primergy 870 is a trademark of Siemens AG.

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

Pentium® II XEON is a registered trademark of Intel.

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

BEA and BEA TUXEDO are registered trademarks of BEA Systems, Inc.

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

Preface

The Transaction Processing Performance Council (TPC), of which Siemens AG is a member, is an organization of computer companies, dedicated to the development of objective, industry-wide performance metrics in the area of transaction processing. Siemens AG is involved in this effort, participating on the council and utilizing TPC benchmarks in performance evaluation.

The TPC Benchmark™ C Standard Specification was developed by the Transaction Processing Performance Council. This benchmark exercises the system components necessary to perform tasks associated with that class of on-line transaction processing (OLTP) environments emphasizing a mixture of read-only and update intensive transactions. This is a complex OLTP application environment exercising a breadth of system components associated by such environments characterized by:

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

This benchmark defines four on-line transactions and one deferred transaction, intended to emulate functions that are common to many OLTP applications. However, this benchmark does not reflect the entire range of OLTP requirements. The extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

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

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 subjected 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.

Summary

This report documents the TPC Benchmark™ C results achieved by the Siemens AG using Microsoft SQL Server 7.0 Enterprise Edition.

The TPC Benchmark™ C tests were run on a Primergy 870-40 system using the Windows NT 4.0 Enterprise Edition operating system.

The results, summarized below, show the number of TPC Benchmark™ C transactions per minute (tpmC) and the price per tpmC (\$/tpmC).

Software	Hardware	tpmC	\$/tpmC
Microsoft SQL Server 7.0 Enterprise Edition, Windows NT 4.0 Enterprise Edition	Siemens AG Primergy 870-40	23,570.33	23.41\$

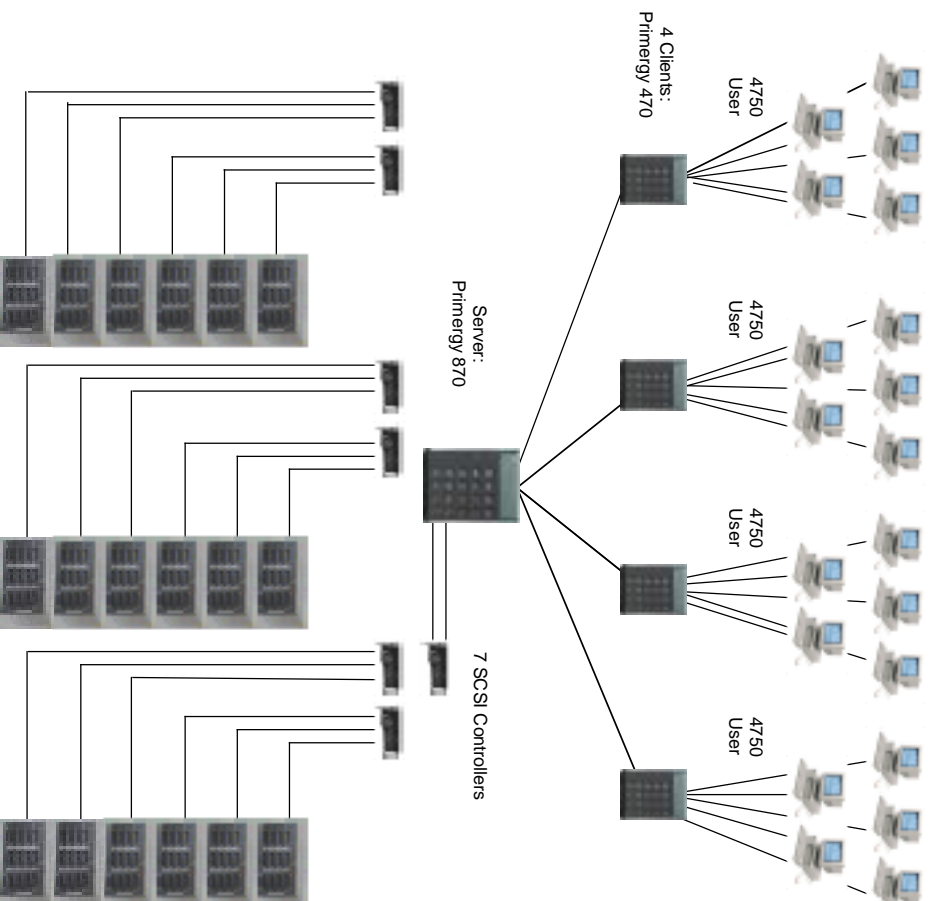
SIEMENS

Primergy 870-40 Client/Server

TPC-C REV 3.4
EXECUTIVE SUMMARY

Report Date: March 18, 1999

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
\$ 551,727	23,570.33 tpmC	\$23.41/tpmC	April 26, 1999
Processors	Database Manager	Operating-System	Other Software
4 Intel Pentium® III Xeon 500 MHz	Microsoft SQL Server 7.0 Enterprise Edition	Microsoft Windows NT 4.0 Enterprise Edition	Microsoft Internet Information Server BEA TUXEDO 6.4 CFS
			Number of Users
			19,000



18 Primergy S40

System Components	Qty/Srv.	1 Primergy 870-40	Qty/Client	4 Primergy 470
Processors	4	Intel Pentium® III Xeon 500 MHz 2 MB SLC	2	Intel Pentium® II 450 MHz 512 KB SLC
Memory	4	GB	384	MB
Disk Controller	7	SCSI Controllers	1	SCSI Controller
Disk Drives	193	9 GB	1	4 GB
	8	18 GB		
Total GB of Storage	1	1772 GB	1	4 GB

SIEMENS

Primergy 870-40

TPC-C REV 3.4
EXECUTIVE SUMMARY

Client/Server

Report Date: March 18, 1999

Description	Part Number	Third Party Brand	Unit Pricing	Qty.	Extended Price	5yr Maint. Price
Base System						
Pentium III Xeon Processor 500MHz/2MB						
2. Memory Board						
Memory 512MB EDO-DIMM	SNP-SY-K504V103-P		\$11,911	1	\$11,911	
DAT Drive DDS-3 12GB, SE-SCSI	SNP-SY-F1824E403-P		\$7,911	4	\$31,644	
Power Supply (add)	SNP-SY-F1825E1-P		\$800	1	\$800	
Keyboard	SNP-SY-F1549E543-P		\$2,311	8	\$18,488	
Country Pack	SNP-SY-F1730E1-P		\$1,156	1	\$1,156	
Monitor MCM 1510	SNP-SY-F1899E109-P		\$933	1	\$933	
	SNP-PS-F234E1-P		\$489	1	\$489	
	S26381-K271-V310		\$25	1	\$25	
	SNP-SY-F1699B213-P		\$36	1	\$36	
	S26361-K494-V150		\$191	1	\$191	
Server Hardware Subtotal						\$65,673

RAID-Controller:3 Chan.,32Mb, LVD	SNP-SY-F1960E32-P		\$2,133	7	\$14,931	
RAID-Controller:3 Chan.,32Mb, LVD (10% spares)	SNP-SY-F1960E32-P		\$2,133	2	\$4,266	
18GB/10K LVD, Hot Plug (incl. 10% spare)	SNP-SY-F1899L118-P		\$1,680	10	\$16,800	
APC Smart UPS 1400 (incl. 10% spare)	S88203 APC		\$587	3	\$1,761	
9GB/10K LVD-SCSI, Hot Plug	SNP-SY-F1899E109-P		\$933	192	\$179,136	
9GB/10K LVD-SCSI, Hot Plug (10% spares)	SNP-SY-F1899L109-P		\$949	20	\$18,980	
Primergy S40 ES, 1 channel stack	SNP-SY-K638V201-P		\$2,356	18	\$42,408	
SCSI Caddy UHD-HD LVD 5m	SNP-SY-F1947L50-P		\$156	18	\$2,808	
Storage Subtotal						\$290,890

Maint. Server + Storage						
Primergy 470, FS Pl, 450	SNP-SY-K482V714-A		\$2,289	4	\$9,156	\$23,296
Pl, 450MHz	SNP-SY-F1727E450-A		\$1,200	4	\$4,800	
Keyboard	S26381-K271-V310		\$25	4	\$100	
Country Pack International	SNP-SY-F1699B153-A		\$36	4	\$144,00	\$8,000
Memory 128 MB SDRAM 100MHz	SNP-SY-F1867E503-A		\$578	4	\$2,312	
Memory 256 MB SDRAM 100MHz	SNP-SY-F1867E504-A		\$1,289	4	\$5,156	
HD 4GB, SCSI-3, Hot Plug	S26361-F1724-E40		\$489	4	\$1,956	
Fast-Ether-Express-Pro/100+ Server (PCI)	SNP-SY-F1868E501-A		\$111	24	\$2,664	
Monitor MCM 1510	S26361-K494-V150		\$191	4	\$764	\$871
Client Hardware Subtotal						\$27,062

Microsoft Windows NT-Server, Enterprise Edition 4.0, Incl 25 CALS	Microsoft	2	\$3,999	1	\$3,999	
MS SQL-Server 7.0 Enterp Edition unlim. License	Microsoft	2	\$28,999	1	\$28,999	
Server Software Subtotal						\$32,998

Microsoft Windows NT-Server 4.0, incl. 5 CALS	Microsoft	2	\$809	4	\$3,236	
Tuxedo 6.4 Core Functionality Services for NT	BEA	4	\$3,000	4	\$12,000	\$9,000
Microsoft Visual C++ Professional 5.0	Microsoft	2	\$499	1	\$499	
Client Software Subtotal						\$15,735

Complex 8-port 10BaseT Hdb (8+1) ports (incl.109	Z85094 Complex	5	\$29.00	2618	\$75,922	
Kingston Switch 8 port 10/100 incl. 10% spare	732084 Kingston	5	\$605	3	\$1,815	
User Connectivity Subtotal						\$77,737
Total						\$500,085

Five-Year Cost of Ownership	\$551,727
tpmC	23570.33
\$/tpmC	\$23.41

1=Siemens, 2=Microsoft, 4=BEA Systems, 5=Software House

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 section 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: \$551,727
tpmC Rating: 23,570.33
\$/tpmC: 23.41

Note: The benchmark results and test methodology were audited by Francois Raab of Information Paradigm, assisted by Larry Fontana

Numerical Quantities Summary

MQTh, computed Maximum Qualified Throughput 23,570.33 tpmC
 % throughput difference, reported & reproducibility runs 0.22 %

Response Times (in seconds)	90th percentile	Average	Maximum
- New-Order	0.53	0.36	6.83
- Payment	0.94	0.49	38.46
- Order-Status	0.42	0.30	3.94
- Delivery (interactive portion)	0.32	0.30	0.85
- Delivery (deferred portion)	1.08	0.63	4.34
- Stock-Level	2.82	2.17	5.66
- Menu	0.32	0.20	0.93

Transaction Mix, in percent of total transactions

- New-Order	44.88 %
- Payment	43.04 %
- Order-Status	4.03 %
- Delivery	4.04 %
- Stock-Level	4.02 %

Emulation Delay (in seconds)

	Response Time	Menu
- New-Order	0.1	0.1
- Payment	0.1	0.1
- Order-Status	0.1	0.1
- Delivery (interactive)	0.1	0.1
- Stock-Level	0.1	0.1

Keying/Think Times (in seconds)

	Minimum	Average	Maximum
- New-Order	18.00/0.00	18.01/12.02	18.07/120.50
- Payment	3.00/0.00	3.01/12.04	3.06/120.50
- Order-Status	2.00/0.00	2.01/10.02	2.05/100.49
- Delivery (interactive)	2.00/0.00	2.01/ 5.03	2.05/ 50.50
- Stock-Level	2.00/0.00	2.01/ 5.06	2.05/ 50.50

Test Duration and Checkpointing

- Ramp-up time	71 minutes
- Measurement interval	30 minutes
- Number of checkpoints	1
- Checkpoint interval	30 minutes
- Transactions during measurement interval (all types)	1,575,620

Contents

PREFACE.....	3
SUMMARY	4
NUMERICAL QUANTITIES SUMMARY	7
CONTENTS.....	9
INTRODUCTION.....	11
<i>Software and Hardware Configuration</i>	11
<i>Full Disclosure</i>	11
<i>Report Format</i>	11
<i>Additional Copies</i>	12
1. GENERAL ITEMS	13
1.1 <i>Application Code</i>	13
1.2 <i>Benchmark Sponsor</i>	13
1.3 <i>Parameter Settings</i>	13
1.4 <i>Configuration Diagrams</i>	14
<i>SUT Configuration</i>	14
2. CLAUSE 1 RELATED ITEMS - LOGICAL DATABASE DESIGN.....	17
2.1 <i>Table Definitions</i>	17
2.2 <i>Physical Organization of Database</i>	17
2.3 <i>Insert and Delete Operations</i>	18
2.4 <i>Database Partitioning</i>	18
2.5 <i>Replication of Tables</i>	18
2.6 <i>Additional and/or Duplicated Attributes</i>	18
3. CLAUSE 2 RELATED ITEMS - TRANSACTION AND TERMINAL PROFILES	19
3.1 <i>Random Number Generator</i>	19
3.2 <i>Input/Output Screen Layout</i>	19
3.3 <i>Configured Terminal Features</i>	19
3.4 <i>Presentation Managers or Intelligent Terminals</i>	19
3.5 <i>Transaction Statistics</i>	20
3.6 <i>Queuing Mechanism</i>	20
4. CLAUSE 3 RELATED ITEMS - TRANSACTION AND SYSTEM PROPERTIES.....	21
4.1 <i>Atomicity</i>	21
4.2 <i>Consistency</i>	22
4.3 <i>Isolation</i>	22
4.4 <i>Durability</i>	23
5. CLAUSE 4 RELATED ITEMS - SCALING AND DATABASE POPULATION	25
5.1 <i>Initial Cardinality of Tables</i>	25
5.2 <i>Distribution of Tables and Log</i>	26
5.3 <i>Database Model, Interface, and Access Language</i>	26
5.4 <i>Database Partitions/Replications Mapping</i>	27
5.5 <i>180 day space Calculation</i>	27
6. CLAUSE 5 RELATED ITEMS - PERFORMANCE METRICS AND RESPONSE TIME	29

6.1 Measured pmc.....	29
6.2 Response Times.....	29
6.3 Keying and Think Times	29
6.4 Graphs.....	30
6.5 Steady State Determination	33
6.6 Work Performed.....	34
6.7 Reproducibility	35
6.8 Duration of Measurement.....	35
6.9 Regulation of Transaction Mix	35
6.10 Transaction Mix.....	35
6.11 Transaction Statistics.....	36
6.12 Checkpoint Statistics.....	36
7. CLAUSE 6 RELATED ITEMS - SUT, DRIVER, AND COMMUNICATION DEFINITION.....	37
7.1 RTE Inputs	37
7.2 Functionality and Performance of Emulated Components	37
7.3 Functional Diagrams of the Benchmarked and Proposed Configuration	37
7.4 Network Configurations of the Tested and Proposed Services	37
7.5 Network Bandwidth	38
7.6 Operator Intervention.....	38
8. CLAUSE 7 RELATED ITEMS - PRICING.....	39
8.1 System Pricing	39
8.2 Availability Dates	39
8.3 Throughput and Price/Performance	39
8.4 Country Specific Pricing.....	39
8.5 Usage Pricing.....	40
9. CLAUSE 8 RELATED ITEMS - AUDIT	41
APPENDIX A - APPLICATION SOURCE CODE.....	43
APPENDIX B - DATABASE DETAILS.....	161
APPENDIX C - TUNABLE PARAMETERS AND OPTIONS	177
APPENDIX D - PRICING DETAILS.....	377
180 Day Space Calculation.....	377
Price/Performance Spreadsheet.....	378
APPENDIX E - PRICE QUOTATIONS.....	379
APPENDIX F - ATTESTATION LETTER	384

Introduction

This is the Full Disclosure Report for the TPC Benchmark™ C running on the Siemens system Primergy 870-40. It meets the requirements of the TPC Benchmark™ C Standard Revision 3.4.

Software and Hardware Configuration

This report documents the compliance of the Siemens AG TPC Benchmark™ C tests using Microsoft SQL Server 7.0 Enterprise Edition Relational Database Management System.

The TPC Benchmark™ C tests were carried out on the Siemens system Primergy 870-40. Primergy 870-40 is a powerful Windows NT Enterprise Server that features an Intel Pentium® III Xeon 500 MHz processors manufactured by Intel.

The processor power may be upgraded by three further Intel Pentium® III Xeon 500 MHz processors with highspeed onboard local memory access. The main memory capacity of the Primergy 870-40 scaled from 256 MB up to 4 GB. The Primergy system family uses the Windows NT 4.0 Enterprise Edition operating system.

Full Disclosure

From Clause 8.1 of the TPC Benchmark™ C Standard Specification:

The intent of this disclosure is for a customer to be able to replicate the results of this benchmark given the appropriate documentation and products.

Siemens believes that this full disclosure report meets the stated intention. Siemens has strived to maintain the integrity of the Specification by adhering not only to the letter of the Specification, but also to its spirit.

Report Format

The format of this document follows Clause 8 of the TPC Benchmark™ C specification (TPC Benchmark™ C Standard Specification, Revision 3.4, Transaction Processing Performance Council) which describes the full disclosure report requirements for the test.

Each section of this report begins with the specification requirement printed in *italic type*. It is followed by plain type text that explains how the test complies with the requirement. Sections which require extensive listings reference appropriate appendices.

Report organization:

- General Items
- Clause 1 Related Items - Logical Database Design
- Clause 2 Related Items - Transaction and Terminal Profiles
- Clause 3 Related Items - Transaction and System Properties
- Clause 4 Related Items - Scaling and Database Population
- Clause 5 Related Items - Performance Metrics and Response Time
- Clause 6 Related Items - SUT, Driver, and Communication Definition
- Clause 7 Related Items - Pricing
- Clause 8 Related Items - Audit
- Appendix A - Application Source Code
- Appendix B - Database Details
- Appendix C - Tunable Parameters and Options
- Appendix D - Pricing Details
- Appendix E - Price Quotations
- Appendix F - Attestation Letter

Additional Copies

Additional copies of this report are available upon request from Siemens AG:

Siemens CS PS DS 5

Entry Server Performance Lab

Mr. Bathe

Heinz-Nixdorf-Ring 1

33106 Paderborn

Germany

1. General Items

1.1 Application Code

The application program (as defined in Clause 2.1.7) must be disclosed. This includes, but is not limited to, the code implementing the five transactions and the terminal input and output functions. [Clause 8.1.1.4]

The source code of the application program is provided in Appendix A - Application Source Code.

1.2 Benchmark Sponsor

A statement identifying the benchmark sponsor(s) and other participating companies must be provided. [Clause 8.1.1.5]

This benchmark was sponsored and executed by Siemens AG. The benchmark was developed and engineered by Siemens AG and Microsoft Corporation. Testing took place at Siemens NT-benchmark laboratories in Paderborn, Germany.

1.3 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 but not limited to:

- *Database tuning options.*
 - *Recovery/commit options.*
 - *Consistency/locking options.*
 - *Operating system and application configuration parameters.*
- [Clause 8.1.1.6]*

The significant parameters and system configuration files are provided in Appendix C - Tunable Parameters and Options.

1.4 Configuration Diagrams

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

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

[Clause 8.1.1.7]

SUT Configuration

The Primergy 870-40 server system included:

- 4 Intel Pentium® III Xeon 500 MHz with 2 MB Second Level Cache
- 4 GB memory
- 7 SCSI controllers
- 193 disks 9 GB
- 8 disks 18 GB
- 1 LAN

The Primergy 470 client system included:

- 2 Intel Pentium® II 450 MHz with 512 KB Second Level Cache
- 384 MB memory
- 1 SCSI controller
- 1 disk 4 GB
- 6 LAN

The benchmarked and priced system configurations are shown in Figure 1 and Figure 2 in accordance with Clause 8.1.1.7.

FIGURE 1: BENCHMARK SYSTEM CONFIGURATION PRIMERGY 870-40

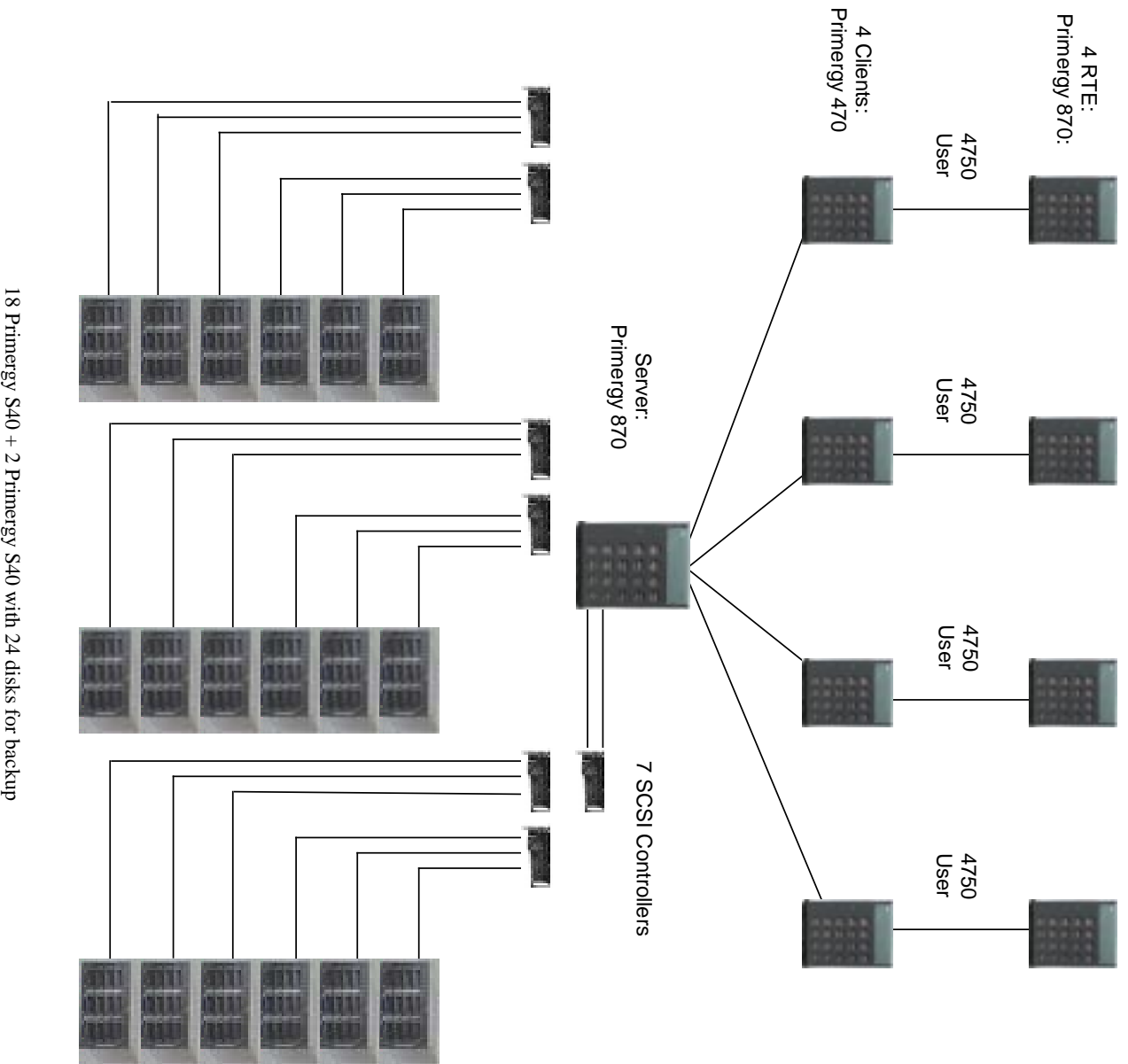
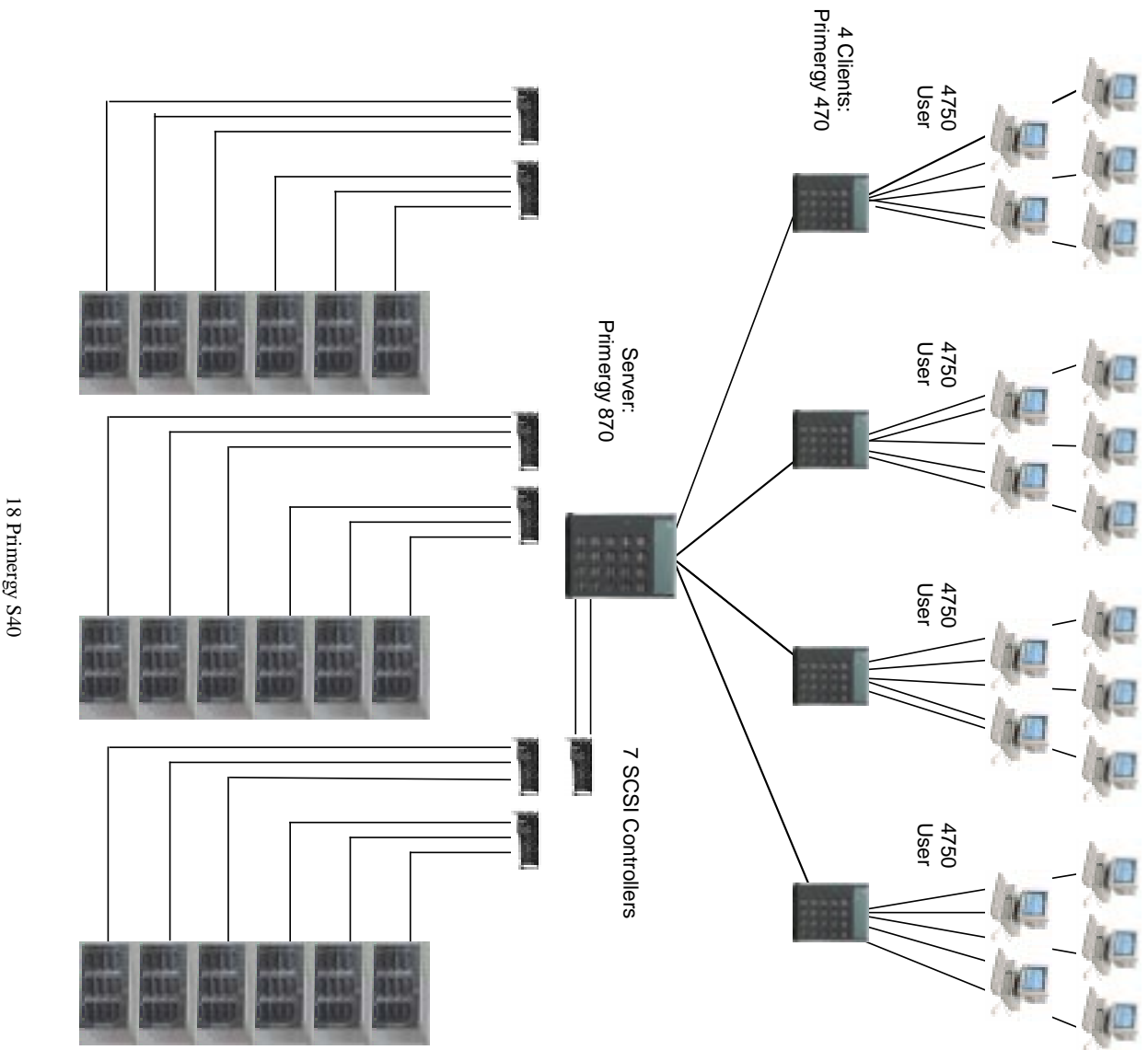


FIGURE 2: PRICED SYSTEM CONFIGURATION PRIMERGY 870-40



2. Clause 1 Related Items - Logical Database Design

2.1 Table Definitions

Listings must be provided for all table definition statements and all other statements used to set-up the database. [Clause 8.1.2.1]

The programs that defined, created, and populated the Microsoft SQL Server 7.0 Enterprise Edition database for this TPC benchmark™ C are listed in Appendix B - Database Details.

2.2 Physical Organization of Database

The physical organization of tables and indices, within the database, must be disclosed. [Clause 8.1.2.2]

FIGURE 1: PHYSICAL ORGANIZATION OF THE DATABASE

Disk #	Controller	Disktype	RAID Configuration	Drive Letter	Size MB	Filegroup or Filesystem
0,1	Symbios (onboard)	9 GB	-	WINNT40EE C: SQL D:	2039 6613	NTFS NTFS
2	DAC1164P #1	8 x 18 GB	RAID 1	L:	50000	log
3	DAC1164P #2	32 x 9 GB	RAID0	H: O:	18000 9600	cs1 misc1
4	DAC1164P #3	32 x 9 GB	RAID0	I: P:	18000 9600	cs2 misc2
5	DAC1164P #4	32 x 9 GB	RAID0	J: Q:	18000 9600	cs3 misc3
6	DAC1164P #5	32 x 9 GB	RAID0	K: R:	18000 9600	cs4 misc4
7	DAC1164P #6	32 x 9 GB	RAID0	M: S:	18000 9600	cs5 misc5
8	DAC1164P #7	32 x 9 GB	RAID0	N: T:	18000 9600	cs6 misc6
9	DAC1164P #8	12 x 9 GB	RAID0	BACKUP 1 X:	104195	NTFS
10	DAC1164P #8	12 x 9 GB	RAID0	BACKUP 2 Y:	104195	NTFS

All controllers were configured with write cache disabled. Write cache was enabled on the log drives and disabled on the data drives.

Space was allocated to Microsoft SQL Server 7.0 Enterprise Edition on SUT disks according to the data in section 5.2. The size of the datafile on each disk drive was calculated to provide even distribution on load across the disk drives. The NT Disk Administrator was used to create raw devices for data/log and NTFS partitions for dump devices. For further information see Appendix B (Disk Usage) and Figure 1 in 5.2 (Distribution of Tables and Log). No attempt

was made to alter the default physical organization of the database tables and indices chosen by Microsoft SQL Server 7.0 Enterprise Edition.

2.3 Insert and Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restriction in the SUT database implementation that precludes inserts beyond the limits defined in Clause 1.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the maximum key value for these new rows. [Clause 8.1.2.3]

There were no restrictions on insert and delete operations to any tables.

2.4 Database Partitioning

While there are a few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC benchmark™ C (see Clause 1.6), any such partitioning must be disclosed. [Clause 8.1.2.4]

There was no partitioning used in this implementation.

2.5 Replication of Tables

Replication of tables, if used, must be disclosed (see Clause 1.4.6). [Clause 8.1.2.5]

Replication of tables was not used in this implementation.

2.6 Additional and/or Duplicated Attributes

Additional and/or duplicated attributes in any table must be disclosed along with a statement on the impact on performance (see Clause 1.4.7). [Clause 8.1.2.6]

No additional and/or duplicated attributes were used.

3. Clause 2 Related Items - Transaction and Terminal Profiles

3.1 Random Number Generator

The method of verification for the random number generation must be described. [Clause 8.1.3.1]

The seeds and offsets for the random number generator were collected and verified to be different for each driver. Samples of the generated numbers were selected and verified to have no discernible patterns.

3.2 Input/Output Screen Layout

The actual layouts of the terminal input/output screens must be disclosed. [Clause 8.1.3.2]

The screen layout corresponded exactly to those of the TPC-C Standard Specification (specified in Clause 2.4.3, 2.5.3, 2.6.3, 2.7.3, and 2.8.3).

3.3 Configured Terminal Features

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). [Clause 8.1.3.3]

The Primergy 470 is commercially available. All of the requirements in clause 2.2.2.4, are supported. This was verified by manually exercising each specification on a Primergy 470.

3.4 Presentation Managers or Intelligent Terminals

Any usage of presentation managers or intelligent terminals must be explained. [Clause 8.1.3.4]

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 application is listed in Appendix A - Application Source Code.

3.5 Transaction Statistics

*The numerical quantities which are required are listed in the following table.
[Clause 8.1.3.5 to 8.1.3.11]*

	Statistics	Percentage
New-Order	Home order-lines	99.00%
	Remote order-lines	1.00%
	Rolled back transactions	0.99%
	Average items per order	10.00
Payment	Home transactions	85.03%
	Remote transactions	14.97%
	Non-primary key access	59.96%
Order-Status	Non-primary key access	59.88
Delivery	Skipped transactions	0
Transaction Mix	New-Order	44.88 %
	Payment	43.04 %
	Order-Status	4.03 %
	Delivery	4.04 %
	Stock-Level	4.02 %

3.6 Queuing Mechanism

The queuing mechanism used to defer the execution of the Delivery transaction must be disclosed. [Clause 8.1.12]

The client application processes submitted delivery transactions to asynchronous TUXEDO servers running on the client machines. There were multiple delivery servers with single execution threads running on each client machine. These delivery servers were responsible for processing deliveries queued to TUXEDO and submitting them to the database server.

The source code is listed in Appendix A - Application Source Code.

4. Clause 3 Related Items - Transaction and System Properties

ACID Tests

The results of the ACID tests must 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. [Clause 8.1.4.1]

All ACID tests were performed successfully. The following sections describe the requirements of each of the tests as described in Clause 3 and the approach used to satisfy them.

All ACID tests were performed on the Primarygy 870-40 system using the fully scaled database, except for the test of durable media failure. The durability test was performed on a database scaled to 15 warehouses. This test would also pass on a fully scaled database.

4.1 Atomicity

The system under test must guarantee that database transactions are atomic; the system will either perform all individual operations on the data, or will assure that no partially-completed operations leave any effects on the data [Clause 3.2.1]

Commit Transaction

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number as specified in Clause 2.5.1.2) and verify that the records in the CUSTOMER, DISTRICT, and WAREHOUSE tables have been changed appropriately. [Clause 3.2.2.1]

The following steps demonstrated atomicity for completed (COMMIT) transactions:

- A row was randomly selected from the warehouse, district and customer table.
- the current balance was noted.
- A payment transaction was executed with the above identifiers and a known amount.
- The transaction was committed.
- It was verified, that the rows contain the correct updated balances.

Rollback Transaction

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number as specified in Clause 2.5.1.2) and substitute a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the records in the CUSTOMER, DISTRICT, and WAREHOUSE tables have NOT been changed. [Clause 3.2.2.2]

The following steps demonstrated atomicity for aborted (ROLLBACK) transactions:

- A row was randomly selected from the warehouse, district and customer table.
- the current balance was noted.
- A payment transaction was executed with the above identifiers and a known amount.
- The transaction was rolled back.
- It was verified, that the rows contain the original balances.

4.2 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. [Clause 3.3.1]

Consistency conditions 1 - 4 were tested by issuing queries to the database. The results of the queries verified that the database was consistent for all these tests. The tests were performed before and after the performance run on the same database that was used for the benchmark.

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

We ran all of the seven isolation tests as described in clause 3.4.2.1 to 3.4.2.7 and additionally the two phantom protection tests. The tests were executed using shell scripts to issue queries to the database. The results of the queries verified that the required isolation had been met.

4.4 Durability

The tested system must guarantee durability: the ability to preserve the effects of committed transactions and insure database consistency after recovery from any one of the failures listed in Clause 3.5.3. [Clause 3.5]

List of single failures:

- 1 *Permanent irrecoverable failure of any single durable medium containing TPC-C database tables or recovery log data.*
- 2 *Instantaneous interruption (system crash / system hang) in processing which requires system reboot to recover.*
- 3 *Failure of all or part of memory (loss of contents).*

[Clause 3.5.3]

The intent of these tests is to demonstrate that all transactions whose output messages have been received at the terminal or RTE have in fact been committed in spite of any single failure from the list in Clause 3.5.3 and that all consistency conditions are still met after the database is recovered.

It is required that the system crash test(s) and the loss of memory test(s) described in Clause 3.5.3.2 and 3.5.3.3 be performed under full terminal load and a fully scaled database. The durable media failure test(s) described in Clause 3.5.3.1 may be performed on a subset of the SUT configuration and database. For the SUT subset, all multiple hardware components, such as processors and disk / controllers in the full SUT configuration, must be represented by the greater of 10% of the configuration or two of each of the multiple hardware components. The database must be scaled to at least 10% of the fully scaled database, with a minimum of two warehouses. ... Furthermore, the standard driving mechanism must be used in this test. The test sponsor must state that to the best of their knowledge, a fully scaled test would also pass all durability tests. [Clause 3.5.4]

The failure of all or part of memory test and the system crash test were combined with the loss of log disk and performed under full load and by using a fully scaled database.

The full hardware configuration of the SUT (in accordance with Clause 3.5.4) and the same test procedure was used during all durability tests, except the test for loss of data.

- The current count of the total number of orders was determined by summing up the D_NEXT_O_ID fields of all rows in the DISTRICT table before the test.
- 5 min after starting the measurement we pulled off one of the log disks. As we use hardware-mirrored diskpairs with the SCSI-controller, execution continued.
- After additional 5 min we powered of the server to emulate the loss of memory. After server system reboot, SQL-Server starts with recovering the database tpc. After completion, we computed the sum of D_NEXT_O_ID from district. Client and RTE systems were interrupted and evaluation started on the RTE. The difference of all D_NEXT_O_ID between RTE an server was in the permitted scope.

The durable media failure test for loss of data disk was performed with 32 of the 192 data disks and a database scaled to 15 warehouses under the load of 150 users. We used one RTE and one client system. To the best of the test sponsor's

- knowledge, a fully loaded and fully scaled database would also pass this durability test.
- The database was backed up.
 - The current count of the total number of orders was determined by summing up the D_NEXT_O_ID fields of all rows in the DISTRICT table before the test.
 - 5 min after starting the measurement, we pulled of one of the data disks.
 - SQL-Server recognized the loss of a device. We shut down SQL-Server.
 - We replaced the disk and made it online.
 - We deleted the database partition to make it inaccessible.
 - To dump the transaction log, we restarted the SQL-Server.
 - We removed the database with dropdevice.
 - We shut down SQL-Server.
 - We restarted SQL-Server, no tpc-c database and none of its devices were present. We recreated the database, loaded dump and load transaction log
 - After completion, we computed the sum of D_NEXT_O_ID from district.
 - Client and RTE systems were interrupted and evaluation started on the RTE. The difference of all D_NEXT_O_ID between RTE an server was in the permitted scope.

5. Clause 4 Related Items - Scaling and Database Population

5.1 Initial Cardinality of Tables

The cardinality (e.g., the number of rows) of each table, as it existed at the start of the benchmark run (see Clause 4.2), must be disclosed. If the database was over-scaled and inactive rows of the WAREHOUSE table were deleted (see Clause 4.2.2), the cardinality of the WAREHOUSE table as initially configured and the number of rows deleted must be disclosed. [Clause 8.1.5.1]

The database for the Primergy 870-40 system was scaled for 1900 warehouses. In accordance with Clause 4.2, the following number of records were loaded in the specified tables:

Table	Number of Records
Warehouse	1900
District	19,000
Customer	57,000,000
History	57,000,000
Order	57,000,000
New-Order	17,100,000
Order-Line	570,001,928
Stock	190,000,000
Item	100,000
Deleted Warehouses	0

The following constant values were used during the database build and benchmark test for the NURand function:

Constant C	Value
C_LAST (build)	123
C_LAST (run)	233

5.2 *The distribution of tables and logs across all media must be explicitly depicted for the tested and priced systems. [Clause 8.1.5.2]*

Distribution of Tables and Log

FIGURE 1: LOGICAL ORGANIZATION OF THE DATABASE

	device	raw size	use
D:	MSSQL70_tpc_root	10 MB	root
L:	MSSQL70_tpc_log	45,000 MB	Log
O:	MSSQL70_misc1	9,500 MB	Warehouse, District, Item, New Order, History, Order, Orderline
P:	MSSQL70_misc2	9,500 MB	Warehouse, District, Item, New Order, History, Order, Orderline
Q:	MSSQL70_misc3	9,500 MB	Warehouse, District, Item, New Order, History, Order, Orderline
R:	MSSQL70_misc4	9,500 MB	Warehouse, District, Item, New Order, History, Order, Orderline
S:	MSSQL70_misc5	9,500 MB	Warehouse, District, Item, New Order, History, Order, Orderline
T:	MSSQL70_misc6	9,500 MB	Warehouse, District, Item, New Order, History, Order, Orderline
H:	MSSQL70_cs1	17,950 MB	Stock, Customer
I:	MSSQL70_cs2	17,950 MB	Stock, Customer
J:	MSSQL70_cs3	17,950 MB	Stock, Customer
K:	MSSQL70_cs4	17,950 MB	Stock, Customer
M:	MSSQL70_cs5	17,950 MB	Stock, Customer
N:	MSSQL70_cs6	17,950 MB	Stock, Customer

5.3 *A statement must be provided that describes:*

Database Model, Interface, and Access Language

1. *The data model implemented by the DBMS used (e.g., relational, network, hierarchical)*

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

[Clause 8.1.5.3]

Microsoft SQL Server 7.0 Enterprise Edition is a Relational DataBase Management System. The interface used was Microsoft SQL Server 7.0 Enterprise Edition stored procedures accessed with Remote Procedure Calls embedded in C code.

5.4 Database Partitions/Replications Mapping

*The mapping of database partitions/replications must be explicitly described.
[Clause 8.1.5.4]*

There was no partitioning and/or replication used in this implementation.

5.5 180 day space Calculation

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 (see Clause 4.2.3). [Clause 8.1.5.5]

Calculations of space requirements in the priced configurations for the 180-day period are provided in Appendix D - Pricing Details.

6. Clause 5 Related Items - Performance Metrics and Response Time

6.1 Measured tpmC

Measured tpmC must be reported. [Clause 8.1.6.1]

During the 30 minutes measurement period on the Primergy 870-40 the throughput measured was 23,570.33 tpmC.

6.2 Response Times

Ninetieth percentile, maximum and average response times must be reported for all transaction types as well as for the Menu response time. [Clause 8.1.6.2]

Type	Average	Maximum	90 Percentile
New-Order	0.36	6.83	0.53
Payment	0.49	38.46	0.94
Order-Status	0.30	3.94	0.42
Interactive Delivery	0.30	0.85	0.32
Deferred Delivery	0.63	4.34	1.08
Stock-Level	2.17	5.66	2.82
Menu	0.20	0.93	0.32

6.3 Keying and Think Times

The minimum, the average, and the maximum keying and think times must be reported for each transaction type. [Clause 8.1.6.3]

Keying Times			
Type	Average	Maximum	Minimum
New-Order	18.01	18.07	18.00
Payment	3.01	3.06	3.00
Order-Status	2.01	2.05	2.00
Delivery	2.01	2.05	2.00
Stock-Level	2.01	2.05	2.00

Think Times			
Type	Average	Maximum	Minimum
New-Order	12.02	120.50	0.00
Payment	12.04	120.50	0.00
Order-Status	10.02	100.49	0.00
Delivery	5.03	50.50	0.00
Stock-Level	5.06	50.50	0.00

6.4 Graphs

Response Time frequency distribution curves (see Clause 5.6.1) must be reported for each transaction type. [Clause 8.1.6.4]

The performance curve for response times versus throughput (see Clause 5.6.2) must be reported for the New-Order transaction. [Clause 8.1.6.5]

Think Time frequency distribution curves (see Clause 5.6.3) must be reported for each transaction type. [Clause 8.1.6.6]

Keying Time frequency distribution curves (see Clause 5.6.4) must be reported for each transaction type. [Clause 8.1.6.7]

A graph of throughput versus elapsed time (see Clause 5.6.5) must be reported for the New-Order transaction. [Clause 8.1.6.8]

FIGURE 1: NEW-ORDER RESPONSE TIME DISTRIBUTION

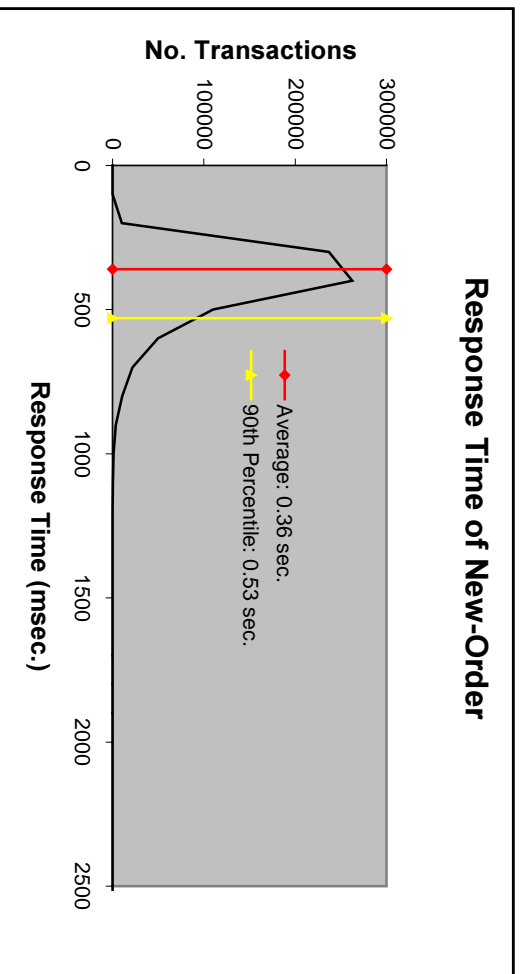


FIGURE 2: PAYMENT RESPONSE TIME DISTRIBUTION

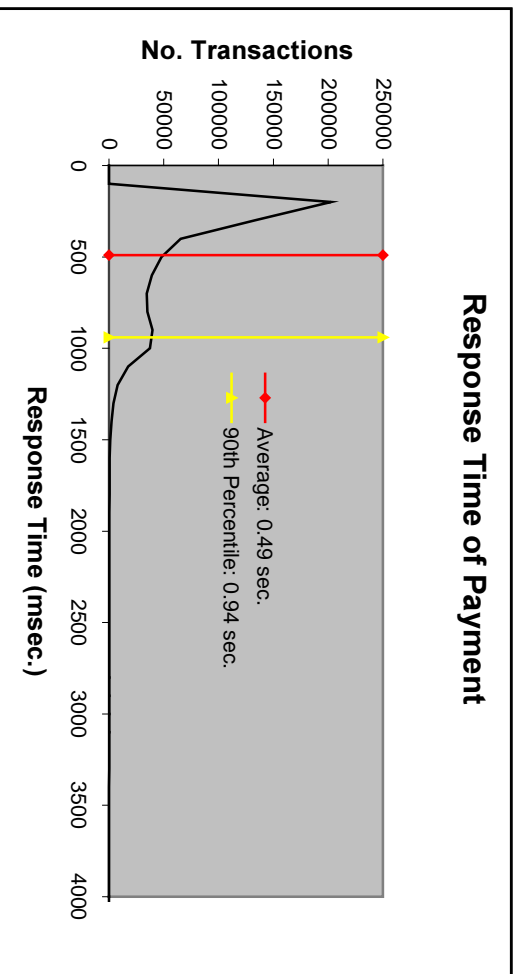


FIGURE 3: ORDER-STATUS RESPONSE TIME DISTRIBUTION

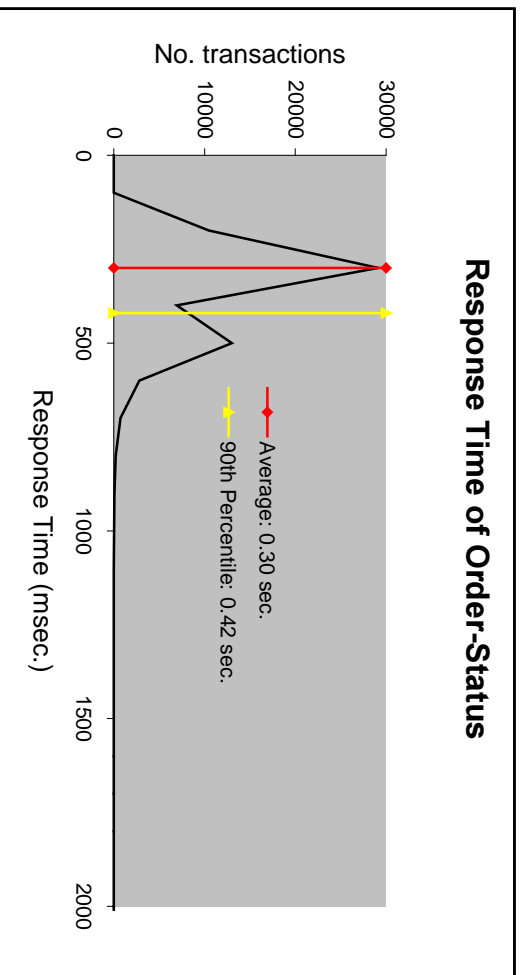


FIGURE 4: DELIVERY RESPONSE TIME DISTRIBUTION

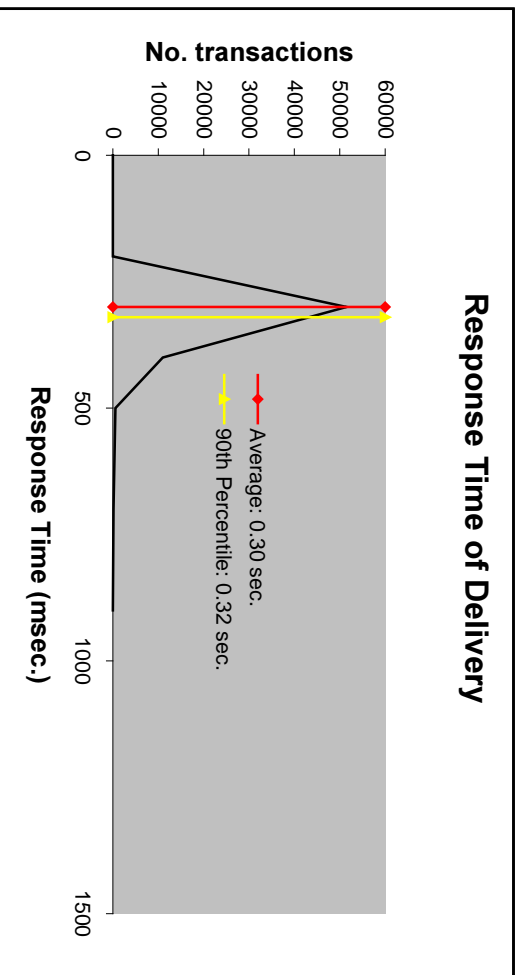


FIGURE 5: STOCK-LEVEL RESPONSE TIME DISTRIBUTION

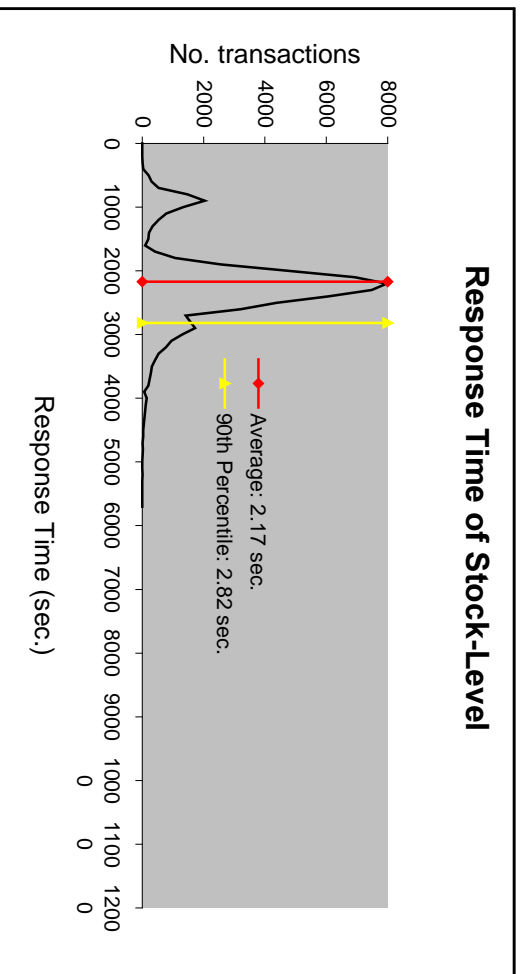


FIGURE 1: RESPONSE TIME VERSUS THROUGHPUT

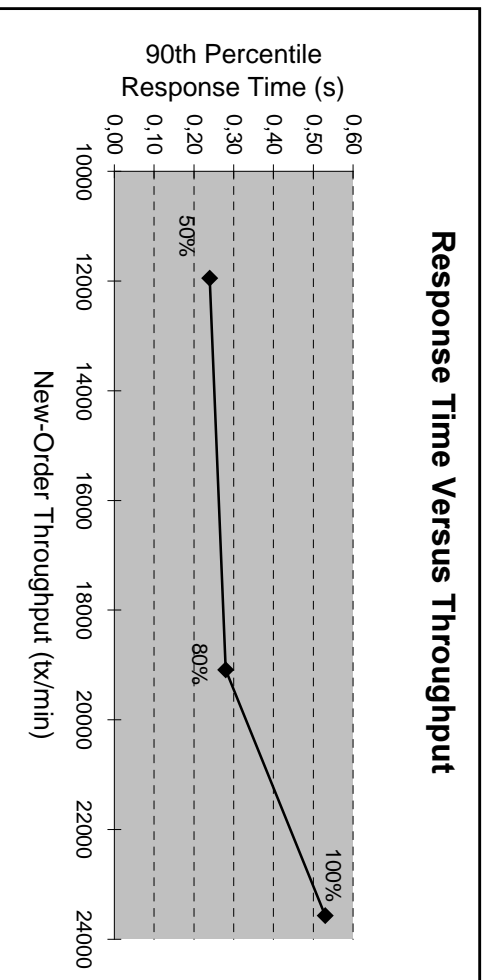


FIGURE 1: NEW-ORDER THINK TIME DISTRIBUTION

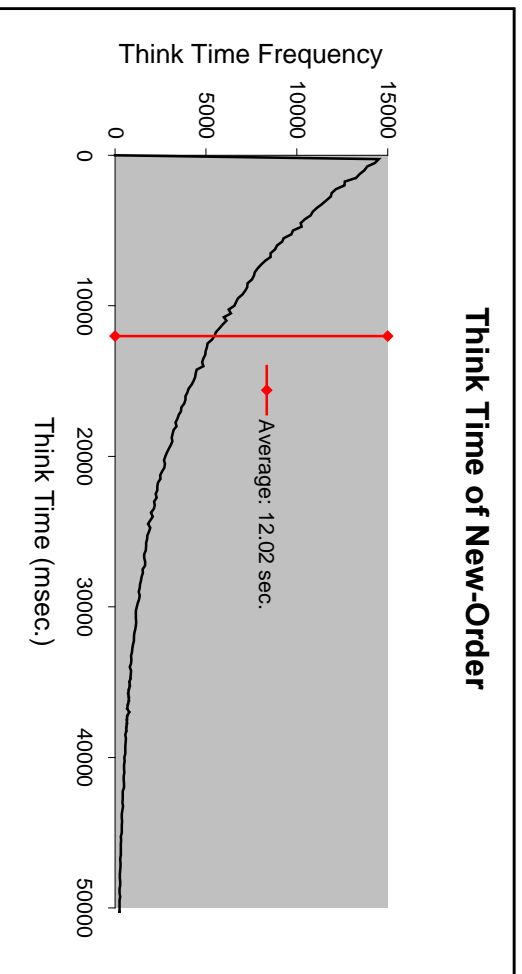
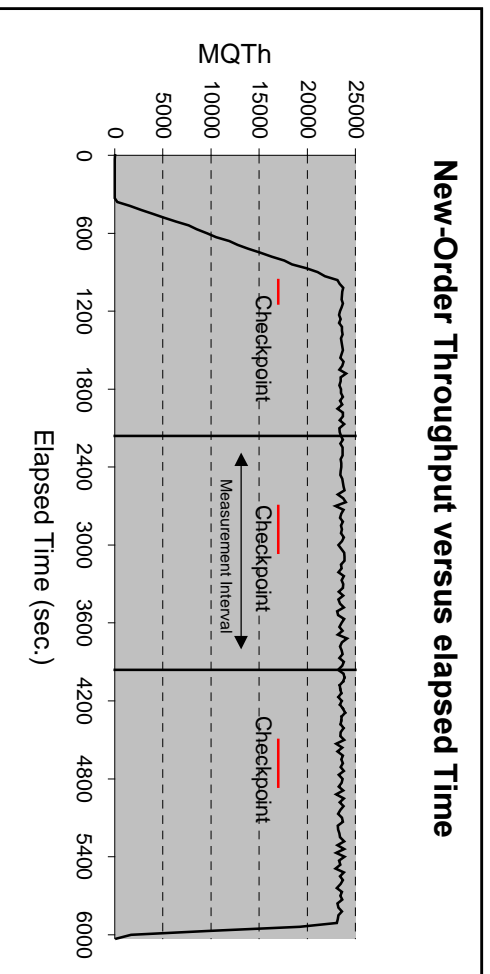


FIGURE 1: THROUGHPUT VERSUS ELAPSED TIME



**6.5
Steady State
Determination**

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval (see Clause 5.5) must be described. [Clause 8.1.6.9]

In all test runs, steady state was achieved before the measurement period began. Steady state was determined to occur based on a visual inspection of tpnC versus time (see graph in section 6.4).

The graph in section 6.4 illustrates that the measurement period was within the steady state period for the run. One checkpoint occurred during the measurement period.

6.6 Work Performed

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. [Clause 8.1.6.10]

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 Internet Information Server running on the client machines through Ethernet LANs. Internet Information Server handled all screen I/O as well as all requests to the database on the server. Internet Information Server communicated with the database server over TUXEDO which was used as transaction monitor.

All database operations like update, select, delete and insert are performed by one of the TPC-C back end programs. The TPC-C backend program commits the transaction after all the corresponding operations are done.

Modified database buffers are migrated to disk a least-recently-used basis independent of transaction commits. In addition, every block modification is protected by log records. Asynchronously the log buffers are flushed to a log file on disk either when the transaction is committed or when the log buffer's fill state reaches it's limit. The log buffer's always flushed by a commit before it become full.

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. By setting the trace flag #3502, SQL Server logged the checkpoint beginning and ending time in the ERRORLOG file. The script included a wait time between each checkpoint equal to the measurement interval which was 30 minutes. The checkpoint script was started manually after the RTE had all users logged in and sending transactions.

At each checkpoint, Microsoft SQL Server wrote to disk all memory pages that had been updated but not yet physically written to disk. Upon completion of the checkpoint, Microsoft SQL Server wrote a special record to the recovery log to indicate that all disk operations had been satisfied to this point.

6.7 Reproducibility

A description of the method used to determine the reproducibility of the measurement results must be reported. [Clause 8.1.6.11]

The Primergy 870-40 system test was run twice to ensure the reproducibility of the results. The reproducibility test run under exactly the same conditions as the reported test. All tests conform to the TPC rules.

The tpmC result from the reproducibility test was within 0.22% of the reported tpmC.

In the following, both results are shown to document the reproducibility:

	tpmC
reported test	23,570.33
reproducibility test	23,518.63

6.8 Duration of Measurement

A statement of the duration of the measurement interval for the reported Maximum Qualified Throughput (tpmC) must be included. [Clause 8.1.6.12]

The measurement interval of the Primergy 870-40 system test was 30 minutes. This measurement interval corresponds to the amount of time from the beginning of one checkpoint to the beginning of the next (which, actually, is less than the amount of time it takes to fill a log file).

6.9 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. [Clause 8.1.6.13]

The transaction mix was regulated by weighted distribution. The chosen weights meet the required minimum percentages of the mix which are described in Clause 5.2.3 of the Standard Specifications. No adjustments were made by the RTE.

6.10 Transaction Mix

The percentage of the total mix for each transaction type must be disclosed. [Clause 8.1.6.14]

	Percentage
New-Order	44.88 %
Payment	43.04 %
Order-Status	4.03 %
Delivery	4.04 %
Stock-Level	4.02 %

6.11 Transaction Statistics

The percentage of New-Order transactions rolled back as a result of invalid item number must be disclosed. [Clause 8.1.6.15]

The average number of order-lines entered per New-Order transaction must be disclosed. [Clause 8.1.6.16]

The percentage of remote order-lines entered per New-Order transaction must be disclosed. [Clause 8.1.6.17]

The percentage of remote Payment transactions must be disclosed. [Clause 8.1.6.18]

The percentage of customer selections by customer last name in the Payment and Order-Status transactions must be disclosed. [Clause 8.1.6.19]

The percentage of Delivery transactions skipped due to there being fewer than necessary orders in the New-Order table must be disclosed. [Clause 8.1.6.20]

The numerical quantities which are required in Clause 8.1.6.15 to 8.1.6.20 are already listed in a table above (see section 3.5).

6.12 Checkpoint Statistics

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. [Clause 8.1.6.21]

There was one checkpoint before and one during the measurement interval. The second checkpoint occurred 9 minutes after the start of the measurement interval. The checkpoint interval was 1800 seconds. The duration of the checkpoint during measurement was 6 minutes.

7. Clause 6 Related Items - SUT, Driver, and Communication Definition

7.1 RTE Inputs

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. [Clause 8.1.7.1]

Microsoft Benchcraft was used as the RTE to emulate the terminals. Its input parameters are shown in Appendix C - Tunable Parameters and Options.

We used TUXEDO 6.4 CFS to simulate terminal users, generate random data, record response times and statistical data. Its input parameters are shown in Appendix C - Tunable Parameters and Options.

7.2 Functionality and Performance of Emulated Components

It must be demonstrated that the functionality and performance of the components being emulated in the Driver System are equivalent to that of the priced system. The results of the test described in Clause 6.6.3.4 must be disclosed. [Clause 8.1.7.2]

The Driver System consisted of a Primergy 870-40. This driver was attached to the client machine through an Ethernet LAN. Since this is exactly the same connectivity as configured in the priced system, no component was emulated. Therefore, the test described in Clause 6.6.3.4 was not required.

7.3 Functional Diagrams of the Benchmarked and Proposed Configuration

A complete functional diagram of both the benchmark configuration and the configuration of the proposed (target) system must be disclosed. A detailed list of all software and hardware functionality being performed on the Driver System, and its interface to the SUT must be disclosed (see Clause 6.6.3.6). [Clause 8.1.7.3]

Figure 1 and Figure 2 in section 1.4 show the functional diagrams of the benchmark configuration and the priced configuration.

7.4 Network Configurations of the Tested and Proposed Services

The network configurations of both the tested services and the proposed (target) services which are being represented and a thorough explanation of exactly which parts of the proposed configuration are being replaced with the Driver System must be disclosed (see Clause 6.6.4). [Clause 8.1.7.4]

Figure 1 and Figure 2 in section 1.4 show the network setup of both configurations. The driver replaces the workstations.

In the tested configuration one standard ethernet LAN segment was used to connect the server with the clients and 5 standard ethernet LAN segments were used to connect each of the 4 clients with the 4 RTE systems.

In the priced configuration 20 standard ethernet LAN segments were used to connect 19,000 workstations. Each client has 4750 users connected with 5 ethernet segments.

7.5 Network Bandwidth

The bandwidth of the network(s) used in the tested / priced configuration must be disclosed. [Clause 8.1.7.5]

The Ethernet used in the local area network (LAN) between the emulated user system and the front-end system complies with the IEEE 802.3 standard. Its bandwidth is 10 Mbps. Between front-end and SUT the bandwidth is 100 Mbps.

7.6 Operator Intervention

If the configuration requires operator intervention (see Clause 6.6.6), the mechanism and the frequency of this intervention must be disclosed. [Clause 8.1.7.6]

The Primergy 870-40 requires no operator intervention to sustain the reported throughput.

8. Clause 7 Related Items - Pricing

8.1 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 date. If package-pricing is used, vendor part number of the package and a description uniquely identifying each of the components of the package must be disclosed. Pricing source(s) and effective date(s) of price(s) must also be reported. [Clause 8.1.8.1]

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. [Clause 8.1.8.2]

The details of the hardware and software are reported in the summary in front of this report. The spreadsheet used to determine the 5-year price and the spreadsheet used to describe the priced configuration can be found in Appendix D - Pricing Details.

8.2 Availability Dates

The committed delivery date for general availability (availability date) of products used in the price calculations must be reported. When the priced system includes products with different availability dates, the reported availability date for the priced system must be the date at which all components are committed to be available. [Clause 8.1.8.3]

All hardware and software components used in the price calculations of the Primergy 870-40 system will be generally available from Siemens AG as of April 26, 1999.

8.3 Throughput and Price/Performance

A statement of the measured tpmC, as well as the respective calculations for 5-year pricing, price/performance (price/tpmC), and the availability date must be included. [Clause 8.1.8.4]

Primergy 870-40 system was measured at 23,570.33 tpmC with Microsoft SQL Server 7.0 Enterprise Edition with a 5-year system price of \$551,727. The respective price/performance for the Primergy 870-40 is \$23.41/tpmC. The priced Primergy 870-40 will be available as of April 26, 1999.

8.4 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 [Clause 8.1.8.5]

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

8.5

Usage Pricing

For any usage pricing, the sponsor must disclose:

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

[Clause 8.1.8.6]

The component pricing based on usage is shown below:

- One Microsoft Windows NT Server 4.0 license (includes 5 client access licenses)
- One Microsoft Windows NT Server, Enterprise Edition 4.0 license (includes 25 client access licenses)
- One Microsoft SQL Server, Enterprise Edition 7.0 license (includes unlimited user license)
- One Microsoft SQL Workstation (includes programmers toolkit)
- Microsoft Visual C++ 32-bit edition

9. Clause 8 Related Items - Audit

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.

A review of the pricing model is required to ensure that all components required are priced (see Clause 9.2.8). The auditor is not required to review the final Full Disclosure Report or the final pricing prior to issuing the attestation letter. [Clause 8.1.9]

The benchmark test of the Primavera 870-40 system with Microsoft SQL Server 7.0 Enterprise Edition was independently audited by:

Francois Raab, a TPC certified auditor with Information Paradigm, Inc. of Colorado Springs, CO. , assisted by Larry Fontana
The attestation letter is included in Appendix F.

Requests for this TPC-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

SIEMENS CS PS DS 5
Entry Server Performance Lab
Mr. Bathé
Heinz-Nixdorf-Ring 1
33106 Paderborn
Germany

Appendix A - Application Source Code

TUXEDO

```
/*      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 ocured reading delivery log file
#define ERR_INSUFFICIENT_MEMORY 1002
                        //insufficient 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 void 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 delirtp 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      iTotalsBuckets;
    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 if 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 too 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        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");
};

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;
}

/* 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;

#ifdef LOCAL_ALLOC
DELIVERY_DATA DeliveryData;
#else
TUX_DATA TuxData;
#endif
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
*             int      -1     Failure
*
*
* COMMENTS:   None
*
*/

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

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

#ifdef LOCAL_ALLOC
    memcpy(&DeliveryData, rqst->data, size);

    if (verbose )
    {
        TMLog(" DELIVERY: w_id %d ", DeliveryData.w_id);
        TMLog(" DELIVERY: d_id %d ", DeliveryData.o_carrier_id);
    }

    bError = FALSE;

    DeliveryData.retval = SQLDelivery( pdbproc, &DeliveryData,
iDeadlockRetry);

    if (bError == TRUE)
        DeliveryData.retval = -1;

    memcpy( rqst->data, &DeliveryData, size);
#else
    memcpy(&TuxData, rqst->data, size);

    if (verbose )
    {
        TMLog(" DELIVERY: w_id %d ", TuxData.DeliveryData.w_id);
        TMLog(" DELIVERY: d_id %d ",
TuxData.DeliveryData.o_carrier_id);

```

```

    }

    bError = FALSE;

    TuxData.DeliveryData.retval = SQLDelivery( pdbproc,
&TuxData.DeliveryData, iDeadlockRetry);

    if (bError == TRUE)
        TuxData.DeliveryData.retval = -1;

    memcpy( rqst->data, &TuxData, size);
#endif
    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;

```



```

        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;
    }
    strcat(szLogPath, "delilog.");

    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);

// Do not set the packet size. Use the size set up in SQL
Server.
DBSETLPACKET(login, (unsigned short)DEFCLPACKSIZE);

// This can potentially cut down on data conversion
DBSETLVERSION(login, DBVER60);

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");

dbsqlxec(*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");

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

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

dbsqlxec(*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)_vsprintf( 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

```



```

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
                break;
            case '?':
                return TRUE;
        }
    }
}
return FALSE;

```

flush when written.

```

/* 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 [-v (verbose)] [-F (Flush delilog)]");
};
    TMLog("Delivery: Server %s Flush %d.", szServer, bFlush);
}

```

```

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

```

```

!ENDIF

!IF "$(SQL_LOC)" == ""
SQL_LOC=E:\mssql7\DevTools
!MESSAGE No SQL_LOC specified. Defaulting to D:\MSSQL\DLIB
!ENDIF

!IF "$(TUXDIR)" == ""
TUXDIR = E:\TUXEDO
!MESSAGE No TUXDIR specified. Defaulting to E:\TUXEDO
!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
DLIB            = $(SQL_LOC)
DLIBINC         = $(DLIB)\include
DLIBDIR         = $(DLIB)\lib

!IF "$(CFG)" != "Debug"
LDEBUG         =
CDEBUG         =
LDEBUG_RG      =
CDEBUG_RG      =
DEBUG          =
FLAGS          = /D "WIN32" /D "_WINDOWS" /D "_TMSTHEADS"
#FLAGS        = /D "WIN32" /D "_WINDOWS" /D "_TMSTHEADS" /D
"LOCAL_ALLOC"
CFLAGS         = /D "WIN32" /D "_WINDOWS" /D "_TMSTHEADS"
OPT            = /Ot
!ELSE
LDEBUG         = /debug /pdb:$(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 "WIN32" /D "_WINDOWS" /D "_TMSTHEADS"

```

```

#FLAGS      = /D "WIN32" /D "_WINDOWS" /D "_TMSTHEADS" /D
"LOCAL_ALLOC"
CFLAGS      = /D "WIN32" /D "_WINDOWS" /D "_TMSTHEADS"
OPT         = /Od
!ENDIF

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
$(LDEBBUG) /def:"$(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
$(LDEBBUG_RG) /out:"$(OUTDIR)\install.exe"

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

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

```

```

del neworder.obj

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

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

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

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

/* FILE: NEWORDER.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;

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;

#ifdef LOCAL_ALLOC
NEW_ORDER_DATA NewOrderData;
#else
TUX_DATA TuxData;
#endif
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: NewOrder: Server %s, Database %s, User
%s, Password %s, Flush %d.",
            szServer, szDatabase, szUser, szPassword,
bFlush);

```

```

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

```

```

    if ( SQLOpenConnection ( NULL, 0, 0, &pdbproc, szServer,
szDatabase, szUser, szPassword, szDatabase, &spId))
    {
        TMLog ( "NEWORDER: 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();
}

```

```

/* 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);
    int size = rqst->len;

```

```

#ifdef LOCAL_ALLOC
    memcpy(&NewOrderData, rqst->data, size);

    if (verbose )
    {
        TMLog(" NEWORDER: w_id %d ", NewOrderData.w_id);
        TMLog(" NEWORDER: d_id %d ", NewOrderData.d_id);
        TMLog(" NEWORDER: c_id %d ", NewOrderData.c_id);
    }

```

```

    }

    bError = FALSE;

    NewOrderData.retval = SQLNewOrder( NULL, 0, 0, pdbproc,
&NewOrderData, iDeadlockRetry);

    if (bError == TRUE)
        NewOrderData.retval = -1;

    if (verbose )
        TMLog(" NEWORDER: Return Value %d",
NewOrderData.retval);

    memcpy( rqst->data, &NewOrderData, size);
#else
    memcpy(&TuxData, rqst->data, size);

    if (verbose )
    {
        TMLog(" NEWORDER: w_id %d ", TuxData.NewOrderData.w_id);
        TMLog(" NEWORDER: d_id %d ", TuxData.NewOrderData.d_id);
        TMLog(" NEWORDER: c_id %d ", TuxData.NewOrderData.c_id);
    }

    bError = FALSE;

    TuxData.NewOrderData.retval = SQLNewOrder( NULL, 0, 0, pdbproc,
&TuxData.NewOrderData, iDeadlockRetry);

    if (bError == TRUE)
        TuxData.NewOrderData.retval = -1;

    if (verbose )
        TMLog(" NEWORDER: Return Value %d",
TuxData.NewOrderData.retval);

    memcpy( rqst->data, &TuxData.NewOrderData, size);
#endif
    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);

            pNewOrder->o_all_local = 1;
            for (i = 0; i < pNewOrder->o_ol_cnt; i++)
            {
                if ( pNewOrder->o_all_local && pNewOrder-
>Ol[i].ol_supply_w_id != pNewOrder->w_id )
                    pNewOrder->o_all_local = 0;
            }
            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 results 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))
        dbconvert(dbproc, SQLNUMERIC, pData, dbdatlen(dbproc,4),
SQLFLT8, (BYTE *)&pNewOrder->Ol[i].ol_i_price, 8);

    if (pData=dbdata(dbproc, 5))
        dbconvert(dbproc, SQLNUMERIC, pData, dbdatlen(dbproc,5),
SQLFLT8, (BYTE *)&pNewOrder->Ol[i].ol_amount, 8);

        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))

                    dbconvert(dbproc, SQLNUMERIC, pData, dbdatlen(dbproc,1),
SQLFLT8, (BYTE *)&pNewOrder->w_tax, 8);

                if (pData=dbdata(dbproc, 2))

                    dbconvert(dbproc, SQLNUMERIC, pData, dbdatlen(dbproc,2),
SQLFLT8, (BYTE *)&pNewOrder->d_tax, 8);

```

```

    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))
        dbconvert(dbproc, SQLNUMERIC, pData, dbdatlen(dbproc,5),
SQLFLT8, (BYTE *)&pNewOrder->c_discount, 8);

    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 committed.");
        return TRUE;
    }
    else
    {
        strcpy(pNewOrder->execution_status,"Item
number is not valid.");
        pNewOrder->error=ERR_BAD_ITEM_ID;
        return FALSE;
    }
}

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

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
 *          int iSyncId
 *          DBPROCESS **dbproc
 *          char *server SQL
 *          char *database SQL
server name
 *          char *user
server database
 *          char *password
 *          char *app
 *          int *spid
 *          long *pack_size
 *          pointer to returned application array
 *          pointer to returned spid
 *          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 dblink 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);

        // Do not set the packet size. Use the size set up in SQL
Server.

        DBSETLPACKET(login, (unsigned short)DEFCLPACKSIZE);

        // This can potentially cut down on data conversion
        DBSETLVERSION(login, DBVER60);

        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
*
* 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)_vsprintf( 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
severity of error
*             int           dberr           dberr
error id
*             int           oserr           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
*
*           message number          DBINT          msgno
*
*           message state          int          msgstate
*
*           message severity          int          severity
*
*           printable message description          char          *msgtext
*
* RETURNS:          int          INT_CONTINUE
continue if error is SQLETIME else INT_CANCEL action
*
*           cancel operation          INT_CANCEL
*
* 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 '?':
                    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 [-v (verbose)]" );
    TMLog("NewOrder: Server %s", szServer);
}

/*
 * FILE:          ORDERSTATUS.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;

#ifdef LOCAL_ALLOC
ORDER_STATUS_DATA OrderStatusData;
#else
TUX_DATA TuxData;
#endif
TERM Term;

static char szTpcLogPath[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 bTpcExit;
        //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: OrderStatus: Server %s, Database %s,
User %s, Password %s, Flush %d.",
            szServer, szDatabase, szUser, szPassword,
bFlush);

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

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

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

```

```

        dbexit();
    }

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

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

#ifdef LOCAL_ALLOC
    memcpy(&OrderStatusData, rqst->data, size);

    if (verbose )
    {
        TMLog(" ORDERSTATUS: w_id %d ", OrderStatusData.w_id);
        TMLog(" ORDERSTATUS: d_id %d ", OrderStatusData.d_id);
        TMLog(" ORDERSTATUS: c_id %d ", OrderStatusData.c_id);
    }

    bError = FALSE;

    OrderStatusData.retval = SQLOrderStatus( NULL, 0, 0, pdbproc,
&OrderStatusData, iDeadlockRetry);

    if (bError == TRUE)
        OrderStatusData.retval = -1;

    if ( verbose )
        TMLog(" ORDERSTATUS: Return Value %d",
OrderStatusData.retval);

    memcpy( rqst->data, &OrderStatusData, size);
#else
    memcpy(&TuxData, rqst->data, size);

    if (verbose )
    {
        TMLog(" ORDERSTATUS: w_id %d ",
TuxData.OrderStatusData.w_id);
        TMLog(" ORDERSTATUS: d_id %d ",
TuxData.OrderStatusData.d_id);

```

```

        TMLog(" ORDERSTATUS: c_id %d ",
TuxData.OrderStatusData.c_id);
    }

    bError = FALSE;

    TuxData.OrderStatusData.retval = SQLOrderStatus( NULL, 0, 0,
pdbproc, &TuxData.OrderStatusData, iDeadlockRetry);

    if (bError == TRUE)
        TuxData.OrderStatusData.retval = -1;

    if ( verbose )
        TMLog(" ORDERSTATUS: Return Value %d error = %d",
            TuxData.OrderStatusData.retval,
TuxData.OrderStatusData.error);

    memcpy( rqst->data, &TuxData, size);
#endif
    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))
{
1)      if (pData=dbdata (dbproc,
                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)      dbconvert (dbproc,
SQLNUMERIC, pData, dbdatlen(dbproc,4), SQLFLT8, (BYTE *)&pOrderStatus-
>OlOrderStatusData[i].ol_amount, 8);
                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))
                {
1)      if (pData=dbdata (dbproc,
                pOrderStatus->c_id
= (*(DBINT *) pData);
2)      if (pData=dbdata (dbproc,
                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)      dbconvert (dbproc,
SQLNUMERIC, pData, dbdatlen (dbproc,7), SQLFLT8, (BYTE *)&pOrderStatus-
>c_balance, 8);
                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.");
                    pOrderStatus->error=ERR_NOSUCH_CUSTOMER;
                    TMLog (" ORDERSTATUS: No such customer ");
                }
                else
                {
                    strcpy (pOrderStatus-
>execution_status, "Transaction committed.");
                    return 1;
                }
            }

```

```

    }
    // If we reached here, it means we quit after MAX_RETRY
deadlocks
    strcpy(pOrderStatus->execution_status,"Hit deadlock max. ");
    pOrderStatus->error=ERR_TYPE_DEADLOCK;
    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
            //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);

        // Do not set the packet size. Use the size set up in SQL
Server.
        DBSETLPACKET(login, (unsigned short)DEFCLPACKSIZE);

        // This can potentially cut down on data conversion
        DBSETLVERSION(login, DBVER60);

        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");

        dbsqlxec(*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");
    dbsqlxexec(*dbproc);
    while (dbresults(*dbproc) != NO_MORE_RESULTS)
    {
        while (dbnextrow(*dbproc) != NO_MORE_ROWS)
            ;
    }

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

    dbsqlxexec(*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)_vsprintf( 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
*
* message number DBINT msgno
*
* message state int msgstate
*
* message severity int severity
*
* printable message description char *msgtext
*
* 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 '?':
                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("OrderStatus: -S Server [-v (verbose)]" );
    TMLog("OrderStatus: Server %s", szServer);
}

/* FILE: STOCKLEVEL.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;

#ifdef LOCAL_ALLOC
STOCK_LEVEL_DATA StockLevelData;
#else
TUX_DATA TuxData;
#endif
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: StockLevel: Server %s, Database %s,
User %s, Password %s, Flush %d.",
            szServer, szDatabase, szUser, szPassword,
bFlush);
}

```

```

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

    if ( SQLOpenConnection ( NULL, 0, 0, &pdbproc, szServer,
szDatabase, szUser, szPassword, szDatabase, &spId))
    {
        TMLog ( "STOCKLEVEL: 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();
}

/* FUNCTION: STOCKLEVEL ( TPSVCINFO *rqst )
*
* PURPOSE:      Process a Stock Level request.
*
* RETURNS:      int      0      Success
*               -1      Failure
*
* COMMENTS:     None
*/

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

```

```

#ifdef LOCAL_ALLOC
    memcpy(&StockLevelData, rqst->data, size);

    if (verbose )
    {
        TMLog(" STOCKLEVEL: w_id %d ", StockLevelData.w_id);
        TMLog(" STOCKLEVEL: d_id %d ", StockLevelData.d_id);
        TMLog(" STOCKLEVEL: c_id %d ",
StockLevelData.thresh_hold);
    }

    bError = FALSE;

    StockLevelData.retval = SQLStockLevel( NULL, 0, 0, pdbproc,
&StockLevelData, iDeadlockRetry);

    if (bError == TRUE)
        StockLevelData.retval = -1;

    if ( verbose )
        TMLog(" STOCKLEVEL: Return Value %d",
StockLevelData.retval);

    memcpy( rqst->data, &StockLevelData, size);
#else
    memcpy(&TuxData, rqst->data, size);

    if (verbose )
    {
        TMLog(" STOCKLEVEL: w_id %d ",
TuxData.StockLevelData.w_id);
        TMLog(" STOCKLEVEL: d_id %d ",
TuxData.StockLevelData.d_id);
        TMLog(" STOCKLEVEL: c_id %d ",
TuxData.StockLevelData.thresh_hold);
    }

    bError = FALSE;

    TuxData.StockLevelData.retval = SQLStockLevel( NULL, 0, 0,
pdbproc, &TuxData.StockLevelData, iDeadlockRetry);

    if (bError == TRUE)
        TuxData.StockLevelData.retval = -1;

    if ( verbose )
        TMLog(" STOCKLEVEL: Return Value %d",
TuxData.StockLevelData.retval);

    memcpy( rqst->data, &TuxData, size);

```

```

#endif
    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) == SUCCEEDED)
        {
            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) == SUCCEEDED)
            {
                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 committed.");
                return TRUE;
            }
        }

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

    /*
    * 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);

        // Do not set the packet size. Use the size set up in SQL
Server.
        DBSETLPACKET(login, (unsigned short)DEFCLPACKSIZE);

        // This can potentially cut down on data conversion
        DBSETLVERSION(login, DBVER60);

        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)_vsprintf( 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 '?':
                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("StockLevel: -S Server [-v (verbose)]" );
    TMLog("StockLevel: Server %s", szServer);
}

/* FILE:          TPCC.H
 *
 * Microsoft TPC-C Kit Ver. 3.00.001
 * 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
 */

#define LOCAL_ALLOC 1

//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

```

```

#define ERR_BAD_ITEM_ID
1 //expected abort record in txnRecord
#define ERR_TYPE_DELIVERY_POST
//expected delivery post failed
#define ERR_TYPE_WEBDDL
3 //tpcc web generated error
#define ERR_TYPE_SQL
//sql server generated error
#define ERR_TYPE_DBLIB
//dblib generated error
#define ERR_TYPE_ODBC
//odbc generated error
#define ERR_TYPE_SOCKET
7 //error on communication socket client rte only
#define ERR_TYPE_DEADLOCK
//dblib and odbc only deadlock condition
#define ERR_TYPE_TUXEDO
9 //tuxedo error

#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 corresponding Item_Id.
#define ERR_NEWORDER_QTY_WITHOUT_ITEMID
1040
//New Order Qty entered without a corresponding 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)

//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 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;

#ifdef USE_ODBC
typedef struct _DBPROCESS
{
    HDBC      hdbc;
    HSTMT     hstmt;
    int       spid;
    void      *uPtr;
} DBPROCESS, *PDBPROCESS;

//dblib error message return values
#define INT_EXIT      0
#define INT_CONTINUE  1
#define INT_CANCEL    2
#endif

//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

    PDBPROCESS dbproc;                //dblib
connection pointer
    int         spid;
    //spid assigned from dblib
    int         iSyncId;
    //synchronization 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

#ifdef LOCAL_ALLOC
    TUX_DATA        *TuxDataPtr;        //Tuxedo
Data Structure for all transactions
#endif // LOCAL_ALLOC

} 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;
    //synchronization 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

    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

//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;

//function prototypes

BOOL WINAPI 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 BeginCmdForm(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);
void NumberOfConnectionsCmd(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 LogTuxError(int Tprc, char *ErrMsg);
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 Rollback, 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, BOOL bSuccess);
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 void FormatHTMLString(char *szBuff, char *szStr, int iLen);
static void PrintParameters(void);

#ifdef USE_ODBC
void dbsetuserdata(PDBPROCESS dbproc, void *uPtr);
void *dbgetuserdata(PDBPROCESS dbproc);
void BindParameter(PDBPROCESS dbproc, UWORD ipar, SWORD fCType,
SWORD fSqlType, UDWORD cbColDef, SWORD ibScale, PTR rgbValue, SDWORD
cbValueMax);
void ODBCError(PDBPROCESS dbproc);
BOOL ExecuteStatement(PDBPROCESS dbproc, char *szStatement);
BOOL BindColumn(PDBPROCESS dbproc, SQLUSMALLINT icol,
SQLSMALLINT fCType, SQLPOINTER rgbValue, SQLINTEGER cbValueMax);
BOOL GetResults(PDBPROCESS dbproc);
BOOL MoreResults(PDBPROCESS dbproc);
BOOL ReopenConnection(PDBPROCESS dbproc);
#endif

/* 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
*/

#ifdef _INC_TRANS
#define _INC_TRANS

#ifdef USE_ODBC
#ifdef TIMESTAMP_STRUCT

```

```

#include <sqltypes.h>
#endif
#else
#ifdef _INC_SQLFRONT
#include <sqlfront.h>
#endif
#endif

#ifdef 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;

```



```

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
    DBDATAREC       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;
    int             error;
    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;

```

Web Client

```

/*****
*
* Copyright (c) 1995 Process Software Corporation
*
* Copyright (c) 1995 Microsoft Corporation
*
*
* Module Name : HttpExt.h
*
* Abstract :
*

```

```

    long           num_deadlocks;
    long           o_id[10];
    int            retval;
    int            error;
    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;
    int            error;
    char           execution_status[STATUS_LEN];
} STOCK_LEVEL_DATA;

typedef struct
{
    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
} TUX_DATA;

#endif

* 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

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_

/*      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     iQSlots;

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    StartWWWService(void);
static BOOL    StopWWWService(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 = 500;
            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_THREAD_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_THREAD_POOL_LIMIT, &d, FALSE);

```

```

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

        ShowWindow(hwnd, SW_HIDE);
        hDlg = CreateDialog(hInst,
MAKEINTRESOURCE(IDD_DIALOG3), hwnd, CopyDlgProc);
        ShowWindow(hDlg, SW_SHOWNA);
        UpdateDialog(hDlg);
        rc = CopyFiles(hDlg, szDllPath);
        if ( !rc )
        {
                ShowWindow(hwnd, SW_SHOWNA);
                DestroyWindow(hDlg);
                MessageBox(hwnd, "Error(s) ocured
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 = 500;
            size = sizeof(szTmp);
            if ( RegQueryValueEx(hKey, "MaximumWarehouses", 0, &type,
szTmp, &size) == ERROR_SUCCESS )
                    iMaxWareHouse = atoi(szTmp);
            if ( iMaxWareHouse == 0 )
                    iMaxWareHouse = 500;
            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 = !StopWWWService();
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;
}

```

```

//{{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_THREAD_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

```

```

//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

#ifdef _AFXDLL || 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_THREAD_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_RTLDREADING
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
Successful",IDC_RESULTS,7,22,
              102,18,0,WS_EX_CLIENTEDGE
  ICON        IDI_ICON2,IDC_STATIC,50,7,18,20,SS_REALSIZEIMAGE,
              WS_EX_TRANSPARENT
END

IDD_DIALOG3 DIALOG DISCARDABLE 0, 0, 91, 40
STYLE DS_SYSMODAL | DS_MODALFRAME | DS_3DLOOK | DS_CENTER | WS_CAPTION
CAPTION "Installing 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"

////////////////////////////////////
////
//
// TPCDLL
//

IDR_TPCDLL1        TPCDLL DISCARDABLE    "tpcc.dll"

#ifdef _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
////////////////////////////////////
////

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

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

//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by TPCC.rc
//

// Next default values for new objects
//

```

```

#ifdef APSTUDIO_INVOKED
#ifdef 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

/*      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.
 *      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 <sqlldb.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>

static TPINIT *tpinf;
static DWORD   TLISisTpInitedKey;
static int     ThrTpInit();

char   szServer[32]   = "EDW"; //global variables used with
this DLL
char   szUser[32]     = "sa";
char   szPassword[32] = "";
char   szDatabase[32] = "tpcc";

BOOL   bLog           = FALSE;
BOOL   dLog           = FALSE;

int     iThreads      = 5;
int     iMaxWareHouses = 500;
int     iQSlotts      = 3000;
int     iDelayMs      = 100;
int     iConnectDelay = 500;
short   iDeadlockRetry = (short)3;
short   iMaxConnections = (short)25;
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", "Users", ""
};

//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,

```



```

};
    NumberOfConnectionsCmd

//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>"
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=\"TERMIN\" VALUE=\"-2\">"
    "<INPUT
TYPE=\"hidden\" NAME=\"SYCID\" 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 based 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;
    }
}

```

```

later
        TLSIsTpInitedKey = TlsAlloc(); // check for failure
        // assumption:value inited to 0
        break;
case DLL_THREAD_ATTACH:
        break;
case DLL_THREAD_DETACH:
        if ( dLog )
        {
                SYSTEMTIME      systemTime;
                FILE *fp;

                GetLocalTime(&systemTime);
                fp = fopen(szErrorLogPath, "ab");
                fprintf(fp, "\r\nError: %2.2d/%2.2d/%2.2d
%2.2d:%2.2d:%2.2d\r\n",
                        systemTime.wYear, systemTime.wMonth,
                        systemTime.wDay,
                        systemTime.wHour,
                        systemTime.wMinute, systemTime.wSecond);
                fprintf(fp, "DLL_THREAD_DETACH \r\n");
                fclose(fp);
        }
        break;
case DLL_PROCESS_DETACH:
        if ( pSD )
                free( pSD );

        bTpccExit = TRUE;

        (*Term.Restore)();

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

        TlsFree(TLSIsTpInitedKey);
        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);
        lstrcpy(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 ( !strcmp(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, FALSE, 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. TERMIID= 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. TERMIID= 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) );
    WriteZString(pECB, MakeDeliveryForm(iTermId, iSyncId, TRUE, 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. TERMIID= 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 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. TERMIID= 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. TERMIID= 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. TERMIID= 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;

    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. TERMIID= 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. TERMIID= 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 NumberOfConnectionsCmd(EXTENSION_CONTROL_BLOCK *pECB,
int iFormId, int iTermId, int iSyncId)
*
* PURPOSE: This function returns to the browser the total number of
active 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: None
*/

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

    // EnterCriticalSection(&CriticalSection);

    iTotal = 0;

    for(i=0; i<Term.iAvailable; i++)
    {
        if ( Term.pClientData[i].inUse )
            iTotal++;
    }

    // LeaveCriticalSection(&CriticalSection);

    h_printf(pECB, "Total Active Connections: %d", iTotal);

    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;
    int iSize;
    char szHeader[128];
    char szHeader1[128];

    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);
    }

    iSize = sprintf(szHeader, "200 Ok");
    sprintf(szHeader1, "Connection: keep-alive\r\nContent-type:
text/html\r\nContent-length: %d\r\n\r\n", lpbSize);
}

```

```

#ifdef PURE_PERFORMIX
    (*pECB->ServerSupportFunction)(pECB->ConnID,
HSE_REQ_DONE_WITH_SESSION, NULL, 0, 0);
#else
    (*pECB->ServerSupportFunction)(pECB->ConnID,
HSE_REQ_SEND_RESPONSE_HEADER, szHeader, &iSize, (LPDWORD)szHeader1);
#endif

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

    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.
*
* printf style format string
*
* char
*
* 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);

    wsprintf(szBuff, "<html>%s</html>", szTmp) + 1;

    WriteZString(pECB, szBuff);
}

```

```

    return;
}

void LogTuxError( int TpErrno, char *ErrorMessage )
{
    FILE *fp;
    SYSTEMTIME systemTime;

    GetLocalTime(&systemTime);

    fp = fopen(szErrorLogPath, "ab");
    fprintf(fp, "\r\nError: %2.2d/%2.2d/%2.2d %2.2d:%2.2d:%2.2d\r\n",
        systemTime.wYear, systemTime.wMonth, systemTime.wDay,
        systemTime.wHour, systemTime.wMinute, systemTime.wSecond);
    fprintf(fp, "Thread %d: TPCCWEB(%d): %s: %s",
        GetCurrentThreadId(), TpErrno, tpsterror(TpErrno),
        ErrorMessage);
    fclose(fp);
}

/* 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;
    SYSTEMTIME      systemTime;

    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 corresponding Item_Id."
    },
    {
        ERR_NEWORDER_QTY_WITHOUT_ITEMID,
        "New Order Qty entered without a corresponding 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*\"."
        },
        {
            0,
            ""
        }
    };

static char szNoMsg[] = "";
char      *szForm;

GetLocalTime(&systemTime);

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\\\">");
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"STATUSID\\\" VALUE=\\\"%d\\\">", iErrorType);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"ERROR\\\" VALUE=\\\"%d\\\">", iError);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"TERMINID\\\" VALUE=\\\"%d\\\">", iTermId);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"SYNCID\\\" VALUE=\\\"%d\\\">", iSyncId);

```

```

        sprintf(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\\\">");
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"STATUSID\\\" VALUE=\\\"%d\\\">", iErrorType);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"ERROR\\\" VALUE=\\\"%d\\\">", iError);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"TERMINID\\\" VALUE=\\\"%d\\\">", iTermId);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"SYNCID\\\" VALUE=\\\"%d\\\">", iSyncId);
        sprintf(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\\\">");
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"STATUSID\\\" VALUE=\\\"%d\\\">", iErrorType);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"ERROR\\\" VALUE=\\\"%d\\\">", iError);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"TERMINID\\\" VALUE=\\\"%d\\\">", iTermId);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"SYNCID\\\" VALUE=\\\"%d\\\">", iSyncId);
        sprintf(szForm+strlen(szForm), "Error: DBLIB(%d):
%s", iError, szMsg);
        strcat(szForm, "</FORM><BODY></HTML>");
        WriteZString(pECB, szForm);
        break;
    case ERR_TYPE_ODBC:
        strcpy(szForm, "<HTML><HEAD><TITLE>Welcome To TPC-
C</TITLE></HEAD><BODY><FORM ACTION=\\\"tpcc.dll\\\" METHOD=\\\"GET\\\">");
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"STATUSID\\\" VALUE=\\\"%d\\\">", iErrorType);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"ERROR\\\" VALUE=\\\"%d\\\">", iError);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"TERMINID\\\" VALUE=\\\"%d\\\">", iTermId);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\\\"hidden\\\" NAME=\\\"SYNCID\\\" VALUE=\\\"%d\\\">", iSyncId);
        sprintf(szForm+strlen(szForm), "Error: ODBC");
        strcat(szForm, "</FORM><BODY></HTML>");
        WriteZString(pECB, szForm);
        break;
    case ERR_TYPE_SOCKET:

```

```

        strcpy(szForm, "<HTML><HEAD><TITLE>Welcome To TPC-
C</TITLE></HEAD><BODY><FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">");
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">", iErrorType);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"%d\">", iError);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">", iTermId);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">", iSyncId);
        sprintf(szForm+strlen(szForm), "Error: SOCKET");
        strcat(szForm, "</FORM><BODY></HTML>");
        WriteZString(pECB, szForm);
        break;
    case ERR_TYPE_DEADLOCK:
        strcpy(szForm, "<HTML><HEAD><TITLE>Welcome To TPC-
C</TITLE></HEAD><BODY><FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">");
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">", iErrorType);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"%d\">", iError);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"%d\">", iTermId);
        sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">", iSyncId);
        sprintf(szForm+strlen(szForm), "Error: Deadlock");
        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 successfull
              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, iTotConnections, iTickCount;

    EnterCriticalSection(&CriticalSection);

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

    if ( iTotConnections >= 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;
    }
}

#ifdef LOCAL_ALLOC
    tpfree((char *)Term.pClientData[id].TuxDataPtr);
#endif // Not LOCAL_ALLOC

}

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];
#ifdef LOCAL_ALLOC
    char    buf[64];
    int     TpRc;
#endif // Not LOCAL_ALLOC

    sprintf(szApp, "TPCC:%ld", (int)iTermId);

    Term.pClientData[iTermId].dbproc = NULL;

#ifdef LOCAL_ALLOC // Globally allocate tuxedo structures
    if ( dLog )
    {
        FILE *fp;
        fp = fopen(szTpccLogPath, "ab");
        fprintf(fp, "Sizeof(TUX_DATA) %d \r\n",
sizeof(TUX_DATA));
        fprintf(fp, "Sizeof(NEW_ORDER_DATA) %d \r\n",
sizeof(NEW_ORDER_DATA));
        fprintf(fp, "Sizeof(PAYMENT_DATA) %d \r\n",
sizeof(PAYMENT_DATA));
        fprintf(fp, "Sizeof(ORDER_STATUS_DATA) %d \r\n",
sizeof(ORDER_STATUS_DATA));
        fprintf(fp, "Sizeof(DELIVERY_DATA) %d \r\n",
sizeof(DELIVERY_DATA));
        fprintf(fp, "Sizeof(STOCK_LEVEL_DATA) %d \r\n",
sizeof(STOCK_LEVEL_DATA));
        fclose(fp);
    }

//    Add initialization of Tuxedo Structures

    if ((Term.pClientData[iTermId].TuxDataPtr = (TUX_DATA
*)tpalloc("CARRAY", NULL, sizeof(TUX_DATA))) == NULL)
    {
        TpRc = tperrno;
        sprintf(buf, "Tuxedo tpalloc failed:");
        LogTuxError(TpRc, buf);
        ErrorMessage(pECB, ERR_CANNOT_INIT_TERMINAL,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return TRUE;
    }
}

```

```

    if ( dLog )
    {
        FILE *fp;

        fp = fopen(szTpccLogPath, "ab");

        fprintf(fp, "Thread %d iTermId %d * TuxDataPtr: %x
\r\n",
                GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].TuxDataPtr);
        fclose(fp);
    }
#endif // LOCAL_ALLOC

    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
 *          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;

    strcpy(szForm, "<HTML><HEAD><TITLE>TPC-C Stock
Level</TITLE></HEAD>");
    strcat(szForm, "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">");
    if ( bInput )
        strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"PI*\"
VALUE=\"\>");
        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=\"FORMID\" VALUE=\"%d\>", STOCK_LEVEL_FORM);
        wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMINID\" VALUE=\"%d\>", iTermId);
        wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\>", 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);
        if ( bInput )
        {
            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\\>" );
        }
        else
        {
            wsprintf(szForm+strlen(szForm), "Stock Level Threshold:
%2.2d<BR><BR>", Term.pClientData[iTermId].StockLevelData.thresh_hold);
            wsprintf(szForm+strlen(szForm), "low stock:
%3.3d</PRE><BR><HR>",
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..\\>" );
        }

        strcat(szForm, "</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=\"TERMINID\\" 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, int SyncId, BOOL
Rollback, 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 Rollback,
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;

    strcpy(szForm, "<HTML>"

" <HEAD><TITLE>TPC-C New

Order</TITLE></HEAD><BODY>"

" <FORM ACTION=\"tpcc.dll\"

METHOD=\"GET\">" );

```

```

    if ( bInput )
    {
        strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"PI*\"
VALUE=\"\">");
        strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">");
    }
    else
    {
        if ( bValid )
            strcat(szForm, "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"0\">");
        else
            wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">", ERR_BAD_ITEM_ID);
    }

    if (Rollback == FALSE)
        strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"0\">");
    else
        strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"1\">");

    wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">", NEW_ORDER_FORM);
    wsprintf(szForm+strlen(szForm),
" <INPUT TYPE=\"hidden\"
NAME=\"TERMINID\" VALUE=\"%d\">", iTermId);
    wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">", iSyncId);
    strcat(szForm, "<PRE>
New
Order<BR>");

    if ( bInput )
    {
        wsprintf(szForm+strlen(szForm), "Warehouse: %4.4d
District: <INPUT NAME=\"DID*\" SIZE=1>
Term.pClientData[iTermId].NewOrderData.w_id);
        Date:<BR>",
            strcat(szForm, "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
" <INPUT NAME=\"SP00*\"
SIZE=4> <INPUT NAME=\"IID00*\" SIZE=6>
NAME=\"Qty00*\" SIZE=1><BR>"
            <INPUT
" <INPUT NAME=\"SP01*\"
SIZE=4> <INPUT NAME=\"IID01*\" SIZE=6>
NAME=\"Qty01*\" SIZE=1><BR>"
            <INPUT
" <INPUT NAME=\"SP02*\"
SIZE=4> <INPUT NAME=\"IID02*\" SIZE=6>
NAME=\"Qty02*\" SIZE=1><BR>"
            <INPUT

```

```

SIZE=4> <INPUT NAME="\IID03*" SIZE=6>
NAME="\Qty03*" SIZE=1><BR>"
" <INPUT NAME="\SP03*"
<INPUT

SIZE=4> <INPUT NAME="\IID04*" SIZE=6>
NAME="\Qty04*" SIZE=1><BR>"
" <INPUT NAME="\SP04*"
<INPUT

SIZE=4> <INPUT NAME="\IID05*" SIZE=6>
NAME="\Qty05*" SIZE=1><BR>"
" <INPUT NAME="\SP05*"
<INPUT

SIZE=4> <INPUT NAME="\IID06*" SIZE=6>
NAME="\Qty06*" SIZE=1><BR>"
" <INPUT NAME="\SP06*"
<INPUT

SIZE=4> <INPUT NAME="\IID07*" SIZE=6>
NAME="\Qty07*" SIZE=1><BR>"
" <INPUT NAME="\SP07*"
<INPUT

SIZE=4> <INPUT NAME="\IID08*" SIZE=6>
NAME="\Qty08*" SIZE=1><BR>"
" <INPUT NAME="\SP08*"
<INPUT

SIZE=4> <INPUT NAME="\IID09*" SIZE=6>
NAME="\Qty09*" SIZE=1><BR>"
" <INPUT NAME="\SP09*"
<INPUT

SIZE=4> <INPUT NAME="\IID10*" SIZE=6>
NAME="\Qty10*" SIZE=1><BR>"
" <INPUT NAME="\SP10*"
<INPUT

SIZE=4> <INPUT NAME="\IID11*" SIZE=6>
NAME="\Qty11*" SIZE=1><BR>"
" <INPUT NAME="\SP11*"
<INPUT

SIZE=4> <INPUT NAME="\IID12*" SIZE=6>
NAME="\Qty12*" SIZE=1><BR>"
" <INPUT NAME="\SP12*"
<INPUT

SIZE=4> <INPUT NAME="\IID13*" SIZE=6>
NAME="\Qty13*" SIZE=1><BR>"
" <INPUT NAME="\SP13*"
<INPUT

SIZE=4> <INPUT NAME="\IID14*" SIZE=6>
NAME="\Qty14*" SIZE=1><BR>"
" <INPUT NAME="\SP14*"
<INPUT

"Execution Status:
"<INPUT TYPE="\submit\"
"<INPUT TYPE="\submit\"
"</FORM>"
"</HTML>" );

}
else
{
    if ( bValid )
        wsprintf(szForm+strlen(szForm), "Warehouse: %4.4d
District: %2.2d Date: %2.2d-%2.2d-%4.4d
%2.2d:%2.2d:%2.2d <BR>",

```

```

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);
}
else
{
    wsprintf(szForm+strlen(szForm), "Warehouse: %4.4d
District: %2.2d Date:<BR>",
Term.pClientData[iTermId].NewOrderData.w_id,
Term.pClientData[iTermId].NewOrderData.d_id);
}
FormatHTMLString(szName,
Term.pClientData[iTermId].NewOrderData.c_last, 16),
FormatHTMLString(szCredit,
Term.pClientData[iTermId].NewOrderData.c_credit, 2);
wsprintf(szForm+strlen(szForm), "Customer: %4.4d Name:
%s Credit: %s ",
Term.pClientData[iTermId].NewOrderData.c_id,
szName, szCredit);
if ( bValid )
{
    sprintf(szForm+strlen(szForm), "%5.2f
<BR>", Term.pClientData[iTermId].NewOrderData.c_discount * 100.0);
    sprintf(szForm+strlen(szForm), "Order Number: %8.8d
Number of Lines: %2.2d W_tax: %5.2f D_tax: %5.2f <BR><BR>",
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);

```

```

        strcat(szForm, " Supp_W Item_Id Item Name
Qty Stock B/G Price Amount<BR>");
        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 %s %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,
szName,
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 );
        }
    }
    else
    {
        strcat(szForm, "%Disc:<BR>");
        sprintf(szForm+strlen(szForm), "Order Number: %8.8d
Number of Lines: W_tax: D_tax:<BR><BR>",
Term.pClientData[iTermId].NewOrderData.o_id);

        strcat(szForm, " Supp_W Item_Id Item Name
Qty Stock B/G Price Amount<BR>");

        i = 0;
        for(; i<15; i++)
            strcat(szForm, "<BR>");

        if ( bValid )
        {
            sprintf(szForm+strlen(szForm), "Execution Status:
%24.24s Total: $%8.2f ",
Term.pClientData[iTermId].NewOrderData.execution_status,
Term.pClientData[iTermId].NewOrderData.total_amount);
        }
    }
}

```

```

    {
        sprintf(szForm+strlen(szForm), "Execution Status:
%24.24s Total:",
Term.pClientData[iTermId].NewOrderData.execution_status);
    }

    strcat(szForm, "</PRE><HR><BR>"
NAME="CMD" VALUE="..NewOrder..">"
NAME="CMD" VALUE="..Payment..">"
NAME="CMD" VALUE="..Delivery..">"
NAME="CMD" VALUE="..Order-Status..">"
NAME="CMD" VALUE="..Stock-Level..">"
NAME="CMD" VALUE="..Exit..">" );
    strcat(szForm, "</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;

strcpy(szForm, "<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD><BODY>"
" <FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">");
if ( bInput )
    strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"PI*\"
VALUE=\"\">");

    strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">");
    strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"0\">");
    sprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">", PAYMENT_FORM);
    sprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">", iTermId);
    sprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">", iSyncId);

    strcat(szForm, "<PRE>
Payment<BR>");

if ( bInput )
    strcat(szForm, "Date:<BR><BR> ");
else
{
    sprintf(szForm+strlen(szForm), "Date: %2.2d-%2.2d-%4.4d
%2.2d:%2.2d:%2.2d <BR><BR>",
        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);
}

```

```

    sprintf(szForm+strlen(szForm), "Warehouse: %4.4d",
Term.pClientData[iTermId].PaymentData.w_id);

    if ( bInput )
    {
        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
    {
        sprintf(szForm+strlen(szForm), "
District: %2.2d<BR>",
            Term.pClientData[iTermId].PaymentData.d_id);

        FormatHTMLString(szTmpStr1,
Term.pClientData[iTermId].PaymentData.w_street_1, 20);
        FormatHTMLString(szTmpStr2,
Term.pClientData[iTermId].PaymentData.d_street_1, 20);

        sprintf(szForm+strlen(szForm), "%s
%s<BR>", szTmpStr1, szTmpStr2);

        FormatHTMLString(szTmpStr1,
Term.pClientData[iTermId].PaymentData.w_street_2, 20);
        FormatHTMLString(szTmpStr2,
Term.pClientData[iTermId].PaymentData.d_street_2, 20);

        sprintf(szForm+strlen(szForm), "%s
%s<BR>", szTmpStr1, szTmpStr2);
    }

```

```

        FormatString(szW_Zip, szZipPic,
Term.pClientData[iTermId].PaymentData.w_zip);
        FormatString(szD_Zip, szZipPic,
Term.pClientData[iTermId].PaymentData.d_zip);

        FormatHTMLString(szTmpStr1,
Term.pClientData[iTermId].PaymentData.w_city, 20);
        FormatHTMLString(szTmpStr2,
Term.pClientData[iTermId].PaymentData.w_state, 2);
        FormatHTMLString(szTmpStr3,
Term.pClientData[iTermId].PaymentData.d_city, 20);
        FormatHTMLString(szTmpStr4,
Term.pClientData[iTermId].PaymentData.d_state, 2);

        wsprintf(szForm+strlen(szForm), "%s %s %10.10s      %s %s
%10.10s<BR><BR>",
                szTmpStr1, szTmpStr2, szW_Zip, szTmpStr3,
szTmpStr4, szD_Zip );

        wsprintf(szForm+strlen(szForm), "Customer: %4.4d Cust-
Warehouse: %4.4d Cust-District: %2.2d<BR>",
                Term.pClientData[iTermId].PaymentData.c_id,
                Term.pClientData[iTermId].PaymentData.c_w_id,
                Term.pClientData[iTermId].PaymentData.c_d_id);

        FormatHTMLString(szTmpStr1,
Term.pClientData[iTermId].PaymentData.c_first, 16);
        FormatHTMLString(szTmpStr2,
Term.pClientData[iTermId].PaymentData.c_middle, 2);
        FormatHTMLString(szTmpStr3,
Term.pClientData[iTermId].PaymentData.c_last, 16);

        wsprintf(szForm+strlen(szForm), "Name:   %s %s %s
Since:  %2.2d-%2.2d-%4.4d<BR>",
                szTmpStr1, szTmpStr2, szTmpStr3,
                Term.pClientData[iTermId].PaymentData.c_since.day,

                Term.pClientData[iTermId].PaymentData.c_since.month,

                Term.pClientData[iTermId].PaymentData.c_since.year);

        FormatHTMLString(szTmpStr1,
Term.pClientData[iTermId].PaymentData.c_street_1, 20);
        FormatHTMLString(szTmpStr2,
Term.pClientData[iTermId].PaymentData.c_credit, 2);

        wsprintf(szForm+strlen(szForm), "          %s
Credit: %s<BR>", szTmpStr1, szTmpStr2);

        FormatHTMLString(szTmpStr1,
Term.pClientData[iTermId].PaymentData.d_street_2, 20);
        sprintf(szForm+strlen(szForm), "          %s
%%Disc:  %5.2f<BR>",

```

```

                szTmpStr1,
Term.pClientData[iTermId].PaymentData.c_discount * 100.0);

        FormatString(szC_Zip, szZipPic,
Term.pClientData[iTermId].PaymentData.c_zip);
        FormatString(szC_Phone, "XXXXXX-XXX-XXX-XXXX",
Term.pClientData[iTermId].PaymentData.c_phone);

        FormatHTMLString(szTmpStr1,
Term.pClientData[iTermId].PaymentData.c_city, 20);
        FormatHTMLString(szTmpStr2,
Term.pClientData[iTermId].PaymentData.c_state, 2);

        wsprintf(szForm+strlen(szForm), "          %s %s %10.10s
Phone:  %-19.19s<BR><BR>",
                szTmpStr1, szTmpStr2, szC_Zip, szC_Phone );

        sprintf(szForm+strlen(szForm), "Amount Paid:
$%7.2f      New Cust Balance: $%14.2f<BR>",
                Term.pClientData[iTermId].PaymentData.h_amount,
                Term.pClientData[iTermId].PaymentData.c_balance);

        sprintf(szForm+strlen(szForm), "Credit Limit:
$%13.2f<BR><BR>",

                Term.pClientData[iTermId].PaymentData.c_credit_lim);

        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 <= 1 )
                                UtilStrCpy(szTmp, ptr, 50);
                        else
                                szTmp[0] = 0;
                        if ( !i )
                        {
                                FormatHTMLString(szTmpStr1, szTmp,
50);
                                wsprintf(szForm+strlen(szForm),
"Cust-Data: %s<BR>", szTmpStr1);
                        }
                        else
                        {
                                FormatHTMLString(szTmpStr1, szTmp,
50);
                                wsprintf(szForm+strlen(szForm), "
%s<BR>", szTmpStr1);
                        }
                }
        }

```



```

    }
    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;

        strcpy(szForm, "<HTML><HEAD><TITLE>TPC-C Order-
Status</TITLE></HEAD><BODY>"
                "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">");

        if ( bInput )
            strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"PI*\"
VALUE=\"\">");

            strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">");
            strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"0\">");
            sprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">", ORDER_STATUS_FORM);
            sprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMINID\" VALUE=\"%d\">", iTermId);
            sprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">", iSyncId);

            strcat(szForm, "<PRE>                                Order-
Status<BR>" );
            sprintf(szForm+strlen(szForm), "Warehouse: %4.4d  ",
Term.pClientData[iTermId].OrderStatusData.w_id);

            if ( bInput )
            {
                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
            {
                sprintf(szForm+strlen(szForm), "District: %2.2d<BR>",
Term.pClientData[iTermId].OrderStatusData.d_id);

                FormatHTMLString(c_first,
Term.pClientData[iTermId].OrderStatusData.c_first, 16);

```

```

        FormatHTMLString(c_middle,
Term.pClientData[iTermId].OrderStatusData.c_middle, 2);
        FormatHTMLString(c_last,
Term.pClientData[iTermId].OrderStatusData.c_last, 16);

        sprintf(szForm+strlen(szForm), "Customer: %4.4d  Name:
%s %s %s<BR>",
                Term.pClientData[iTermId].OrderStatusData.c_id,
c_first, c_middle, c_last);

        sprintf(szForm+strlen(szForm), "Cust-Balance:
$%9.2f<BR><BR>",
                Term.pClientData[iTermId].OrderStatusData.c_balance);

        sprintf(szForm+strlen(szForm), "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.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, BOOL bSuccess)
*
* 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 bSuccess TRUE if
Delivery succeeded 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, BOOL
bSuccess)
{
    char    *szForm;

    szForm = (char *)Term.pClientData[iTermId].szBuffer;

    Term.pClientData[iTermId].DeliveryData.w_id =
Term.pClientData[iTermId].w_id;

    strcpy( szForm,          "<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD><BODY>"
           "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">");

    if ( bInput )
    {
        strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"PI*\"
VALUE=\"\">");
        strcat(szForm, "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">");
    }
    else
    {
        if ( !bSuccess )
        {
            sprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">",
ERR_TYPE_DELIVERY_POST);
            strcat(szForm, "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"2\">");
        }
        else
        {
            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=\"FORMID\" VALUE=\"%d\">", DELIVERY_FORM);
    wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">", iTermId);
    wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">", iSyncId);

    strcat(szForm, "<PRE>
Delivery<BR> ");

    wsprintf(szForm+strlen(szForm), "Warehouse: %4.4d<BR><BR>",
Term.pClientData[iTermId].DeliveryData.w_id);

```

```

        if ( bInput )
            strcat( szForm, "Carrier Number: <INPUT NAME=\"OCD*\"
SIZE=1><BR><BR>");
        else
        {
            wsprintf(szForm+strlen(szForm), "Carrier Number:
%2.2d<BR><BR>",

Term.pClientData[iTermId].DeliveryData.o_carrier_id);
        }
        if ( bInput )
        {
            strcat( szForm, "Execution Status:<BR></PRE>"
" <HR><INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">"
" <INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">" );
        }
        else
        {
            wsprintf(szForm+strlen(szForm), "Execution Status:
%25.25s<BR></PRE>",

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..\">" );
        }

        strcat( szForm,          "</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
*/

#ifdef LOCAL_ALLOC // Allocate the tmalloc structure for each transaction
// This saves on some memory at the expense of some CPU cycles.
static void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId)
{
    int  TpRc, iRc, iError;
    long ilen, *olen;
    char buf[128];

```

```

NEW_ORDER_DATA          *NewOrderDataPtr; //New Order
Tuxedo Buffer

    if((iRc = ThrTpInit()) <0)
    {
        // This is bad
        sprintf(buf, "ProcessNewOrder Failed ThrTpInit: iRc =
%d", iRc);
        LogTuxError(0, buf);
        ErrorMessage(pECB, ERR_NEW_ORDER_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }

    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;
    }

    if ((NewOrderDataPtr = (NEW_ORDER_DATA *)tpalloc("CARRAY", NULL,
sizeof(NEW_ORDER_DATA))) == NULL)
    {
        TpRc = tperrno;
        sprintf(buf, "ProcessNewOrder Tpcalloc Failed");
        LogTuxError(TpRc, buf);
        ErrorMessage(pECB, ERR_NEW_ORDER_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }

    *NewOrderDataPtr = Term.pClientData[iTermId].NewOrderData;

    ilen = sizeof(NEW_ORDER_DATA);
    olen = &ilen;

    if ( dLog )
    {
        FILE *fp;

        fp = fopen(szTpccLogPath, "ab");
        fprintf(fp, "* ProcessNewOrderL Thread %d iTermId %d
NewOrderDataPtr: %x size %d \r\n",

```

```

        GetCurrentThreadId(), iTermId, &NewOrderDataPtr,
sizeof(*NewOrderDataPtr));
        fclose(fp);
    }

    if ((iRc = tpcall("NEWORDER", (char *)NewOrderDataPtr, ilen,
        (char **)&NewOrderDataPtr, (long *)olen, TPSIGRSTRT)) ==
-1)
    {
        TpRc = tperrno;
        sprintf(buf, "Neworder tpcall failed");
        LogTuxError(TpRc, buf);
        ErrorMessage(pECB, ERR_NEW_ORDER_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }

    if ( dLog )
    {
        FILE *fp;

        fp = fopen(szTpccLogPath, "ab");
        fprintf(fp, "*** ProcessNewOrderL Thread %d iTermId %d
NewOrderDataPtr: %x size %d\r\n",
        GetCurrentThreadId(), iTermId, &NewOrderDataPtr,
sizeof(*NewOrderDataPtr));
        fclose(fp);
    }

    Term.pClientData[iTermId].NewOrderData = *NewOrderDataPtr;

iRc = NewOrderDataPtr->retval;
iError = NewOrderDataPtr->error;

    tpfree((char *)NewOrderDataPtr);

    if ( iRc < 0 )
    {
        if (iError == ERR_TYPE_DEADLOCK)
            ErrorMessage(pECB, ERR_NEW_ORDER_NOT_PROCESSED,
ERR_TYPE_DEADLOCK, NULL, iTermId, iSyncId);
        else
            ErrorMessage(pECB, ERR_NEW_ORDER_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
    }
    else
        if ( iError == ERR_BAD_ITEM_ID)
            WriteZString(pECB, MakeNewOrderForm(iTermId,
iSyncId, TRUE, FALSE, (BOOL)iRc) );
        else
            WriteZString(pECB, MakeNewOrderForm(iTermId,
iSyncId, FALSE, FALSE, (BOOL)iRc) );

```

```

        return;
    }
}
#else // Not LOCAL_ALLOC
static void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId)
{
    int TpRc, iRc, iError;
    long ilen, *olen;
    char buf[128];

    if((iRc = ThrTpInit()) <0)
    {
        // This is bad
        sprintf(buf, "ProcessNewOrder Failed ThrTpInit: iRc =
%d", iRc);
        LogTuxError(0, buf);
        ErrorMessage(pECB, ERR_NEW_ORDER_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }

    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;
    }

    Term.pClientData[iTermId].TuxDataPtr->NewOrderData =
Term.pClientData[iTermId].NewOrderData;

    ilen = sizeof(TUX_DATA);
    olen = &ilen;

    if ( dLog )
    {
        FILE *fp;

        fp = fopen(szTpccLogPath, "ab");
        fprintf(fp, "*** ProcessNewOrder Thread %d iTermId %d
TuxDataPtr: %x size %d \r\n",
        GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].TuxDataPtr,
sizeof(*Term.pClientData[iTermId].TuxDataPtr));
        fclose(fp);
    }
}

```

```

    }

    if ((iRc = tpcall("NEWORDER", (char
*)Term.pClientData[iTermId].TuxDataPtr, ilen,
    (char **)&Term.pClientData[iTermId].TuxDataPtr, (long
*)olen, TPSIGRSTRT)) == -1)
    {
        TpRc = tperrno;
        sprintf(buf, "Neworder tpcall failed");
        LogTuxError(TpRc, buf);
        ErrorMessage(pECB, ERR_NEW_ORDER_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }

    if ( dLog )
    {
        FILE *fp;

        fp = fopen(szTpccLogPath, "ab");
        fprintf(fp, "*** ProcessNewOrder Thread %d iTermId %d
TuxDataPtr: %x size %d\r\n",
            GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].TuxDataPtr,
            sizeof(*Term.pClientData[iTermId].TuxDataPtr));
        fclose(fp);
    }

    Term.pClientData[iTermId].NewOrderData =
Term.pClientData[iTermId].TuxDataPtr->NewOrderData;

    iRc = Term.pClientData[iTermId].TuxDataPtr->NewOrderData.retval;
    iError = Term.pClientData[iTermId].TuxDataPtr->NewOrderData.error;

    if ( iRc < 0 )
    {
        if (iError == ERR_TYPE_DEADLOCK)
            ErrorMessage(pECB, ERR_NEW_ORDER_NOT_PROCESSED,
ERR_TYPE_DEADLOCK, NULL, iTermId, iSyncId);
        else
            ErrorMessage(pECB, ERR_NEW_ORDER_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
    }
    else
        if ( iError == ERR_BAD_ITEM_ID)
            WriteZString(pECB, MakeNewOrderForm(iTermId,
iSyncId, TRUE, FALSE, (BOOL)iRc) );
        else
            WriteZString(pECB, MakeNewOrderForm(iTermId,
iSyncId, FALSE, FALSE, (BOOL)iRc) );

```

```

        return;
    }
#endif // LOCAL_ALLOC

/* 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
*/

#ifdef LOCAL_ALLOC // Allocate the tmalloc structure for each transaction
// This saves on some memory at the expense of some CPU cycles.
static void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId)
{
    int TpRc, iRc, iError;
    long ilen, *olen;
    char buf[128];

    PAYMENT_DATA *PaymentDataPtr; //Payment Tuxedo
Buffer

    if((iRc = ThrTpInit()) <0)
    {
        // This is bad
        sprintf(buf, "ProcessPayment Failed ThrTpInit: iRc = %d",
iRc);
        LogTuxError(0, buf);
        ErrorMessage(pECB, ERR_PAYMENT_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }

    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;
}

if ((PaymentDataPtr = (PAYMENT_DATA *)tpalloc("CARRAY", NULL,
sizeof(PAYMENT_DATA))) == NULL)
{
    TpRc = tperrno;
    sprintf(buf, "ProcessPayment Tpcalloc Failed");
    LogTuxError(TpRc, buf);
    ErrorMessage(pECB, ERR_PAYMENT_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
    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);
}

if (( iRc = tpcall("PAYMENT", (char *)PaymentDataPtr, ilen,
(char **)&PaymentDataPtr, (long *)olen, TPSIGRSTRT)) == -
1)
{
    TpRc = tperrno;
    sprintf(buf, "ProcessPayment tpcall failed");
    LogTuxError(TpRc, buf);
    ErrorMessage(pECB, ERR_PAYMENT_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
    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;
iError = PaymentDataPtr->error;

tpfree((char *)PaymentDataPtr);

if ( iRc < 0 )
{
    if (iError == ERR_TYPE_DEADLOCK )
        ErrorMessage(pECB, ERR_PAYMENT_NOT_PROCESSED,
ERR_TYPE_DEADLOCK, NULL, iTermId, iSyncId);
    else if (iError == ERR_NOSUCH_CUSTOMER)
        ErrorMessage(pECB, ERR_PAYMENT_INVALID_CUSTOMER,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
    else
        ErrorMessage(pECB, ERR_PAYMENT_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
}
else
    WriteZString(pECB, MakePaymentForm(iTermId, iSyncId, FALSE) );

return;
}
#else // Not LOCAL_ALLOC
static void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId)
{
    int          TpRc, iRc, iError;
    long         ilen, *olen;
    char         buf[128];

    if((iRc = ThrTpInit()) <0)
    {
        // This is bad
        sprintf(buf, "ProcessPayment Failed ThrTpInit: iRc = %d",
iRc);

        LogTuxError(0, buf);
        ErrorMessage(pECB, ERR_PAYMENT_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }
}

```

```

        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;
    }

    Term.pClientData[iTermId].TuxDataPtr->PaymentData =
Term.pClientData[iTermId].PaymentData;

    ilen = sizeof(TUX_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].TuxDataPtr);
        fclose(fp);
    }

    if (( iRc = tpcall("PAYMENT", (char
*)Term.pClientData[iTermId].TuxDataPtr, ilen,
(char **)&Term.pClientData[iTermId].TuxDataPtr, (long
*)olen, TPSIGRSTRT)) == -1)
    {
        TpRc = tperrno;
        sprintf(buf, "ProcessPayment tpcall failed");
        LogTuxError(TpRc, buf);
        ErrorMessage(pECB, ERR_PAYMENT_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }

    if ( dLog )
    {
        FILE *fp;

        fp = fopen(szTpccLogPath, "ab");
        fprintf(fp, "*** ProcessPayment Thread %d iTermId %d
TuxDataPtr: %x \r\n",

```

```

                GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].TuxDataPtr);
        fclose(fp);
    }

    Term.pClientData[iTermId].PaymentData =
Term.pClientData[iTermId].TuxDataPtr->PaymentData;

    iRc = Term.pClientData[iTermId].TuxDataPtr->PaymentData.retval;
    iError = Term.pClientData[iTermId].TuxDataPtr->PaymentData.error;

    if ( iRc < 0 )
    {
        if (iError == ERR_TYPE_DEADLOCK )
            ErrorMessage(pECB, ERR_PAYMENT_NOT_PROCESSED,
ERR_TYPE_DEADLOCK, NULL, iTermId, iSyncId);
        else if (iError == ERR_NOSUCH_CUSTOMER)
            ErrorMessage(pECB, ERR_PAYMENT_INVALID_CUSTOMER,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        else
            ErrorMessage(pECB, ERR_PAYMENT_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
    }
    else
        WriteZString(pECB, MakePaymentForm(iTermId, iSyncId, FALSE) );

    return;
}
#endif // LOCAL_ALLOC

/* 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.
* iTermId client browser terminal id
* iSyncId client browser sync id
*
* RETURNS: None
*
* COMMENTS: None
*/

```



```

#ifdef LOCAL_ALLOC // Allocate the tmalloc structure for each transaction
// This saves on some memory at the expense of some CPU cycles.
static void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId)
{
    int          TpRc, iRc, iError;
    long         ilen, *olen;
    char        buf[128];

    ORDER_STATUS_DATA      *OrderStatusDataPtr; //Order Status Tuxedo
Buffer

    if((iRc = ThrTpInit()) <0)
    {
        // This is bad
        sprintf(buf, "ProcessOrderStatus Failed ThrTpInit: iRc =
%d", iRc);
        LogTuxError(0, buf);
        ErrorMessage(pECB, ERR_ORDER_STATUS_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }

    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;
    }

    if ((OrderStatusDataPtr = (ORDER_STATUS_DATA *)tpalloc("CARRAY",
NULL, sizeof(ORDER_STATUS_DATA))) == NULL)
    {
        TpRc = tperrno;
        sprintf(buf, "ProcessOrderStatus Tpcalloc Failed");
        LogTuxError(TpRc, buf);
        ErrorMessage(pECB, ERR_ORDER_STATUS_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        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);
        }

        if ( ( iRc = tpcall("ORDERSTATUS", (char *)OrderStatusDataPtr, ilen,
(char **)&OrderStatusDataPtr, (long *)olen, TPSIGRSTRT)
== -1)
        {
            TpRc = tperrno;
            sprintf(buf, "ProcessOrderStatus tpcall failed");
            LogTuxError(TpRc, buf);
            ErrorMessage(pECB, ERR_ORDER_STATUS_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
            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;
        iError = OrderStatusDataPtr->error;

        tpfree((char *)OrderStatusDataPtr);

        if ( iRc < 0 )
        {
            if ( iError == ERR_TYPE_DEADLOCK )
                ErrorMessage(pECB, ERR_ORDER_STATUS_NOT_PROCESSED,
ERR_TYPE_DEADLOCK, NULL, iTermId, iSyncId);
            else if (iError == ERR_NOSUCH_CUSTOMER)
                ErrorMessage(pECB, ERR_NOSUCH_CUSTOMER,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
            else

```

```

        ErrorMessage(pECB, ERR_ORDER_STATUS_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
    }
    else
        WriteZString(pECB, MakeOrderStatusForm(iTermId, iSyncId, FALSE)
);

    return;
}
#else // Not LOCAL_ALLOC
static void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId)
{
    int          TpRc, iRc, iError;
    long         ilen, *olen;
    char         buf[128];

    if((iRc = ThrTpInit()) <0)
    {
        // This is bad
        sprintf(buf, "ProcessOrderStatus Failed ThrTpInit: iRc =
%d", iRc);
        LogTuxError(0, buf);
        ErrorMessage(pECB, ERR_ORDER_STATUS_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }

    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;
    }

    Term.pClientData[iTermId].TuxDataPtr->OrderStatusData =
Term.pClientData[iTermId].OrderStatusData;

    ilen = sizeof(TUX_DATA);
    olen = &ilen;

    if ( dLog )
    {
        FILE *fp;

        fp = fopen(szTpccLogPath, "ab");

```

```

        fprintf(fp, "** ProcessOrderStatus Thread %d iTermId %d
TuxDataPtr: %x \r\n",
                GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].TuxDataPtr);
        fclose(fp);
    }

    if (( iRc = tpcall("ORDERSTATUS", (char
*)Term.pClientData[iTermId].TuxDataPtr, ilen,
                (char **)&Term.pClientData[iTermId].TuxDataPtr, (long
*)olen, TPSIGRSTRT)) == -1)
    {
        TpRc = tperno;
        sprintf(buf, "ProcessOrderStatus tpcall failed");
        LogTuxError(TpRc, buf);
        ErrorMessage(pECB, ERR_ORDER_STATUS_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }

    if ( dLog )
    {
        FILE *fp;

        fp = fopen(szTpccLogPath, "ab");
        fprintf(fp, "*** ProcessOrderStatus Thread %d iTermId %d
TuxDataPtr: %x \r\n",
                GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].TuxDataPtr);
        fclose(fp);
    }

    Term.pClientData[iTermId].OrderStatusData =
Term.pClientData[iTermId].TuxDataPtr->OrderStatusData;

    iRc = Term.pClientData[iTermId].TuxDataPtr->OrderStatusData.retval;
    iError = Term.pClientData[iTermId].TuxDataPtr-
>OrderStatusData.error;

    if ( iRc < 0 )
    {
        if ( iError == ERR_TYPE_DEADLOCK )
            ErrorMessage(pECB, ERR_ORDER_STATUS_NOT_PROCESSED,
ERR_TYPE_DEADLOCK, NULL, iTermId, iSyncId);
        else if (iError == ERR_NOSUCH_CUSTOMER)
            ErrorMessage(pECB, ERR_NOSUCH_CUSTOMER,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        else
            ErrorMessage(pECB, ERR_ORDER_STATUS_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
    }
    else

```

```

        WriteZString(pECB, MakeOrderStatusForm(iTermId, iSyncId, FALSE)
);
    return;
}
#endif // LOCAL_ALLOC
/* 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
*/
#ifdef LOCAL_ALLOC // Allocate the tmalloc structure for each transaction
// This saves on some memory at the expense of some CPU cycles.
static void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId)
{
    int TpRc, iRc;
    char szTmp[26];
    BOOL bSuccess;
    long ilen, *olen;
    char buf[128];

    DELIVERY_DATA *DeliveryDataPtr; //Delivery Tuxedo
Buffer

    if((iRc = ThrTpInit()) <0)
    {
        // This is bad
        sprintf(buf, "ProcessDelivery Failed ThrTpInit: iRc =
%d", iRc);
        LogTuxError(0, buf);
        ErrorMessage(pECB, ERR_DELIVERY_MISSING_OCD_KEY,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);

```

```

        return;
    }
    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);
    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 ((DeliveryDataPtr = (DELIVERY_DATA *)tmalloc("CARRAY", NULL,
sizeof(DELIVERY_DATA))) == NULL)
    {
        TpRc = tperno;
        sprintf(buf, "ProcessDelivery Tpcalloc Failed");
        LogTuxError(TpRc, buf);
        strcpy(Term.pClientData[iTermId].DeliveryData.execution_status,
"Delivery Post Failed");
        bSuccess = FALSE;
        WriteZString(pECB, MakeDeliveryForm(iTermId, iSyncId,
FALSE, bSuccess) );
        return;
    }
    *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, &DeliveryDataPtr);
        fclose(fp);
    }

    if (( iRc = tpacall("DELIVERY", (char *)DeliveryDataPtr, ilen,
TPNOREPLY)) == -1)
    {
        TpRc = tperrno;
        sprintf(buf, "ProcessDelivery Tpcalloc Failed");
        LogTuxError(TpRc, buf);

        strcpy(Term.pClientData[iTermId].DeliveryData.execution_status,
"Delivery Post Failed");
        bSuccess = FALSE;
        WriteZString(pECB, MakeDeliveryForm(iTermId, iSyncId,
FALSE, bSuccess) );
        return;
    }

    if ( dLog )
    {
        FILE *fp;

        fp = fopen(szTpccLogPath, "ab");
        fprintf(fp, "** ProcessDelivery Thread %d iTermId %d
DeliveryDataPtr: %x TpRc = %d \r\n",
                GetCurrentThreadId(), iTermId, &DeliveryDataPtr,
TpRc);
        fclose(fp);
    }

    tpfree((char *)DeliveryDataPtr);

    strcpy(Term.pClientData[iTermId].DeliveryData.execution_status,
"Delivery has been queued.");
    bSuccess = TRUE;

    WriteZString(pECB, MakeDeliveryForm(iTermId, iSyncId, FALSE,
bSuccess) );

    return;

```

```

}
#else // Not LOCAL_ALLOC
static void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId)
{
    int          TpRc, iRc;
    char        szTmp[26];
    BOOL        bSuccess;
    long        ilen, *olen;
    char        buf[128];

    if((iRc = ThrTpInit()) <0)
    {
        // This is bad
        sprintf(buf, "ProcessDelivery Failed ThrTpInit: iRc =
%d", iRc);
        LogTuxError(0, buf);
        ErrorMessage(pECB, ERR_DELIVERY_MISSING_OCD_KEY,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }

    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);

    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);

Term.pClientData[iTermId].TuxDataPtr->DeliveryData =
Term.pClientData[iTermId].DeliveryData;

ilen = sizeof(TUX_DATA);
olen = &ilen;

    if ( dLog )
    {
        FILE *fp;

        fp = fopen(szTpccLogPath, "ab");
        fprintf(fp, "** ProcessDelivery Thread %d iTermId %d
TuxDataPtr: %x \r\n",
                GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].TuxDataPtr);
        fclose(fp);
    }

    if (( iRc = tpacall("DELIVERY", (char
*)Term.pClientData[iTermId].TuxDataPtr, ilen, TPNOREPLY)) == -1)
    {
        TpRc = tperrno;
        sprintf(buf, "ProcessDelivery Tpcalloc Failed");
        LogTuxError(TpRc, buf);

        strcpy(Term.pClientData[iTermId].DeliveryData.execution_status,
"Delivery Post Failed");
        bSuccess = FALSE;
        WriteZString(pECB, MakeDeliveryForm(iTermId, iSyncId,
FALSE, bSuccess) );
        return;
    }

    if ( dLog )
    {
        FILE *fp;

        fp = fopen(szTpccLogPath, "ab");
        fprintf(fp, "** ProcessDelivery Thread %d iTermId %d
TuxDataPtr: %x TpRc = %d \r\n",
                GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].TuxDataPtr, TpRc);
        fclose(fp);
    }

    strcpy(Term.pClientData[iTermId].DeliveryData.execution_status,
"Delivery has been queued.");
    bSuccess = TRUE;

```

```

        WriteZString(pECB, MakeDeliveryForm(iTermId, iSyncId, FALSE,
bSuccess) );
        return;
    }
#endif // LOCAL_ALLOC

/* 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
*/

#ifdef LOCAL_ALLOC // Allocate the tmalloc structure for each transaction
// This saves on some memory at the expense of some CPU cycles.
static void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId)
{
    int TpRc, iRc, iError;
    char szTmp[26];
    long ilen, *olen;
    char buf[128];

    STOCK_LEVEL_DATA *StockLevelDataPtr; //Stock Level Tuxedo
Buffer

    if((iRc = ThrTpInit()) <0)
    {
        // This is bad
        sprintf(buf, "ProcessStockLevel Failed ThrTpInit: iRc =
%d", iRc);
        LogTuxError(0, buf);
        ErrorMessage(pECB, ERR_STOCKLEVEL_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }

```

```

    }

    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;
    }

    if ((StockLevelDataPtr = (STOCK_LEVEL_DATA *)tpalloc("CARRAY",
NULL, sizeof(STOCK_LEVEL_DATA))) == NULL)
    {
        TpRc = tperrno;
        sprintf(buf, "ProcessStockLevel Tpcalloc Failed");
        LogTuxError(TpRc, buf);
        ErrorMessage(pECB, ERR_STOCKLEVEL_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        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);
    }

    if (( iRc = tpcall("STOCKLEVEL", (char *)StockLevelDataPtr, ilen,
(char **)&StockLevelDataPtr, (long *) olen, TPSIGRSTRT))
== -1)
    {
        TpRc = tperrno;
        sprintf(buf, "ProcessStockLevel tpcall failed");
        LogTuxError(TpRc, buf);
        ErrorMessage(pECB, ERR_STOCKLEVEL_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        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;
    iError = StockLevelDataPtr->error;

    tpfree((char *)StockLevelDataPtr);

    if ( iRc == 0 )
    {
        if ( iError == ERR_TYPE_DEADLOCK )
            ErrorMessage(pECB, ERR_STOCKLEVEL_NOT_PROCESSED,
ERR_TYPE_DEADLOCK, NULL, iTermId, iSyncId);
        else
            ErrorMessage(pECB, ERR_STOCKLEVEL_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
    }
    else
        WriteZString(pECB, MakeStockLevelForm(iTermId, iSyncId, FALSE)
);

```

```

    return;
}
#else // Not LOCAL_ALLOC
static void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, int iSyncId)
{
    int          TpRc, iRc, iError;
    char        szTmp[26];
    long        ilen, *olen;
    char        buf[128];

    if((iRc = ThrTpInit()) <0)
    {
        // This is bad
        sprintf(buf, "ProcessStockLevel Failed ThrTpInit: iRc =
%d", iRc);
        LogTuxError(0, buf);
        ErrorMessage(pECB, ERR_STOCKLEVEL_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }

    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;
}

Term.pClientData[iTermId].TuxDataPtr->StockLevelData =
Term.pClientData[iTermId].StockLevelData;

    ilen = sizeof(TUX_DATA);
    olen = &ilen;

    if ( dLog )
    {
        FILE *fp;

        fp = fopen(szTpccLogPath, "ab");
        fprintf(fp, "** ProcessStockLevel Thread %d iTermId %d
TuxDataPtr: %x \r\n",
                GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].TuxDataPtr);
        fclose(fp);
    }

    if (( iRc = tpcall("STOCKLEVEL", (char
*)Term.pClientData[iTermId].TuxDataPtr, ilen,
(char **)&Term.pClientData[iTermId].TuxDataPtr, (long *)
olen, TPSIGRSTRT)) == -1)
    {
        TpRc = tperrno;
        sprintf(buf, "ProcessStockLevel tpcall failed");
        LogTuxError(TpRc, buf);
        ErrorMessage(pECB, ERR_STOCKLEVEL_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        return;
    }

    if ( dLog )
    {
        FILE *fp;

        fp = fopen(szTpccLogPath, "ab");
        fprintf(fp, "*** ProcessStockLevel Thread %d iTermId %d
TuxDataPtr: %x \r\n",
                GetCurrentThreadId(), iTermId,
&Term.pClientData[iTermId].TuxDataPtr);
        fclose(fp);
    }

    Term.pClientData[iTermId].StockLevelData =
Term.pClientData[iTermId].TuxDataPtr->StockLevelData;

    iRc = Term.pClientData[iTermId].TuxDataPtr->StockLevelData.retval;
    iError = Term.pClientData[iTermId].TuxDataPtr-
>StockLevelData.error;

```

```

        if ( iRc == 0 )
        {
            if ( iError == ERR_TYPE_DEADLOCK )
                ErrorMessage(pECB, ERR_STOCKLEVEL_NOT_PROCESSED,
ERR_TYPE_DEADLOCK, NULL, iTermId, iSyncId);
            else
                ErrorMessage(pECB, ERR_STOCKLEVEL_NOT_PROCESSED,
ERR_TYPE_WEBDLL, NULL, iTermId, iSyncId);
        }
        else
            WriteZString(pECB, MakeStockLevelForm(iTermId, iSyncId, FALSE)
);

        return;
    }
#endif // LOCAL_ALLOC

/* 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];
    char szKey[26];
    int i;
    short items;
    BOOL bCheck;

    if ( !GetKeyValue(lpszQueryString, "DID*", szTmp, sizeof(szTmp)) )
        return ERR_NEWORDER_FORM_MISSING_DID;

    if ( !IsNumeric(szTmp) )
        return ERR_NEWORDER_DISTRICT_INVALID;

```

```

pNewOrderData->d_id = atoi(szTmp);

    if ( !GetKeyValue(lpszQueryString, "CID*", szTmp, sizeof(szTmp)) )
        return ERR_NEWORDER_CUSTOMER_KEY;

    if ( !IsNumeric(szTmp) )
        return ERR_NEWORDER_CUSTOMER_INVALID;
    pNewOrderData->c_id = atoi(szTmp);

    bCheck = FALSE;
    for(i=0, items=0; i<15; i++)
    {
        wsprintf(szKey, "IID%2.2d*", i);
        if ( !GetKeyValue(lpszQueryString, szKey, szTmp,
sizeof(szTmp)) )
            return ERR_NEWORDER_MISSING_IID_KEY;
        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);

            wsprintf(szKey, "SP%2.2d*", i);
            if ( !GetKeyValue(lpszQueryString, szKey, szTmp,
sizeof(szTmp)) )
                return ERR_NEWORDER_MISSING_SUPPW_KEY;
            // ETW Fix for warehouse out of range
            if ( !IsNumeric(szTmp) )
                return ERR_NEWORDER_SUPPW_INVALID;
            if ( (short)atoi(szTmp) > iMaxWareHouses )
                return ERR_NEWORDER_SUPPW_RANGE;
            pNewOrderData->Ol[i].ol_supply_w_id =
(short)atoi(szTmp);

            wsprintf(szKey, "Qty%2.2d*", i);
            if ( !GetKeyValue(lpszQueryString, szKey, szTmp,
sizeof(szTmp)) )
                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
    {
        wsprintf(szKey, "SP%2.2d*", i);
        if ( !GetKeyValue(lpszQueryString, szKey, szTmp,
sizeof(szTmp)) )
            return ERR_NEWORDER_MISSING_QTY_KEY;

        if ( szTmp[0] )
            return ERR_NEWORDER_SUPPW_WITHOUT_ITEMID;

        wsprintf(szKey, "Qty%2.2d*", i);
        if ( !GetKeyValue(lpszQueryString, szKey, szTmp,
sizeof(szTmp)) )
            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;

    if ( !GetKeyValue(lpszQueryString, "DID*", szTmp, sizeof(szTmp)) )
        return ERR_PAYMENT_MISSING_DID_KEY;
    if ( !IsNumeric(szTmp) )
        return ERR_PAYMENT_DISTRICT_INVALID;
    pPaymentData->d_id = atoi(szTmp);

    if ( !GetKeyValue(lpszQueryString, "CID*", szTmp, sizeof(szTmp)) )
        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 )
    {
        if ( !GetKeyValue(lpszQueryString, "CLT*", szTmp,
sizeof(szTmp)) )
            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
    {
        if ( !GetKeyValue(lpszQueryString, "CLT*", szTmp,
sizeof(szTmp)) )
            return ERR_PAYMENT_MISSING_CLT_KEY;
        if ( szTmp[0] )
            return ERR_PAYMENT_CID_AND_CLT;
    }

    if ( !GetKeyValue(lpszQueryString, "CDI*", szTmp, sizeof(szTmp)) )
        return ERR_PAYMENT_MISSING_CDI_KEY;
    if ( !IsNumeric(szTmp) )
        return ERR_PAYMENT_CDI_INVALID;
    pPaymentData->c_d_id = atoi(szTmp);

    if ( !GetKeyValue(lpszQueryString, "CWI*", szTmp, sizeof(szTmp)) )
        return ERR_PAYMENT_MISSING_CWI_KEY;

    if ( !IsNumeric(szTmp) )
        return ERR_PAYMENT_CWI_INVALID;
}

```

```

pPaymentData->c_w_id = atoi(szTmp);

if ( !GetKeyValue(lpszQueryString, "HAM*", szTmp, sizeof(szTmp)) )
    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];

    if ( !GetKeyValue(lpszQueryString, "DID*", szTmp, sizeof(szTmp)) )
        return ERR_ORDERSTATUS_MISSING_DID_KEY;
    if ( !IsNumeric(szTmp) )
        return ERR_ORDERSTATUS_DID_INVALID;
    pOrderStatusData->d_id = atoi(szTmp);

    if ( !GetKeyValue(lpszQueryString, "CID*", szTmp, sizeof(szTmp)) )
        return ERR_ORDERSTATUS_MISSING_CID_KEY;

    if ( szTmp[0] == 0 )
    {
        pOrderStatusData->c_id = 0;
        if ( !GetKeyValue(lpszQueryString, "CLT*", szTmp,
sizeof(szTmp)) )
            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);
        if ( !GetKeyValue(lpszQueryString, "CLT*", szTmp,
sizeof(szTmp)) )
            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 = 500;
    iThreads      = 5;
    iQSlotts     = 3000;
    iDelayMs      = 100;
    iDeadlockRetry = (short)3;
    strcpy(szTpccLogPath, "tpcclog.");
    strcpy(szErrorLogPath, "tpccerr.");

    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 ( !strcmp(szTmp, "ON") )
            bLog = TRUE;
    }

    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey, "DEBUG", 0, &type, szTmp, &size) ==
ERROR_SUCCESS )
    {
        if ( !strcmp(szTmp, "ON") )

```

```

        dLog = TRUE;
    }

    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey, "MaximumWarehouses", 0, &type, szTmp,
&size) == ERROR_SUCCESS )
    {
        iMaxWareHouses = atoi(szTmp);
        if ( iMaxWareHouses == 0 )
            iMaxWareHouses = 500;
    }

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

    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 );
}

/* 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()
{
    static int num_tpinit=0;
    static int x=1;
    static int once=0;
    static CRITICAL_SECTION    TpCriticalSection;
    int lasterr, iRc, TpRc;
    int retry = 0;
    BOOL Success = FALSE;

    if (!TlsGetValue(TLSIsTpInitKey))
    {
        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)
                {
                    LeaveCriticalSection(&TpCriticalSection);
                    TpRc = tperrno;
                    {
                        FILE *fp;

                        fp = fopen(szErrorLogPath,
"ab");

                        fprintf(fp, ">>>>
ThrTpInit:%d : tmalloc of tpinit failed: %d : %s\r\n",
GetCurrentThreadId(),
TpRc, tpsterror(TpRc));

                        fclose(fp);
                    }
                }
            }
        }
    }
}

```

```

        }
        retry++;
        continue;
    }
    tpinf->flags|=TPMULTICONTEXTS;
}

if (retry == 0)
    itoa(++num_tpinit, tpinf->cltname, 10);

iRc = tpinit(tpinf);
TpRc = tperrno;

// check tmalloc() ?
if (iRc < 0)
{
    LeaveCriticalSection(&TpCriticalSection);
    retry++;
    lasterr = GetLastError();
    TpRc = tperrno;
    {
        FILE *fp;

        fp = fopen(szErrorLogPath, "ab");
        fprintf(fp, ">>>> ThrTpInit:%d :
GetCurrentThreadId(), iRc,
tpsterror(TpRc), retry);

        fclose(fp);
    }
}
else
{
    Success = TRUE;
    LeaveCriticalSection(&TpCriticalSection);
    break;
}

Sleep(50); // Relinquish thread timeslice
} // retry the tpinit if it failed the first time

if ( Success == FALSE )
{
    char ebuf[128];

    sprintf(ebuf, ">>>> ThrTpInit %d : Cannot
tpinit after %d tries iRc = %d LastErr = %d \r\n",
GetCurrentThreadId(), TP_MAX_RETRIES, iRc,
lasterr);

    LogTuxError(TpRc, ebuf);
}
}

```

```

    }
    return -1;
}
if ( Success == TRUE )
{
    if ( retry > 0 )
    {
        char ebuf[128];

        sprintf(ebuf, ">>>> ThrTpInit %d : Cannot
tpinit after %d tries iRc = %d LastErr = %d \r\n",
            GetCurrentThreadId(), TP_MAX_RETRIES, iRc,
lasterr);

        sprintf(ebuf, "* ThrTpInit Thread %d
Success retry count %d with LastErr = %d * \r\n",
            GetCurrentThreadId(), retry,
lasterr);

        LogTuxError(TpRc, ebuf);
    }

    if ( ( iRc=TlsSetValue(TLSIsTpInitiatedKey,&x)) == 0 )
    {
        {
            FILE *fp;

            fp = fopen(szErrorLogPath, "ab");
            fprintf(fp, ">>>> ThrTpInit %d :
TlsSetValue Failed iRc: %d \r\n",
                GetCurrentThreadId(), iRc);
            fclose(fp);
        }
    }
}
}
else
{
    if ( dLog )
    {
        FILE *fp;

        fp = fopen(szTpccLogPath, "ab");
        fprintf(fp, "* ThrTpInit Thread %d already tpinited
* \r\n", GetCurrentThreadId());
        fclose(fp);
    }
}
return 0;
}
}

LIBRARY TPCC.DLL

```

```

EXPORTS

    GetExtensionVersion    @1
    HttpExtensionProc      @2

/*    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

#ifndef 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
    DBDATAREC o_entry_d;
#endif
    short o_all_local;
    double total_amount;

```

```

    long num_deadlocks;
    int retval;
    int error;
    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
    DBDATAREC 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
    DBDATAREC 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;
        int                error;
        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
    DBDATAREC          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
    DBDATAREC          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;
    int                 error;
    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;
        int               error;
        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;
    int                 error;
    char
execution_status[STATUS_LEN];
} STOCK_LEVEL_DATA;

typedef struct
{
    NEW_ORDER_DATA      NewOrderData;           //new order
    PAYMENT_DATA        PaymentData;           //payment
    ORDER_STATUS_DATA   OrderStatusData;       //order
    DELIVERY_DATA       DeliveryData;          //delivery
    STOCK_LEVEL_DATA    StockLevelData;
    //stock level form data
} TUX_DATA;
#endif

```


Appendix B - Database Details

```
-- File:      BACKUP.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.00
--           Copyright Microsoft, 1996
-- Purpose:   Creates backup of tpcc database
```

```
declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)
```

```
dump database tpcc to tpccback1, tpccback2 with init, stats = 1
```

```
select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate,
@enddate)
```

```
go
```

```
-- 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'
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="D:\MSSQL7\DATA\tpcc_root.mdf",
size=10MB, FILEGROWTH=0)
```

```
log on
    (name="MSSQL70_tpcc_log",filename="L:",size=45000MB, FILEGROWTH=0)
```

```
-- create filegroups
```

```
alter database tpcc add filegroup MSSQL70_cs_fg
alter database tpcc add filegroup MSSQL70_misc_fg
```

```
-- add files to filegroups
```

```
alter database tpcc add file
    (name="MSSQL70_cs1",filename="H:",size=17950MB, FILEGROWTH=0),
    (name="MSSQL70_cs2",filename="I:",size=17950MB, FILEGROWTH=0),
    (name="MSSQL70_cs3",filename="J:",size=17950MB, FILEGROWTH=0),
    (name="MSSQL70_cs4",filename="K:",size=17950MB, FILEGROWTH=0),
    (name="MSSQL70_cs5",filename="M:",size=17950MB, FILEGROWTH=0),
    (name="MSSQL70_cs6",filename="N:",size=17950MB, FILEGROWTH=0)
to filegroup MSSQL70_cs_fg
```

```
alter database tpcc add file
    (name="MSSQL70_misc1",filename="O:",size=9500MB, FILEGROWTH=0),
    (name="MSSQL70_misc2",filename="P:",size=9500MB, FILEGROWTH=0),
    (name="MSSQL70_misc3",filename="Q:",size=9500MB, FILEGROWTH=0),
    (name="MSSQL70_misc4",filename="R:",size=9500MB, FILEGROWTH=0),
    (name="MSSQL70_misc5",filename="S:",size=9500MB, FILEGROWTH=0),
    (name="MSSQL70_misc6",filename="T:",size=9500MB, FILEGROWTH=0)
to filegroup MSSQL70_misc_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','X:\tpccback1.dmp'
exec sp_addumpdevice 'disk','tpccback2','Y:\tpccback2.dmp'
go
```

```
-- 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

-- File:      DBOPT2.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.00
--           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 'district','AllowPageLocks',FALSE
go
sp_indexoption 'warehouse','AllowPageLocks',FALSE

```

```

go
sp_indexoption 'stock','AllowPageLocks',FALSE
go
sp_indexoption 'order_line','AllowPageLocks',FALSE
go
sp_indexoption 'orders','AllowPageLocks',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-pecified 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

```

```

-- 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 uplock)
        where no_w_id = @w_id and
              no_d_id = @d_id
        order by no_o_id asc

        if (@@rowcount <> 0)
        begin

```

```

-- claim the order for this district

        delete new_order
        where no_w_id = @w_id and
              no_d_id = @d_id and
              no_o_id = @o_id

-- set carrier_id on this order (and get customer id)

        update orders
        set o_carrier_id = @o_carrier_id,
            @c_id         = o_c_id
        where o_w_id = @w_id and
              o_d_id = @d_id and
              o_id      = @o_id

-- set date in all lineitems for this order (and sum amounts)

        update order_line
        set ol_delivery_d = getdate(),
            @total         = @total + ol_amount
        where ol_w_id = @w_id and
              ol_d_id = @d_id and
              ol_o_id = @o_id

-- accumulate lineitem amounts for this order into customer

        update customer
        set c_balance      = c_balance + @total,
            c_delivery_cnt = c_delivery_cnt + 1

        where c_w_id = @w_id and
              c_d_id = @d_id and
              c_id    = @c_id

    end

    select @oid1 = case @d_id when 1 then @o_id else @oid1 end,
           @oid2 = case @d_id when 2 then @o_id else @oid2 end,
           @oid3 = case @d_id when 3 then @o_id else @oid3 end,
           @oid4 = case @d_id when 4 then @o_id else @oid4 end,
           @oid5 = case @d_id when 5 then @o_id else @oid5 end,
           @oid6 = case @d_id when 6 then @o_id else @oid6 end,
           @oid7 = case @d_id when 7 then @o_id else @oid7 end,
           @oid8 = case @d_id when 8 then @o_id else @oid8 end,
           @oid9 = case @d_id when 9 then @o_id else @oid9 end,
           @oid10 = case @d_id when 10 then @o_id else @oid10 end

    end

commit tran d

-- return delivery data to client

```

```

select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

go

-- 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_cl' )
           drop index customer.customer_cl

create unique clustered index customer_cl on customer(c_w_id, c_d_id,
c_id)
           on MSSQL70_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate,
@enddate)

go

-- 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 MSSQL70_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate,
@enddate)

go

-- 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_cl' )
           drop index district.district_cl

create unique clustered index district_cl on district(d_w_id, d_id)
           with fillfactor=100 on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate,
@enddate)

go

-- 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 MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate,
@enddate)

go

-- 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 MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate,
@enddate)

go

-- 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 MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate,
@enddate)

go

-- 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 MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate,
@enddate)

```

go

```
-- 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 MSSQL70_misc_fg
```

```
select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate,
@enddate)
```

go

```
-- 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 MSSQL70_cs_fg
```

```
select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate,
@enddate)
```

go

```
-- 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 MSSQL70_misc_fg
```

```
select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate,
@enddate)
```

go

```
-- 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

smallint,

tinyint,

tinyint,

tinyint,

@s_w_id1 smallint = 0, @ol_qty1 smallint = 0,
@s_w_id2 smallint = 0, @ol_qty2 smallint = 0,
@s_w_id3 smallint = 0, @ol_qty3 smallint = 0,
@s_w_id4 smallint = 0, @ol_qty4 smallint = 0,
@s_w_id5 smallint = 0, @ol_qty5 smallint = 0,
@s_w_id6 smallint = 0, @ol_qty6 smallint = 0,
@s_w_id7 smallint = 0, @ol_qty7 smallint = 0,
@s_w_id8 smallint = 0, @ol_qty8 smallint = 0,
@s_w_id9 smallint = 0, @ol_qty9 smallint = 0,
@s_w_id10 smallint = 0, @ol_qty10 smallint = 0,
@s_w_id11 smallint = 0, @ol_qty11 smallint = 0,
@s_w_id12 smallint = 0, @ol_qty12 smallint = 0,
@s_w_id13 smallint = 0, @ol_qty13 smallint = 0,
@s_w_id14 smallint = 0, @ol_qty14 smallint = 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,

        @w_id          int,
        @d_id          int,
        @c_id          int,
        @o_ol_cnt      int,
        @o_all_local   int,
        @i_id1         int = 0,
        @i_id2         int = 0,
        @i_id3         int = 0,
        @i_id4         int = 0,
        @i_id5         int = 0,
        @i_id6         int = 0,
        @i_id7         int = 0,
        @i_id8         int = 0,
        @i_id9         int = 0,
        @i_id10        int = 0,
        @i_id11        int = 0,
        @i_id12        int = 0,
        @i_id13        int = 0,
        @i_id14        int = 0,
        @i_id15        int = 0,

as

```

```

        @remote_flag   int,
        @s_quantity    smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no         int,
        @o_id          int,
        @commit_flag   tinyint,
        @li_id         int,
        @li_s_w_id     smallint,
        @li_qty        smallint,
        @ol_number     int,
        @c_id_local    int

begin

    begin transaction n

    -- get district tax and next available order id and update
    -- plus initialize local variables

    update district
    set @d_tax      = d_tax,
        @o_id       = d_next_o_id,
        d_next_o_id = d_next_o_id + 1,
        @o_entry_d = getdate(),
        @li_no     = 0,
        @commit_flag = 1
    where d_w_id = @w_id and
        d_id   = @d_id

    -- process orderlines

    while (@li_no < @o_ol_cnt)
    begin

        select @li_no = @li_no + 1

    -- set i_id, s_w_id, and qty for this lineitem

        select @li_id = case @li_no
            when 1 then @i_id1
            when 2 then @i_id2
            when 3 then @i_id3
            when 4 then @i_id4
            when 5 then @i_id5
            when 6 then @i_id6
            when 7 then @i_id7
            when 8 then @i_id8
            when 9 then @i_id9
            when 10 then @i_id10
            when 11 then @i_id11
            when 12 then @i_id12
            when 13 then @i_id13

```

```

        when 14 then @i_id14
        when 15 then @i_id15
        end,

        @li_s_w_id = case @li_no
        when 1 then @s_w_id1
        when 2 then @s_w_id2
        when 3 then @s_w_id3
        when 4 then @s_w_id4
        when 5 then @s_w_id5
        when 6 then @s_w_id6
        when 7 then @s_w_id7
        when 8 then @s_w_id8
        when 9 then @s_w_id9
        when 10 then @s_w_id10
        when 11 then @s_w_id11
        when 12 then @s_w_id12
        when 13 then @s_w_id13
        when 14 then @s_w_id14
        when 15 then @s_w_id15
        end,

        @li_qty = case @li_no
        when 1 then @ol_qty1
        when 2 then @ol_qty2
        when 3 then @ol_qty3
        when 4 then @ol_qty4
        when 5 then @ol_qty5
        when 6 then @ol_qty6
        when 7 then @ol_qty7
        when 8 then @ol_qty8
        when 9 then @ol_qty9
        when 10 then @ol_qty10
        when 11 then @ol_qty11
        when 12 then @ol_qty12
        when 13 then @ol_qty13
        when 14 then @ol_qty14
        when 15 then @ol_qty15
        end

-- get item data (no one updates item)

        select @i_price = i_price,
               @i_name = i_name,
               @i_data = i_data
        from item (tablock repeatableread)
        where i_id = @li_id

-- 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

-- 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          int,
                             @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

-- 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,

```

```

""
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
    @c_last char(16) = -- 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

-- File:      RESTORE.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.00
--           Copyright Microsoft, 1996
-- Purpose:   Loads database backup from backup files

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

load database tpcc from tpccback1, tpccback2 with stats = 1

```

```

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate,
@enddate)

go

-- 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,
                              @threshold     smallint

as

    declare @o_id_low int,
            @o_id_high int

    select @o_id_low = (d_next_o_id - 20),
           @o_id_high = (d_next_o_id - 1)
    from district
    where d_w_id = @w_id and
          d_id = @d_id

    select count(distinct(s_i_id))
           from stock, order_line
    where ol_w_id = @w_id and
          ol_d_id = @d_id and
          ol_o_id between @o_id_low and @o_id_high and
          s_w_id = ol_w_id and
          s_i_id = ol_i_id and
          s_quantity < @threshold

go

-- 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 MSSQL70_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 MSSQL70_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 MSSQL70_cs_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 MSSQL70_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 MSSQL70_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 MSSQL70_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 MSSQL70_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 MSSQL70_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 MSSQL70_cs_fg  
go
```


Appendix C - Tunable Parameters and Options

This section discloses the Windows NT 4.0 Enterprise Edition registry parameters used on the Primergy 870-40 server system.

***** NT registry *****

Key Name: HARDWARE
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:31 PM

Key Name: HARDWARE\DESCRIPTION
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:31 PM

Key Name: HARDWARE\DESCRIPTION\System
 Class Name: System
 Last Write Time: 3/9/99 - 12:31 PM

Value 0
 Name: Component Information
 Type: REG_BINARY
 Data:
 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00

Value 1
 Name: Configuration Data
 Type: REG_FULL_RESOURCE_DESCRIPTOR
 Interface Type: Invalid
 Bus Number: -1
 Version: 0
 Revision: 0
 Partial Descriptor 0
 Resource: Device Specific
 Disposition: Undetermined
 Reserved1: 0x00000000
 Reserved2: 0x00000000
 Data:
 00000000 80 00 ff 03 00 00 3f 00 - fe 00 01 00
?.....

Value 2
 Name: Identifier
 Type: REG_SZ

Data: AT/AT COMPATIBLE

Value 3
 Name: SystemBiosDate
 Type: REG_SZ
 Data: 02/12/99

Value 4
 Name: SystemBiosVersion
 Type: REG_MULTI_SZ
 Data: PhoenixBIOS Version 4.06 Rev. 2.00d.1998
 PhoenixBIOS Version 4.06 Rev. 2.00d.1998

Value 5
 Name: VideoBiosDate
 Type: REG_SZ
 Data: 05/21/97

Value 6
 Name: VideoBiosVersion
 Type: REG_MULTI_SZ
 Data: CL-GD5446 PCI VGA BIOS Version 1.33
 Rel. 1.00

S Version 4.06 Rev. 2.00d.1998

Key Name: HARDWARE\DESCRIPTION\System\CentralProcessor
Class Name: Processor
Last Write Time: 3/9/99 - 12:31 PM

Key Name: HARDWARE\DESCRIPTION\System\CentralProcessor\0
Class Name: Processor
Last Write Time: 3/9/99 - 12:32 PM

Value 0
Name: Component Information
Type: REG_BINARY
Data:
00000000 00 00 00 00 00 00 00 00 00 - 00 00 00 00 01 00 00 00
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Invalid

Bus Number: -1
Version: 0
Revision: 0

Value 2
Name: FeatureSet
Type: REG_DWORD
Data: 0x3ff

Value 3
Name: Identifier
Type: REG_SZ
Data: x86 Family 6 Model 7 Stepping 2

Value 4
Name: Update Signature
Type: REG_BINARY
Data:
00000000 00 00 00 00 20 00 00 00 -

Value 5
Name: Update Status
Type: REG_DWORD
Data: 0x2

Value 6
Name: VendorIdentifier
Type: REG_SZ
Data: GenuineIntel

Value 7
Name: ~MHz
Type: REG_DWORD
Data: 0x1f4

Key Name: HARDWARE\DESCRIPTION\System\CentralProcessor\1
Class Name: Processor
Last Write Time: 3/9/99 - 12:32 PM

Value 0
Name: Component Information
Type: REG_BINARY
Data:
00000000 00 00 00 00 00 00 00 00 - 01 00 00 00 02 00 00 00
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Invalid
Bus Number: -1
Version: 0

```

Revision: 0

Value 2
Name: FeatureSet
Type: REG_DWORD
Data: 0x3ff

Value 3
Name: Identifier
Type: REG_SZ
Data: x86 Family 6 Model 7 Stepping 2

Value 4
Name: Update Signature
Type: REG_BINARY
Data: 00000000 00 00 00 00 20 00 00 00 - ....

Value 5
Name: Update Status
Type: REG_DWORD
Data: 0x2

Value 6
Name: VendorIdentifier
Type: REG_SZ
Data: GenuineIntel

Value 7
Name: ~MHz
Type: REG_DWORD
Data: 0x1f4

Key Name: HARDWARE\DESCRIPTION\System\CentralProcessor\2
Class Name: Processor
Last Write Time: 3/9/99 - 12:32 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 00 00 00 00 00 00 00 00 - 02 00 00 00 04 00 00 00
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Invalid
Bus Number: -1
Version: 0
Revision: 0

```

```

Value 2
Name: FeatureSet
Type: REG_DWORD
Data: 0x3ff

Value 3
Name: Identifier
Type: REG_SZ
Data: x86 Family 6 Model 7 Stepping 2

Value 4
Name: Update Signature
Type: REG_BINARY
Data: 00000000 00 00 00 00 20 00 00 00 - ....

Value 5
Name: Update Status
Type: REG_DWORD
Data: 0x2

Value 6
Name: VendorIdentifier
Type: REG_SZ
Data: GenuineIntel

Value 7
Name: ~MHz
Type: REG_DWORD
Data: 0x1f4

Key Name: HARDWARE\DESCRIPTION\System\CentralProcessor\3
Class Name: Processor
Last Write Time: 3/9/99 - 12:32 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 00 00 00 00 00 00 00 00 - 03 00 00 00 08 00 00 00
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Invalid
Bus Number: -1
Version: 0
Revision: 0

Value 2

```

```

Name:          FeatureSet
Type:          REG_DWORD
Data:          0x3ff

Value 3
Name:          Identifier
Type:          REG_SZ
Data:          x86 Family 6 Model 7 Stepping 2

Value 4
Name:          Update Signature
Type:          REG_BINARY
Data:          00000000 00 00 00 00 20 00 00 00 -
00000000      ....

Value 5
Name:          Update Status
Type:          REG_DWORD
Data:          0x2

Value 6
Name:          VendorIdentifier
Type:          REG_SZ
Data:          GenuineIntel

Value 7
Name:          ~MHz
Type:          REG_DWORD
Data:          0x1f4

Key Name:      HARDWARE\DESCRIPTION\System\FloatingPointProcessor
Class Name:    Processor
Last Write Time: 3/9/99 - 12:31 PM

Key Name:      HARDWARE\DESCRIPTION\System\FloatingPointProcessor\0
Class Name:    Processor
Last Write Time: 3/9/99 - 12:31 PM

Value 0
Name:          Component Information
Type:          REG_BINARY
Data:          00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 01 00 00 00
.....

Value 1
Name:          Configuration Data
Type:          REG_FULL_RESOURCE_DESCRIPTOR
                Interface Type:   Invalid
                Bus Number:      -1
                Version:         0
                Revision:        0

```

```

Value 2
Name:          Identifier
Type:          REG_SZ
Data:          x86 Family 6 Model 7 Stepping 2

Key Name:      HARDWARE\DESCRIPTION\System\FloatingPointProcessor\1
Class Name:    Processor
Last Write Time: 3/9/99 - 12:31 PM

Value 0
Name:          Component Information
Type:          REG_BINARY
Data:          00000000 00 00 00 00 00 00 00 00 - 01 00 00 00 02 00 00 00
.....

Value 1
Name:          Configuration Data
Type:          REG_FULL_RESOURCE_DESCRIPTOR
                Interface Type:   Invalid
                Bus Number:      -1
                Version:         0
                Revision:        0

Value 2
Name:          Identifier
Type:          REG_SZ
Data:          x86 Family 6 Model 7 Stepping 2

Key Name:      HARDWARE\DESCRIPTION\System\FloatingPointProcessor\2
Class Name:    Processor
Last Write Time: 3/9/99 - 12:31 PM

Value 0
Name:          Component Information
Type:          REG_BINARY
Data:          00000000 00 00 00 00 00 00 00 00 - 02 00 00 00 04 00 00 00
.....

Value 1
Name:          Configuration Data
Type:          REG_FULL_RESOURCE_DESCRIPTOR
                Interface Type:   Invalid
                Bus Number:      -1
                Version:         0
                Revision:        0

Value 2
Name:          Identifier

```

```

Type:          REG_SZ
Data:          x86 Family 6 Model 7 Stepping 2

Key Name:      HARDWARE\DESCRIPTION\System\FloatingPointProcessor\3
Class Name:    Processor
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name:          Component Information
Type:          REG_BINARY
Data:          00000000 00 00 00 00 00 00 00 00 - 03 00 00 00 08 00 00 00
.....

Value 1
Name:          Configuration Data
Type:          REG_FULL_RESOURCE_DESCRIPTOR
                Interface Type:   Invalid
                Bus Number:       -1
                Version:          0
                Revision:         0

Value 2
Name:          Identifier
Type:          REG_SZ
Data:          x86 Family 6 Model 7 Stepping 2

Key Name:      HARDWARE\DESCRIPTION\System\MultifunctionAdapter
Class Name:    Adapter
Last Write Time: 3/9/99 - 12:31 PM

Key Name:      HARDWARE\DESCRIPTION\System\MultifunctionAdapter\0
Class Name:    Adapter
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name:          Component Information
Type:          REG_BINARY
Data:          00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name:          Configuration Data
Type:          REG_FULL_RESOURCE_DESCRIPTOR
                Interface Type:   PCI
                Bus Number:       0
                Version:          0
                Revision:         0
                Partial Descriptor 0
                Resource:         Device Specific
                Disposition:      Undetermined

```

```

Reserved1:     0x00000000
Reserved2:     0x00000000
Data:          00000000 02 10 0c 01 .....

Value 2
Name:          Identifier
Type:          REG_SZ
Data:          PCI

Key Name:      HARDWARE\DESCRIPTION\System\MultifunctionAdapter\1
Class Name:    Adapter
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name:          Component Information
Type:          REG_BINARY
Data:          00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name:          Configuration Data
Type:          REG_FULL_RESOURCE_DESCRIPTOR
                Interface Type:   PCI
                Bus Number:       1
                Version:          0
                Revision:         0

Value 2
Name:          Identifier
Type:          REG_SZ
Data:          PCI

Key Name:      HARDWARE\DESCRIPTION\System\MultifunctionAdapter\10
Class Name:    Adapter
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name:          Component Information
Type:          REG_BINARY
Data:          00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name:          Configuration Data
Type:          REG_FULL_RESOURCE_DESCRIPTOR
                Interface Type:   PCI
                Bus Number:       10

```

Version: 0
Revision: 0

Resource: Device Specific
Disposition: Undetermined
Reserved1: 0x00000000
Reserved2: 0x00000000
Data:

Value 2
Name: Identifier
Type: REG_SZ
Data: PCI

00000000 24 50 6e 50 10 21 00 00 - 06 00 04 00 00 71 a8 00
\$PnP.!.....q..

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\11
Class Name: Adapter
Last Write Time: 3/9/99 - 12:31 PM

00000010 f0 8f a8 00 00 0f 00 00 - 00 00 00 40 00 00 04 00
.....@....
00000020 00 3a 00 00 41 d0 0c 02 - 08 80 00 03 00 47 01 80
...A.....G..

Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff

00000030 00 80 00 01 01 47 01 90 - 0c 90 0c 01 08 86 09 00
.....G.....
00000040 00 00 00 e0 ff 00 00 20 - 00 86 09 00 00 00 00 e0

.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: PCI
Bus Number: 11
Version: 0
Revision: 0

00000050 fe 00 00 10 00 79 00 79 - 00 79 00 32 00 01 41 d0
.....y.y.y.2..A.
00000060 0c 02 06 01 00 03 00 47 - 01 d0 04 d0 04 01 02 47
.....G.....G
00000070 01 00 10 00 10 01 40 47 - 01 40 10 40 10 01 10 47
.....@G.@...G

Value 2
Name: Identifier
Type: REG_SZ
Data: PCI

00000080 01 60 03 60 03 01 03 79 - 00 79 00 79 00 2a 00 02
..'.y.y.y.*..
00000090 41 d0 0c 02 08 80 00 03 - 00 86 09 00 00 00 00 c1
A.....

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\12
Class Name: Adapter
Last Write Time: 3/9/99 - 12:31 PM

000000a0 fe 00 00 0f 00 86 09 00 - 00 00 00 f0 fe 00 00 10
.....
000000b0 00 79 00 79 00 36 - 00 03 41 d0 0c 01 05 00
.y.y.y.6..A....

Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff

000000c0 00 03 00 86 09 00 01 00 - 00 00 00 00 00 0a 00 86
.....
000000d0 09 00 60 00 80 0e 00 00 - 80 01 00 86 09 00 00 00
..'.y.y.y.-..

.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Internal
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0

000000e0 00 10 00 00 00 00 00 79 - 00 79 00 79 00 2d 00 04
.....Y.Y.Y.-..
000000f0 41 d0 02 00 08 01 01 03 - 00 47 01 00 00 00 00 01
A.....G.....
00000100 10 47 01 81 00 81 00 01 - 0f 47 01 c0 00 c0 00 01
.G.....G.....
00000110 20 2a 10 01 79 00 79 00 - 79 00 25 00 05 41 d0 00
*.y.y.y.%.A..
00000120 00 08 00 01 03 00 47 01 - 20 00 20 00 01 02 47 01

```

00000180 09 00 00 03 00 47 01 60 - 00 60 00 01 01 47 01 64
.....G.'...'G.d
00000190 00 64 00 01 01 22 02 00 - 79 00 79 00 79 00 1d 00
.d...'..y.y.y...
000001a0 09 41 d0 0c 04 0b 80 00 - 03 00 47 01 f0 00 f0 00
.A.....G.....
000001b0 01 10 22 00 20 79 00 79 - 00 79 00 1a 00 0a 41 d0 .."
y.y.y....A.
000001c0 08 00 04 01 00 03 00 47 - 01 61 00 61 00 01 01 79
.....G.a.a...y
000001d0 00 79 00 79 00 26 00 0b - 41 d0 0c 02 05 00 00 03
.y.y.&..A.....
000001e0 00 85 11 00 01 00 00 00 - 00 00 00 00 00 01 00 00
.....
000001f0 00 00 00 00 00 79 00 79 - 00 79 00 1a 00 0c 41 d0
....y.y.y....A.
00000200 0a 03 06 04 00 03 00 47 - 01 f8 0c f8 0c 01 08 79
.....G.....y
00000210 00 79 00 79 00 1e 00 0e - 41 d0 0c 02 05 00 00 03
.y.y....A.....
00000220 00 86 09 00 20 00 e8 0c - 00 00 18 00 00 79 00 79 ....
.....y.y
00000230 00 79 00 18 00 10 41 d0 - 0f 13 09 02 00 88 00 22
.y....A....."
00000240 00 10 79 00 22 00 10 79 - 00 79 00 6e 00 12 41 d0
..y..'..y.y.n..A.
00000250 07 00 01 02 00 90 00 47 - 01 f0 03 f0 03 08 06 47
.....G.....G
00000260 01 f7 03 f7 03 01 01 22 - 40 00 2a 04 00 79 00 30
....."@.*..y.0
00000270 47 01 f0 03 f0 03 08 06 - 47 01 f7 03 f7 03 01 01
G.....G.....
00000280 22 40 00 2a 04 00 30 47 - 01 70 03 70 03 08 06 47
"@.*..0G.p.p...G
00000290 01 77 03 77 03 01 01 22 - 40 00 2a 04 00 30 47 01
.w.w...'@.*..0G.
000002a0 00 01 f8 0f 08 08 47 01 - 00 00 00 00 01 00 22 ff
.....G....."
000002b0 ff 2a 0f 00 38 79 00 79 - 00
...8y.y.

```

```

Value 2
Name: Identifier
Type: REG_SZ
Data: PNP BIOS

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13
Class Name: Adapter
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name: Component Information

```

```

Type: REG_BINARY
Data: 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0

Value 2
Name: Identifier
Type: REG_SZ
Data: ISA

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\DiskController
Class Name: Controller
Last Write Time: 3/9/99 - 12:31 PM

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\DiskController\0
Class Name: Controller
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 64 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
d.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Port
Disposition: Device Exclusive
Start: 0x000003f0
Length: 0x8
Type: Port

Partial Descriptor 1
Resource: Interrupt
Disposition: Undetermined

```

Vector: 6
Level: 6
Affinity: 0xffffffff
Type: Latched

Partial Descriptor 2
Resource: DMA
Disposition: Undetermined
Channel: 2
Port: 0

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\DiskController\0\DiskPeripheral
Class Name: Peripheral
Last Write Time: 3/9/99 - 12:31 PM

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\DiskController\0\DiskPeripheral\0
Class Name: Peripheral
Last Write Time: 3/9/99 - 12:31 PM

Value 0
Name: Component Information
Type: REG_BINARY
Data:
00000000 60 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
\.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Device Specific
Disposition: Undetermined
Reserved1: 0x00000000
Reserved2: 0x00000000
Data:
00000000 00 02 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
\.....

Value 2
Name: Identifier
Type: REG_SZ

Data: e7a99ecf-3e343dc6-A

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\DiskController\0\FloppyDiskPeripheral
Class Name: Peripheral
Last Write Time: 3/9/99 - 12:31 PM

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\DiskController\0\FloppyDiskPeripheral\0
Class Name: Peripheral
Last Write Time: 3/9/99 - 12:31 PM

Value 0
Name: Component Information
Type: REG_BINARY
Data:
00000000 00 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
\.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Device Specific
Disposition: Undetermined
Reserved1: 0x00000000
Reserved2: 0x00000000
Data:
00000000 02 00 00 00 00 00 00 00 00 - 00 00 00 00 a0 05 00 00
\.....
00000010 00 00 00 00 df 02 25 02 - 12 1b ff 6c f6 0f 05 4f
\.....%...l...O
00000020 00

Value 2
Name: Identifier
Type: REG_SZ
Data: FLOPPY1

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\KeyboardController
Class Name: Controller
Last Write Time: 3/9/99 - 12:31 PM

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\KeyboardController\0
Class Name: Controller
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data:
00000000 28 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
(.....)

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Port
Disposition: Device Exclusive
Start: 0x00000060
Length: 0x1
Type: Port
Partial Descriptor 1
Resource: Port
Disposition: Device Exclusive
Start: 0x00000064
Length: 0x1
Type: Port
Partial Descriptor 2
Resource: Interrupt
Disposition: Undetermined
Vector: 1
Level: 1
Affinity: 0xffffffff
Type: Latched

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\KeyboardController\0\KeyboardPeripheral
Class Name: Peripheral
Last Write Time: 3/9/99 - 12:31 PM

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\KeyboardController\0\KeyboardPeripheral\0
Class Name: Peripheral

Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data:
00000000 28 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
(.....)

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Device Specific
Disposition: Undetermined
Reserved1: 0x00000000
Reserved2: 0x00000000
Data:
00000000 00 00 00 00 04 00 20 00 -

Value 2
Name: Identifier
Type: REG_SZ
Data: PCAT_ENHANCED

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\PointerController
Class Name: Controller
Last Write Time: 3/9/99 - 12:31 PM

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\PointerController\0
Class Name: Controller
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data:
00000000 20 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0

Revision: 0
Partial Descriptor 0
Resource: Interrupt
Disposition: Undetermined
Vector: 12
Level: 12
Affinity: 0xffffffff
Type: Latched

00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: PCI
Bus Number: 2
Version: 0
Revision: 0

Value 2
Name: Identifier
Type: REG_SZ
Data: PCI

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\3
Class Name: Adapter
Last Write Time: 3/9/99 - 12:31 PM

Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: PCI
Bus Number: 3
Version: 0
Revision: 0

Value 2
Name: Identifier
Type: REG_SZ
Data: PCI

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4
Class Name: Adapter
Last Write Time: 3/9/99 - 12:31 PM

Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\PointerController\0\PointerPeripheral
Class Name: Peripheral
Last Write Time: 3/9/99 - 12:31 PM

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\PointerController\0\PointerPeripheral\0
Class Name: Peripheral
Last Write Time: 3/9/99 - 12:31 PM

Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 20 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0

Value 2
Name: Identifier
Type: REG_SZ
Data: MICROSOFT PS2 MOUSE

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\2
Class Name: Adapter
Last Write Time: 3/9/99 - 12:31 PM

Value 0
Name: Component Information
Type: REG_BINARY
Data:

```

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
      Interface Type: PCI
      Bus Number: 4
      Version: 0
      Revision: 0

Value 2
Name: Identifier
Type: REG_SZ
Data: PCI

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\5
Class Name: Adapter
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
      Interface Type: PCI
      Bus Number: 5
      Version: 0
      Revision: 0

Value 2
Name: Identifier
Type: REG_SZ
Data: PCI

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\6
Class Name: Adapter
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR

```

```

      Interface Type: PCI
      Bus Number: 6
      Version: 0
      Revision: 0

Value 2
Name: Identifier
Type: REG_SZ
Data: PCI

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\7
Class Name: Adapter
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
      Interface Type: PCI
      Bus Number: 7
      Version: 0
      Revision: 0

Value 2
Name: Identifier
Type: REG_SZ
Data: PCI

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\8
Class Name: Adapter
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
      Interface Type: PCI
      Bus Number: 8
      Version: 0

```

```

Revision: 0

Value 2
Name: Identifier
Type: REG_SZ
Data: PCI

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\9
Class Name: Adapter
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: PCI
Bus Number: 9
Version: 0
Revision: 0

Value 2
Name: Identifier
Type: REG_SZ
Data: PCI

Key Name: HARDWARE\DEVICEMAP
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:31 PM

Key Name: HARDWARE\DEVICEMAP\KeyboardClass
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM
Value 0
Name: \Device\KeyboardClass0
Type: REG_SZ
Data:
\REGISTRY\Machine\System\ControlSet001\Services\Kbdclass

Key Name: HARDWARE\DEVICEMAP\KeyboardPort
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM
Value 0
Name: \Device\KeyboardPort0

```

```

Type: REG_SZ
Data:
\REGISTRY\Machine\System\ControlSet001\Services\i8042prt

Key Name: HARDWARE\DEVICEMAP\PointerClass
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM
Value 0
Name: \Device\PointerClass0
Type: REG_SZ
Data:
\REGISTRY\Machine\System\ControlSet001\Services\Mouclass

Key Name: HARDWARE\DEVICEMAP\PointerPort
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM
Value 0
Name: \Device\PointerPort0
Type: REG_SZ
Data:
\REGISTRY\Machine\System\ControlSet001\Services\i8042prt

Key Name: HARDWARE\DEVICEMAP\Scsi
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:31 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name: DMAEnabled
Type: REG_DWORD
Data: 0x1

Value 1
Name: Driver
Type: REG_SZ
Data: symc8XX

Value 2
Name: Interrupt
Type: REG_DWORD
Data: 0x8

Value 3
Name: IOAddress
Type: REG_DWORD
Data: 0x3000

```

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:31 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Initiator Id 7
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:31 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 5
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:31 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 5\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:31 PM
 Value 0
 Name: Identifier
 Type: REG_SZ
 Data: TOSHIBA CD-ROM XM-5701TA0167

Value 1
 Name: Type
 Type: REG_SZ
 Data: CdRomPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 1
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:31 PM
 Value 0
 Name: DMAEnabled
 Type: REG_DWORD
 Data: 0x1

Value 1
 Name: Driver
 Type: REG_SZ
 Data: symc8XX

Value 2
 Name: Interrupt
 Type: REG_DWORD
 Data: 0xc

Value 3
 Name: IOAddress
 Type: REG_DWORD
 Data: 0x3400

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 1\Scsi Bus 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:31 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 1\Scsi Bus 0\Initiator Id 7
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:31 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 1\Scsi Bus 0\Target Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:31 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 1\Scsi Bus 0\Target Id 0\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:31 PM
 Value 0
 Name: Identifier
 Type: REG_SZ
 Data: SEAGATE ST39102LC 7503

Value 1
 Name: Type
 Type: REG_SZ
 Data: DiskPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 1\Scsi Bus 0\Target Id 8
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:31 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 1\Scsi Bus 0\Target Id 8\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:31 PM
 Value 0
 Name: Identifier
 Type: REG_SZ
 Data: SDR GEM200 2

Value 1
 Name: Type
 Type: REG_SZ
 Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 10
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM
 Value 0

Name: DMAEnabled
Type: REG_DWORD
Data: 0

Value 1
Name: Driver
Type: REG_SZ
Data: dac960nt

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 10\Scsi Bus 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 10\Scsi Bus 0\Initiator Id 255
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 10\Scsi Bus 0\Target Id 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 10\Scsi Bus 0\Target Id 0\Logical Unit Id 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Value 0
Name: Identifier
Type: REG_SZ
Data: MylexPHPPSEUDO 0100

Value 1
Name: Type
Type: REG_SZ
Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Value 0
Name: DMAEnabled
Type: REG_DWORD
Data: 0x1

Value 1
Name: Driver
Type: REG_SZ
Data: dac960nt

Value 2

Name: Interrupt
Type: REG_DWORD
Data: 0x20

Value 3
Name: IOAddress
Type: REG_DWORD
Data: 0xe4310000

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 0\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 0\Target Id 8
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 0\Target Id 8\Logical Unit Id 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Value 0
Name: Identifier
Type: REG_SZ
Data: SDR GEM200 2

Value 1
Name: Type
Type: REG_SZ
Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 1
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 1\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 1\Target Id 8
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 1\Target
 Id 8\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: Identifier
 Type: REG_SZ
 Data: SDR GEM200 2

Value 1
 Name: Type
 Type: REG_SZ
 Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 2
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus
 2\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 3
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus
 3\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 3\Target
 Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 3\Target
 Id 0\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: Identifier
 Type: REG_SZ
 Data: MYLEX DAC1164P 0507

Value 1
 Name: Type
 Type: REG_SZ
 Data: DiskPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 4
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus
 4\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 4\Target
 Id 6
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 4\Target
 Id 6\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: Identifier
 Type: REG_SZ
 Data: MYLEX GAM DEVICE

Value 1
 Name: Type
 Type: REG_SZ
 Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: DMAEnabled
 Type: REG_DWORD
 Data: 0x1

Value 1
 Name: Driver
 Type: REG_SZ
 Data: dac960nt

Value 2
 Name: Interrupt
 Type: REG_DWORD
 Data: 0x20

Value 3
 Name: IOAddress
 Type: REG_DWORD
 Data: 0xec110000

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus
 0\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus 1
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus
 1\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus 2
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus
 2\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus 3
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus
 3\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus 3\Target
 Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus 3\Target
 Id 0\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: Identifier
 Type: REG_SZ
 Data: MYLEX DAC1164P 0507

Value 1
 Name: Type
 Type: REG_SZ

Data: DiskPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus 4
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus
 4\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus 4\Target
 Id 6
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus 4\Target
 Id 6\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: Identifier
 Type: REG_SZ
 Data: MYLEX GAM DEVICE

Value 1
 Name: Type
 Type: REG_SZ
 Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: DMAEnabled
 Type: REG_DWORD
 Data: 0x1

Value 1
 Name: Driver
 Type: REG_SZ
 Data: dac960nt

Value 2
 Name: Interrupt
 Type: REG_DWORD
 Data: 0x20

Value 3
 Name: IOAddress
 Type: REG_DWORD

Data: 0xec210000

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 0\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 1
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 1\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 2
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 2\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 3
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 3\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 3\Target Id 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 3\Target Id 0\Logical Unit Id 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Value 0
Name: Identifier
Type: REG_SZ
Data: MYLEX DAC1164P 0507

Value 1
Name: Type
Type: REG_SZ
Data: DiskPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 4
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 4\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 4\Target Id 6
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 4\Target Id 6\Logical Unit Id 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Value 0
Name: Identifier
Type: REG_SZ
Data: MYLEX GAM DEVICE

Value 1
Name: Type
Type: REG_SZ
Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Value 0
Name: DMAEnabled
Type: REG_DWORD
Data: 0x1

Value 1
Name: Driver
Type: REG_SZ
Data: dac960nt

Value 2
Name: Interrupt
Type: REG_DWORD
Data: 0x20

Value 3
 Name: IOAddress
 Type: REG_DWORD
 Data: 0xec310000

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 0\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 1
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 1\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 2
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 2\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 3
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 3\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 3\Target Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 3\Target Id 0\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: Identifier

Type: REG_SZ
 Data: MYLEX DAC1164P 0507

Value 1
 Name: Type
 Type: REG_SZ
 Data: DiskPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 4
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 4\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 4\Target Id 6
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 4\Target Id 6\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: Identifier
 Type: REG_SZ
 Data: MYLEX GAM DEVICE

Value 1
 Name: Type
 Type: REG_SZ
 Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: DMAEnabled
 Type: REG_DWORD
 Data: 0x1

Value 1
 Name: Driver
 Type: REG_SZ
 Data: dac960nt

Value 2
 Name: Interrupt

Type: REG_DWORD
Data: 0x20

Value 3
Name: IOAddress
Type: REG_DWORD
Data: 0xf4110000

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 0\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 1
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 1\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 2
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 2\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 3
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 3\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 3\Target Id 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 3\Target Id 0\Logical Unit Id 0
Class Name: <NO CLASS>

Last Write Time: 3/9/99 - 12:32 PM
Value 0
Name: Identifier
Type: REG_SZ
Data: MYLEX DAC1164P 0507

Value 1
Name: Type
Type: REG_SZ
Data: DiskPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 4
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 4\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 4\Target Id 6
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 4\Target Id 6\Logical Unit Id 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM
Value 0
Name: Identifier
Type: REG_SZ
Data: MYLEX GAM DEVICE

Value 1
Name: Type
Type: REG_SZ
Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM
Value 0
Name: DMAEnabled
Type: REG_DWORD
Data: 0x1

Value 1
Name: Driver
Type: REG_SZ
Data: dac960nt

Value 2
 Name: Interrupt
 Type: REG_DWORD
 Data: 0x20

Value 3
 Name: IOAddress
 Type: REG_DWORD
 Data: 0xf4210000

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 0\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 1
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 1\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 2
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 2\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 3
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 3\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 3\Target Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 3\Target Id 0\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: Identifier
 Type: REG_SZ
 Data: MYLEX DAC1164P 0507

Value 1
 Name: Type
 Type: REG_SZ
 Data: DiskPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 4
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 4\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 4\Target Id 6
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 4\Target Id 6\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: Identifier
 Type: REG_SZ
 Data: MYLEX GAM DEVICE

Value 1
 Name: Type
 Type: REG_SZ
 Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: DMAEnabled
 Type: REG_DWORD
 Data: 0x1

Value 1

Name: Driver
Type: REG_SZ
Data: dac960nt

Value 2
Name: Interrupt
Type: REG_DWORD
Data: 0x20

Value 3
Name: IOAddress
Type: REG_DWORD
Data: 0xf4310000

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus
0\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 1
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus
1\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 2
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus
2\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 3
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus
3\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 3\Target
Id 0

Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 3\Target
Id 0\Logical Unit Id 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Value 0
Name: Identifier
Type: REG_SZ
Data: MYLEX DAC1164P 0507

Value 1
Name: Type
Type: REG_SZ
Data: DiskPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 4
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus
4\Initiator Id 254
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 4\Target
Id 6
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 4\Target
Id 6\Logical Unit Id 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Value 0
Name: Identifier
Type: REG_SZ
Data: MYLEX GAM DEVICE

Value 1
Name: Type
Type: REG_SZ
Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Value 0
Name: DMAEnabled
Type: REG_DWORD

Data: 0x1

Value 1
 Name: Driver
 Type: REG_SZ
 Data: dac960nt

Value 2
 Name: Interrupt
 Type: REG_DWORD
 Data: 0x20

Value 3
 Name: IOAddress
 Type: REG_DWORD
 Data: 0xf4410000

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 0\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 0\Target Id 8
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 0\Target Id 8\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: Identifier
 Type: REG_SZ
 Data: SNI STM/S R3 2

Value 1
 Name: Type
 Type: REG_SZ
 Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 1
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 1\Initiator Id 254

Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 1\Target Id 8
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 1\Target Id 8\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: Identifier
 Type: REG_SZ
 Data: SNI STM/S R3 2

Value 1
 Name: Type
 Type: REG_SZ
 Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 2
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 2\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 3
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 3\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 3\Target Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 3\Target Id 0\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: Identifier
 Type: REG_SZ

Data: MYLEX DAC1164P 0507

Value 1
 Name: Type
 Type: REG_SZ
 Data: DiskPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 3\Target
 Id 1
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 3\Target
 Id 1\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: Identifier
 Type: REG_SZ
 Data: MYLEX DAC1164P 0507

Value 1
 Name: Type
 Type: REG_SZ
 Data: DiskPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 4
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus
 4\Initiator Id 254
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 4\Target
 Id 6
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 4\Target
 Id 6\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:32 PM

Value 0
 Name: Identifier
 Type: REG_SZ
 Data: MYLEX GAM DEVICE

Value 1
 Name: Type

Type: REG_SZ
 Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\VIDEO
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:33 PM

Value 0
 Name: \Device\Video0
 Type: REG_SZ
 Data: \REGISTRY\Machine\System\ControlSet001\Services\cirrus\Device0

Value 1
 Name: VgaCompatible
 Type: REG_SZ
 Data: \Device\Video0

Key Name: HARDWARE\OWNERMAP
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 12:33 PM

Value 0
 Name: PCI_0_3
 Type: REG_SZ
 Data: \Device\Video0

Value 1
 Name: PCI_10_8
 Type: REG_SZ
 Data: \Device\ScsiPort8

Value 2
 Name: PCI_11_8
 Type: REG_SZ
 Data: \Device\ScsiPort9

Value 3
 Name: PCI_1_1
 Type: REG_SZ
 Data: \Device\E100B1

Value 4
 Name: PCI_1_2
 Type: REG_SZ
 Data: \Device\ScsiPort0

Value 5
 Name: PCI_1_3
 Type: REG_SZ
 Data: \Device\ScsiPort1

Value 6

```

Name:          PCI_2_8
Type:          REG_SZ
Data:          \Device\ScsiPort2

Value 7
Name:          PCI_4_8
Type:          REG_SZ
Data:          \Device\ScsiPort3

Value 8
Name:          PCI_5_8
Type:          REG_SZ
Data:          \Device\ScsiPort4

Value 9
Name:          PCI_6_8
Type:          REG_SZ
Data:          \Device\ScsiPort5

Value 10
Name:          PCI_8_8
Type:          REG_SZ
Data:          \Device\ScsiPort6

Value 11
Name:          PCI_9_8
Type:          REG_SZ
Data:          \Device\ScsiPort7

Key Name:      HARDWARE\RESOURCEMAP
Class Name:    <NO CLASS>
Last Write Time: 3/9/99 - 12:31 PM

Key Name:      HARDWARE\RESOURCEMAP\Hardware Abstraction Layer
Class Name:    <NO CLASS>
Last Write Time: 3/9/99 - 12:31 PM

Key Name:      HARDWARE\RESOURCEMAP\Hardware Abstraction Layer\MPS
1.4 - APIC platform
Class Name:    <NO CLASS>
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name:          .Raw
Type:          REG_RESOURCE_LIST
Data:          Full Resource Descriptor 0
                Interface Type:  Isa
                Bus Number:      0
                Version:         0
                Revision:        0
                Partial Descriptor 0
                Resource:         Interrupt

```

```

Disposition:   Driver Exclusive
Vector:        8
Level:         8
Affinity:      0x0000000f
Type:          Level Sensitive

Full Resource Descriptor 1
Interface Type: Internal
Bus Number:    0
Version:       0
Revision:      0
Partial Descriptor 0
Resource:      Interrupt
Disposition:   Driver Exclusive
Vector:        0
Level:         0
Affinity:      0x0000000f
Type:          Level Sensitive

Partial Descriptor 1
Resource:      Interrupt
Disposition:   Driver Exclusive
Vector:        1
Level:         1
Affinity:      0x0000000f
Type:          Level Sensitive

Partial Descriptor 2
Resource:      Interrupt
Disposition:   Driver Exclusive
Vector:        2
Level:         2
Affinity:      0x0000000f
Type:          Level Sensitive

Partial Descriptor 3
Resource:      Interrupt
Disposition:   Driver Exclusive
Vector:        3
Level:         3
Affinity:      0x0000000f
Type:          Level Sensitive

Partial Descriptor 4
Resource:      Interrupt
Disposition:   Driver Exclusive
Vector:        4
Level:         4
Affinity:      0x0000000f
Type:          Level Sensitive

Partial Descriptor 5

```


Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 5
 Level: 5
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 6
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 6
 Level: 6
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 7
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 7
 Level: 7
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 8
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 8
 Level: 8
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 9
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 9
 Level: 9
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 10
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 10
 Level: 10
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 11
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 11
 Level: 11
 Affinity: 0x0000000f

Type: Level Sensitive

Partial Descriptor 12
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 12
 Level: 12
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 13
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 13
 Level: 13
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 14
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 14
 Level: 14
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 15
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 15
 Level: 15
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 16
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 16
 Level: 16
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 17
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 17
 Level: 17
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 18
 Resource: Interrupt
 Disposition: Driver Exclusive

Vector: 18
 Level: 18
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 19
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 19
 Level: 19
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 20
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 20
 Level: 20
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 21
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 21
 Level: 21
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 22
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 22
 Level: 22
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 23
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 23
 Level: 23
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 24
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 24
 Level: 24
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 25
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 25
 Level: 25
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 26
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 26
 Level: 26
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 27
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 27
 Level: 27
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 28
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 28
 Level: 28
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 29
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 29
 Level: 29
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 30
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 30
 Level: 30
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 31
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 31
 Level: 31

Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 32
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 32
 Level: 32
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 33
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 33
 Level: 33
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 34
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 34
 Level: 34
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 35
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 35
 Level: 35
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 36
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 36
 Level: 36
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 37
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 37
 Level: 37
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 38
 Resource: Interrupt

Disposition: Driver Exclusive
 Vector: 38
 Level: 38
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 39
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 39
 Level: 39
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 40
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 40
 Level: 40
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 41
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 41
 Level: 41
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 42
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 42
 Level: 42
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 43
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 43
 Level: 43
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 44
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 44
 Level: 44
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 45
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 45
Level: 45
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 46
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 46
Level: 46
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 47
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 47
Level: 47
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 48
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 61
Level: 61
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 49
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 65
Level: 65
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 50
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 80
Level: 80
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 51
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 193

Level: 193
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 52
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 225
Level: 225
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 53
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 253
Level: 253
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 54
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 254
Level: 254
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 55
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 255
Level: 255
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 56
Resource: Port
Disposition: Driver Exclusive
Start: 0x00000000
Length: 0x10
Type: Port

Partial Descriptor 57
Resource: Port
Disposition: Driver Exclusive
Start: 0x00000020
Length: 0x2
Type: Port

Partial Descriptor 58
Resource: Port
Disposition: Driver Exclusive

```

Start:          0x00000040
Length:         0x4
Type:           Port

Partial Descriptor 59
Resource:       Port
Disposition:    Driver Exclusive
Start:          0x00000048
Length:         0x4
Type:           Port

Partial Descriptor 60
Resource:       Port
Disposition:    Driver Exclusive
Start:          0x00000061
Length:         0x1
Type:           Port

Partial Descriptor 61
Resource:       Port
Disposition:    Driver Exclusive
Start:          0x00000070
Length:         0x2
Type:           Port

Partial Descriptor 62
Resource:       Port
Disposition:    Driver Exclusive
Start:          0x00000080
Length:         0x10
Type:           Port

Partial Descriptor 63
Resource:       Port
Disposition:    Driver Exclusive
Start:          0x00000092
Length:         0x1
Type:           Port

Partial Descriptor 64
Resource:       Port
Disposition:    Driver Exclusive
Start:          0x000000a0
Length:         0x2
Type:           Port

Partial Descriptor 65
Resource:       Port
Disposition:    Driver Exclusive
Start:          0x000000c0
Length:         0x10
Type:           Port

```

```

Value 1
Name:
Type:
Data:

```

```

Partial Descriptor 66
Resource:       Port
Disposition:    Driver Exclusive
Start:          0x000000f0
Length:         0x10
Type:           Port

Partial Descriptor 67
Resource:       Memory
Disposition:    Driver Exclusive
Start:          0xfec00000
Length:         0x400
Type:           Read / Write

Partial Descriptor 68
Resource:       Memory
Disposition:    Driver Exclusive
Start:          0xfec00000
Length:         0x400
Type:           Read / Write

.Translated
REG_RESOURCE_LIST

Full Resource Descriptor 0
Interface Type: Isa
Bus Number:     0
Version:        0
Revision:       0
Partial Descriptor 0
Resource:       Interrupt
Disposition:    Driver Exclusive
Vector:         209
Level:          28
Affinity:       0x0000000f
Type:           Level Sensitive

Full Resource Descriptor 1
Interface Type: Internal
Bus Number:     0
Version:        0
Revision:       0
Partial Descriptor 0
Resource:       Interrupt
Disposition:    Driver Exclusive
Vector:         0
Level:          0
Affinity:       0x0000000f
Type:           Level Sensitive

```

Partial Descriptor 1
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 1
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 2
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 2
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 3
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 3
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 4
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 4
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 5
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 5
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 6
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 6
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 7
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 7

Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 8
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 8
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 9
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 9
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 10
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 10
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 11
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 11
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 12
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 12
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 13
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 13
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 14

Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 14
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 15
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 15
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 16
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 16
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 17
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 17
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 18
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 18
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 19
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 19
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 20
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 20
 Level: 0
 Affinity: 0x0000000f

Type: Level Sensitive

Partial Descriptor 21
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 21
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 22
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 22
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 23
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 23
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 24
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 24
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 25
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 25
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 26
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 26
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 27
 Resource: Interrupt
 Disposition: Driver Exclusive

Vector: 27
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 28
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 28
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 29
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 29
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 30
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 30
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 31
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 31
 Level: 31
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 32
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 32
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 33
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 33
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 34
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 34
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 35
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 35
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 36
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 36
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 37
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 37
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 38
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 38
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 39
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 39
 Level: 0
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 40
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 40
 Level: 0

Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 41
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 41
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 42
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 42
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 43
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 43
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 44
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 44
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 45
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 45
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 46
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 46
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 47
Resource: Interrupt

Disposition: Driver Exclusive
Vector: 47
Level: 0
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 48
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 61
Level: 1
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 49
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 65
Level: 2
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 50
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 80
Level: 255
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 51
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 193
Level: 27
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 52
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 225
Level: 29
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 53
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 253
Level: 30
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 54
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 254
Level: 30
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 55
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 255
Level: 31
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 56
Resource: Port
Disposition: Driver Exclusive
Start: 0x00000000
Length: 0x10
Type: Port

Partial Descriptor 57
Resource: Port
Disposition: Driver Exclusive
Start: 0x00000020
Length: 0x2
Type: Port

Partial Descriptor 58
Resource: Port
Disposition: Driver Exclusive
Start: 0x00000040
Length: 0x4
Type: Port

Partial Descriptor 59
Resource: Port
Disposition: Driver Exclusive
Start: 0x00000048
Length: 0x4
Type: Port

Partial Descriptor 60
Resource: Port
Disposition: Driver Exclusive
Start: 0x00000061
Length: 0x1
Type: Port

Partial Descriptor 61

Resource: Port
Disposition: Driver Exclusive
Start: 0x00000070
Length: 0x2
Type: Port

Partial Descriptor 62
Resource: Port
Disposition: Driver Exclusive
Start: 0x00000080
Length: 0x10
Type: Port

Partial Descriptor 63
Resource: Port
Disposition: Driver Exclusive
Start: 0x00000092
Length: 0x1
Type: Port

Partial Descriptor 64
Resource: Port
Disposition: Driver Exclusive
Start: 0x000000a0
Length: 0x2
Type: Port

Partial Descriptor 65
Resource: Port
Disposition: Driver Exclusive
Start: 0x000000c0
Length: 0x10
Type: Port

Partial Descriptor 66
Resource: Port
Disposition: Driver Exclusive
Start: 0x000000f0
Length: 0x10
Type: Port

Partial Descriptor 67
Resource: Memory
Disposition: Driver Exclusive
Start: 0xfec00000
Length: 0x400
Type: Read / Write

Partial Descriptor 68
Resource: Memory
Disposition: Driver Exclusive
Start: 0xfef00000
Length: 0x400

```

Type:          Read / Write

Key Name:      HARDWARE\RESOURCEMAP\KeyboardPort/PointerPort
Class Name:    <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name:      HARDWARE\RESOURCEMAP\KeyboardPort/PointerPort\i8042prt
Class Name:    <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM
Value 0
Name:          \Device\KeyboardPort0.Raw
Type:          REG_RESOURCE_LIST
Data:
Full Resource Descriptor 0
  Interface Type:  Isa
  Bus Number:     0
  Version:        0
  Revision:       0
  Partial Descriptor 0
    Resource:      Interrupt
    Disposition:   Device Exclusive
    Vector:        1
    Level:         1
    Affinity:      0xffffffff
    Type:          Latched
  Partial Descriptor 1
    Resource:      Interrupt
    Disposition:   Device Exclusive
    Vector:        12
    Level:         12
    Affinity:      0xffffffff
    Type:          Latched
  Partial Descriptor 2
    Resource:      Port
    Disposition:   Driver Exclusive
    Start:         0x00000060
    Length:        0x1
    Type:          Port
  Partial Descriptor 3
    Resource:      Port
    Disposition:   Driver Exclusive
    Start:         0x00000064
    Length:        0x1
    Type:          Port

```

```

Value 1
Name:          \Device\KeyboardPort0.Translated
Type:          REG_RESOURCE_LIST
Data:
Full Resource Descriptor 0
  Interface Type:  Isa
  Bus Number:     0
  Version:        0
  Revision:       0
  Partial Descriptor 0
    Resource:      Interrupt
    Disposition:   Device Exclusive
    Vector:        145
    Level:         8
    Affinity:      0x0000000f
    Type:          Latched
  Partial Descriptor 1
    Resource:      Interrupt
    Disposition:   Device Exclusive
    Vector:        161
    Level:         9
    Affinity:      0x0000000f
    Type:          Latched
  Partial Descriptor 2
    Resource:      Port
    Disposition:   Driver Exclusive
    Start:         0x00000060
    Length:        0x1
    Type:          Port
  Partial Descriptor 3
    Resource:      Port
    Disposition:   Driver Exclusive
    Start:         0x00000064
    Length:        0x1
    Type:          Port

Key Name:      HARDWARE\RESOURCEMAP\OtherDrivers
Class Name:    <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name:      HARDWARE\RESOURCEMAP\OtherDrivers\E100B
Class Name:    <NO CLASS>
Last Write Time: 3/9/99 - 12:33 PM
Value 0
Name:          \Device\E100B1.Raw
Type:          REG_RESOURCE_LIST
Data:

```

```

Full Resource Descriptor 0
Interface Type:  PCI
Bus Number:     1
Version:        0
Revision:       0
Partial Descriptor 0
Resource:       Memory
Disposition:    Device Exclusive
Start:          0xe80ff000
Length:         0x1e
Type:           Read / Write

Partial Descriptor 1
Resource:       Port
Disposition:    Device Exclusive
Start:          0x00003800
Length:         0x1e
Type:           Port

Partial Descriptor 2
Resource:       Interrupt
Disposition:    Shared
Vector:         4
Level:          4
Affinity:       0x00000000
Type:           Level Sensitive

```

```

Resource:       Interrupt
Disposition:    Shared
Vector:         82
Level:          4
Affinity:       0x0000000f
Type:           Level Sensitive

Key Name:       HARDWARE\RESOURCEMAP\OtherDrivers\Floppy
Class Name:     <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM
Value 0
Name:           .Raw
Type:           REG_RESOURCE_LIST
Data:

```

```

Full Resource Descriptor 0
Interface Type:  Isa
Bus Number:     0
Version:        0
Revision:       0
Partial Descriptor 0
Resource:       Port
Disposition:    Shared
Start:          0x000003f0
Length:         0x6
Type:           Port

```

```

Partial Descriptor 1
Resource:       Port
Disposition:    Shared
Start:          0x000003f7
Length:         0x1
Type:           Port

```

```

Partial Descriptor 2
Resource:       DMA
Disposition:    Shared
Channel:        2
Port:           0

```

```

Partial Descriptor 3
Resource:       Interrupt
Disposition:    Shared
Vector:         6
Level:          6
Affinity:       0x00000000
Type:           Latched

```

Value 1

```

Value 1
Name:           \Device\E100B1.Translated
Type:           REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type:  PCI
Bus Number:     1
Version:        0
Revision:       0
Partial Descriptor 0
Resource:       Memory
Disposition:    Device Exclusive
Start:          0xe80ff000
Length:         0x1e
Type:           Read / Write

Partial Descriptor 1
Resource:       Port
Disposition:    Device Exclusive
Start:          0x00003800
Length:         0x1e
Type:           Port

Partial Descriptor 2

```

```

Name: .Translated
Type: REG_RESOURCE_LIST
Data:
  Full Resource Descriptor 0
    Interface Type: Isa
    Bus Number: 0
    Version: 0
    Revision: 0
    Partial Descriptor 0
      Resource: Port
      Disposition: Shared
      Start: 0x000003f0
      Length: 0x6
      Type: Port

    Partial Descriptor 1
      Resource: Port
      Disposition: Shared
      Start: 0x000003f7
      Length: 0x1
      Type: Port

    Partial Descriptor 2
      Resource: DMA
      Disposition: Shared
      Channel: 2
      Port: 0

    Partial Descriptor 3
      Resource: Interrupt
      Disposition: Shared
      Vector: 177
      Level: 10
      Affinity: 0x0000000f
      Type: Latched

```

```

Key Name: HARDWARE\RESOURCEMAP\ScsiAdapter
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:31 PM

```

```

Key Name: HARDWARE\RESOURCEMAP\ScsiAdapter\dac960nt
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

```

```

Value 0
  Name: \Device\ScsiPort10.Raw
  Type: REG_RESOURCE_LIST
  Data:
    Full Resource Descriptor 0
      Interface Type: PCI
      Bus Number: 0

```

```

Version: 0
Revision: 0

```

```

Value 1
  Name: \Device\ScsiPort10.Translated
  Type: REG_RESOURCE_LIST
  Data:

```

```

  Full Resource Descriptor 0
    Interface Type: PCI
    Bus Number: 0
    Version: 0
    Revision: 0

```

```

Value 2
  Name: \Device\ScsiPort2.Raw
  Type: REG_RESOURCE_LIST
  Data:

```

```

  Full Resource Descriptor 0
    Interface Type: PCI
    Bus Number: 2
    Version: 0
    Revision: 0
    Partial Descriptor 0
      Resource: Interrupt
      Disposition: Shared
      Vector: 32
      Level: 32
      Affinity: 0x00000000
      Type: Level Sensitive

```

```

  Partial Descriptor 1
    Resource: Memory
    Disposition: Device Exclusive
    Start: 0xe4310000
    Length: 0x80
    Type: Read / Write

```

```

  Partial Descriptor 2
    Resource: Port
    Disposition: Device Exclusive
    Start: 0x00004000
    Length: 0x80
    Type: Port

```

```

  Partial Descriptor 3
    Resource: Memory
    Disposition: Device Exclusive
    Start: 0xea000000
    Length: 0x2000000
    Type: Write Only

```

Value 3
Name:
Type:
Data:

```
\Device\ScsiPort2.Translated
REG_RESOURCE_LIST

Full Resource Descriptor 0
  Interface Type:  PCI
  Bus Number:      2
  Version:         0
  Revision:        0
  Partial Descriptor 0
    Resource:      Interrupt
    Disposition:   Shared
    Vector:        97
    Level:         5
    Affinity:      0x0000000f
    Type:          Level Sensitive

  Partial Descriptor 1
    Resource:      Memory
    Disposition:   Device Exclusive
    Start:         0xe4310000
    Length:        0x80
    Type:          Read / Write

  Partial Descriptor 2
    Resource:      Port
    Disposition:   Device Exclusive
    Start:         0x00004000
    Length:        0x80
    Type:          Port

  Partial Descriptor 3
    Resource:      Memory
    Disposition:   Device Exclusive
    Start:         0xea000000
    Length:        0x2000000
    Type:          Read / Write
```

Value 4
Name:
Type:
Data:

```
\Device\ScsiPort3.Raw
REG_RESOURCE_LIST

Full Resource Descriptor 0
  Interface Type:  PCI
  Bus Number:      4
  Version:         0
  Revision:        0
  Partial Descriptor 0
    Resource:      Interrupt
```

```
Disposition:  Shared
Vector:       32
Level:        32
Affinity:     0x00000000
Type:         Level Sensitive
```

```
Partial Descriptor 1
  Resource:      Memory
  Disposition:   Device Exclusive
  Start:         0xec110000
  Length:        0x80
  Type:          Read / Write
```

```
Partial Descriptor 2
  Resource:      Port
  Disposition:   Device Exclusive
  Start:         0x00005000
  Length:        0x80
  Type:          Port
```

```
Partial Descriptor 3
  Resource:      Memory
  Disposition:   Device Exclusive
  Start:         0xee000000
  Length:        0x2000000
  Type:          Write Only
```

Value 5
Name:
Type:
Data:

```
\Device\ScsiPort3.Translated
REG_RESOURCE_LIST

Full Resource Descriptor 0
  Interface Type:  PCI
  Bus Number:      4
  Version:         0
  Revision:        0
  Partial Descriptor 0
    Resource:      Interrupt
    Disposition:   Shared
    Vector:        81
    Level:         4
    Affinity:      0x0000000f
    Type:          Level Sensitive

  Partial Descriptor 1
    Resource:      Memory
    Disposition:   Device Exclusive
    Start:         0xec110000
    Length:        0x80
    Type:          Read / Write
```

Partial Descriptor 2
Resource: Port
Disposition: Device Exclusive
Start: 0x00005000
Length: 0x80
Type: Port

Partial Descriptor 3
Resource: Memory
Disposition: Device Exclusive
Start: 0xee000000
Length: 0x2000000
Type: Read / Write

Value 6
Name: \Device\ScsiPort4.Raw
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: PCI
Bus Number: 5
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Interrupt
Disposition: Shared
Vector: 32
Level: 32
Affinity: 0x00000000
Type: Level Sensitive

Partial Descriptor 1
Resource: Memory
Disposition: Device Exclusive
Start: 0xec210000
Length: 0x80
Type: Read / Write

Partial Descriptor 2
Resource: Port
Disposition: Device Exclusive
Start: 0x00006000
Length: 0x80
Type: Port

Partial Descriptor 3
Resource: Memory
Disposition: Device Exclusive
Start: 0xf0000000
Length: 0x2000000
Type: Write Only

Value 7
Name: \Device\ScsiPort4.Translated
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: PCI
Bus Number: 5
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Interrupt
Disposition: Shared
Vector: 178
Level: 10
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 1
Resource: Memory
Disposition: Device Exclusive
Start: 0xec210000
Length: 0x80
Type: Read / Write

Partial Descriptor 2
Resource: Port
Disposition: Device Exclusive
Start: 0x00006000
Length: 0x80
Type: Port

Partial Descriptor 3
Resource: Memory
Disposition: Device Exclusive
Start: 0xf0000000
Length: 0x2000000
Type: Read / Write

Value 8
Name: \Device\ScsiPort5.Raw
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: PCI
Bus Number: 6
Version: 0
Revision: 0
Partial Descriptor 0

Resource: Interrupt
Disposition: Shared
Vector: 32
Level: 32
Affinity: 0x00000000
Type: Level Sensitive

Partial Descriptor 1
Resource: Memory
Disposition: Device Exclusive
Start: 0xec310000
Length: 0x80
Type: Read / Write

Partial Descriptor 2
Resource: Port
Disposition: Device Exclusive
Start: 0x00007000
Length: 0x80
Type: Port

Partial Descriptor 3
Resource: Memory
Disposition: Device Exclusive
Start: 0xf2000000
Length: 0x2000000
Type: Write Only

Value 9

Name: \Device\ScsiPort5.Translated
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: PCI
Bus Number: 6
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Interrupt
Disposition: Shared
Vector: 162
Level: 9
Affinity: 0x0000000f
Type: Level Sensitive
Partial Descriptor 1
Resource: Memory
Disposition: Device Exclusive
Start: 0xec310000
Length: 0x80
Type: Read / Write

Value 10

Name: \Device\ScsiPort6.Raw
Type: REG_RESOURCE_LIST
Data:

Partial Descriptor 2
Resource: Port
Disposition: Device Exclusive
Start: 0x00007000
Length: 0x80
Type: Port

Partial Descriptor 3
Resource: Memory
Disposition: Device Exclusive
Start: 0xf2000000
Length: 0x2000000
Type: Read / Write

Full Resource Descriptor 0
Interface Type: PCI
Bus Number: 8
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Interrupt
Disposition: Shared
Vector: 32
Level: 32
Affinity: 0x00000000
Type: Level Sensitive

Partial Descriptor 1
Resource: Memory
Disposition: Device Exclusive
Start: 0xf4110000
Length: 0x80
Type: Read / Write

Partial Descriptor 2
Resource: Port
Disposition: Device Exclusive
Start: 0x00008000
Length: 0x80
Type: Port

Partial Descriptor 3
Resource: Memory
Disposition: Device Exclusive
Start: 0xf6000000
Length: 0x2000000

Type: Write Only

Value 11

Name: \Device\ScsiPort6.Translated
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0

Interface Type: PCI
Bus Number: 8
Version: 0
Revision: 0

Partial Descriptor 0

Resource: Interrupt
Disposition: Shared
Vector: 146
Level: 8
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 1

Resource: Memory
Disposition: Device Exclusive
Start: 0xf4110000
Length: 0x80
Type: Read / Write

Partial Descriptor 2

Resource: Port
Disposition: Device Exclusive
Start: 0x00008000
Length: 0x80
Type: Port

Partial Descriptor 3

Resource: Memory
Disposition: Device Exclusive
Start: 0xf6000000
Length: 0x2000000
Type: Read / Write

Value 12

Name: \Device\ScsiPort7.Raw
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0

Interface Type: PCI
Bus Number: 9
Version: 0
Revision: 0

Partial Descriptor 0

Resource: Interrupt
Disposition: Shared
Vector: 32
Level: 32
Affinity: 0x00000000
Type: Level Sensitive

Partial Descriptor 1

Resource: Memory
Disposition: Device Exclusive
Start: 0xf4210000
Length: 0x80
Type: Read / Write

Partial Descriptor 2

Resource: Port
Disposition: Device Exclusive
Start: 0x00009000
Length: 0x80
Type: Port

Partial Descriptor 3

Resource: Memory
Disposition: Device Exclusive
Start: 0xf8000000
Length: 0x2000000
Type: Write Only

Value 13

Name: \Device\ScsiPort7.Translated
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0

Interface Type: PCI
Bus Number: 9
Version: 0
Revision: 0

Partial Descriptor 0

Resource: Interrupt
Disposition: Shared
Vector: 130
Level: 7
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 1

Resource: Memory
Disposition: Device Exclusive
Start: 0xf4210000
Length: 0x80

Type: Read / Write

Partial Descriptor 2
 Resource: Port
 Disposition: Device Exclusive
 Start: 0x00009000
 Length: 0x80
 Type: Port

Partial Descriptor 3
 Resource: Memory
 Disposition: Device Exclusive
 Start: 0xf8000000
 Length: 0x2000000
 Type: Read / Write

Value 15
 Name:
 Type:
 Data:

Length: 0x2000000
 Type: Write Only

\Device\ScsiPort8.Translated
 REG_RESOURCE_LIST

Full Resource Descriptor 0
 Interface Type: PCI
 Bus Number: 10
 Version: 0
 Revision: 0
 Partial Descriptor 0
 Resource: Interrupt
 Disposition: Shared
 Vector: 114
 Level: 6
 Affinity: 0x0000000f
 Type: Level Sensitive

Partial Descriptor 1
 Resource: Memory
 Disposition: Device Exclusive
 Start: 0xf4310000
 Length: 0x80
 Type: Read / Write

Partial Descriptor 2
 Resource: Port
 Disposition: Device Exclusive
 Start: 0x0000a000
 Length: 0x80
 Type: Port

Partial Descriptor 3
 Resource: Memory
 Disposition: Device Exclusive
 Start: 0xfa000000
 Length: 0x2000000
 Type: Read / Write

Value 16
 Name:
 Type:
 Data:

\Device\ScsiPort9.Raw
 REG_RESOURCE_LIST

Full Resource Descriptor 0
 Interface Type: PCI
 Bus Number: 11
 Version: 0

Value 14
 Name:
 Type:
 Data:

\Device\ScsiPort8.Raw
 REG_RESOURCE_LIST

Full Resource Descriptor 0
 Interface Type: PCI
 Bus Number: 10
 Version: 0
 Revision: 0
 Partial Descriptor 0
 Resource: Interrupt
 Disposition: Shared
 Vector: 32
 Level: 32
 Affinity: 0x00000000
 Type: Level Sensitive

Partial Descriptor 1
 Resource: Memory
 Disposition: Device Exclusive
 Start: 0xf4310000
 Length: 0x80
 Type: Read / Write

Partial Descriptor 2
 Resource: Port
 Disposition: Device Exclusive
 Start: 0x0000a000
 Length: 0x80
 Type: Port

Partial Descriptor 3
 Resource: Memory
 Disposition: Device Exclusive
 Start: 0xfa000000

```

Revision:          0
Partial Descriptor 0
  Resource:        Interrupt
  Disposition:     Shared
  Vector:          32
  Level:           32
  Affinity:        0x00000000
  Type:            Level Sensitive

Partial Descriptor 1
  Resource:        Memory
  Disposition:     Device Exclusive
  Start:           0xf4410000
  Length:          0x80
  Type:            Read / Write

Partial Descriptor 2
  Resource:        Port
  Disposition:     Device Exclusive
  Start:           0x0000b000
  Length:          0x80
  Type:            Port

Partial Descriptor 3
  Resource:        Memory
  Disposition:     Device Exclusive
  Start:           0xfc000000
  Length:          0x2000000
  Type:            Write Only

```

```

Length:            0x80
Type:              Read / Write

Partial Descriptor 2
  Resource:        Port
  Disposition:     Device Exclusive
  Start:           0x0000b000
  Length:          0x80
  Type:            Port

Partial Descriptor 3
  Resource:        Memory
  Disposition:     Device Exclusive
  Start:           0xfc000000
  Length:          0x2000000
  Type:            Read / Write

```

```

Key Name:          HARDWARE\RESOURCEMAP\ScsiAdapter\symc8XX
Class Name:        <NO CLASS>
Last Write Time:  3/9/99 - 12:31 PM
Value 0
Name:              \Device\ScsiPort0.Raw
Type:              REG_RESOURCE_LIST
Data:

```

```

Full Resource Descriptor 0
  Interface Type:  PCI
  Bus Number:     1
  Version:         0
  Revision:        0
  Partial Descriptor 0
    Resource:      Interrupt
    Disposition:   Shared
    Vector:        8
    Level:         8
    Affinity:      0x00000000
    Type:          Level Sensitive

  Partial Descriptor 1
    Resource:      Port
    Disposition:   Device Exclusive
    Start:         0x00003000
    Length:        0x100
    Type:          Port

  Partial Descriptor 2
    Resource:      Memory
    Disposition:   Device Exclusive
    Start:         0xe4202000
    Length:        0x100
    Type:          Read / Write

```

Value 17

```

Name:              \Device\ScsiPort9.Translated
Type:              REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
  Interface Type:  PCI
  Bus Number:     11
  Version:         0
  Revision:        0
  Partial Descriptor 0
    Resource:      Interrupt
    Disposition:   Shared
    Vector:        98
    Level:         5
    Affinity:      0x0000000f
    Type:          Level Sensitive

  Partial Descriptor 1
    Resource:      Memory
    Disposition:   Device Exclusive
    Start:         0xf4410000

```

Partial Descriptor 3
Resource: Memory
Disposition: Device Exclusive
Start: 0xe4200000
Length: 0x1000
Type: Read / Write

Value 1

Name: \Device\ScsiPort0.Translated
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: PCI
Bus Number: 1
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Interrupt
Disposition: Shared
Vector: 129
Level: 7
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 1
Resource: Port
Disposition: Device Exclusive
Start: 0x00003000
Length: 0x100
Type: Port

Partial Descriptor 2
Resource: Memory
Disposition: Device Exclusive
Start: 0xe4202000
Length: 0x100
Type: Read / Write

Partial Descriptor 3
Resource: Memory
Disposition: Device Exclusive
Start: 0xe4200000
Length: 0x1000
Type: Read / Write

Value 2

Name: \Device\ScsiPort1.Raw
Type: REG_RESOURCE_LIST

Data:

Full Resource Descriptor 0
Interface Type: PCI
Bus Number: 1
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Interrupt
Disposition: Shared
Vector: 12
Level: 12
Affinity: 0x00000000
Type: Level Sensitive

Partial Descriptor 1
Resource: Port
Disposition: Device Exclusive
Start: 0x00003400
Length: 0x100
Type: Port

Partial Descriptor 2
Resource: Memory
Disposition: Device Exclusive
Start: 0xe4202400
Length: 0x100
Type: Read / Write

Partial Descriptor 3
Resource: Memory
Disposition: Device Exclusive
Start: 0xe4201000
Length: 0x1000
Type: Read / Write

Value 3

Name: \Device\ScsiPort1.Translated
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: PCI
Bus Number: 1
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Interrupt
Disposition: Shared
Vector: 113
Level: 6
Affinity: 0x0000000f
Type: Level Sensitive

Partial Descriptor 1
Resource: Port
Disposition: Device Exclusive
Start: 0x00003400
Length: 0x100
Type: Port

Partial Descriptor 2
Resource: Memory
Disposition: Device Exclusive
Start: 0xe4202400
Length: 0x100
Type: Read / Write

Partial Descriptor 3
Resource: Memory
Disposition: Device Exclusive
Start: 0xe4201000
Length: 0x1000
Type: Read / Write

Key Name: HARDWARE\RESOURCEMAP\System Resources
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:31 PM

Key Name: HARDWARE\RESOURCEMAP\System Resources\Physical Memory
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:31 PM

Value 0
Name: .Translated
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: Internal
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Memory
Disposition: Device Exclusive
Start: 0x00001000
Length: 0x9b000
Type: Read / Write
Partial Descriptor 1
Resource: Memory
Disposition: Device Exclusive
Start: 0x00100000
Length: 0xeff000

Type: Read / Write

Partial Descriptor 2
Resource: Memory
Disposition: Device Exclusive
Start: 0x01000000
Length: 0xe3000000
Type: Read / Write

Key Name: HARDWARE\RESOURCEMAP\System Resources\Reserved
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name: .Translated
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: Internal
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Memory
Disposition: Device Exclusive
Start: 0x0009c000
Length: 0x4000
Type: Read / Write

Partial Descriptor 1
Resource: Memory
Disposition: Device Exclusive
Start: 0x000ea000
Length: 0x16000
Type: Read / Write

Partial Descriptor 2
Resource: Memory
Disposition: Device Exclusive
Start: 0x00fff000
Length: 0x1000
Type: Read / Write

Partial Descriptor 3
Resource: Memory
Disposition: Device Exclusive
Start: 0xfec00000
Length: 0x100000
Type: Read / Write

Partial Descriptor 4

Resource: Memory
Disposition: Device Exclusive
Start: 0xfee00000
Length: 0x100000
Type: Read / Write

Partial Descriptor 5

Resource: Memory
Disposition: Device Exclusive
Start: 0xffe00000
Length: 0x200000
Type: Read / Write

Key Name: HARDWARE\RESOURCEMAP\VIDEO
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\RESOURCEMAP\VIDEO\cirrus
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Value 0
Name: \Device\Video0.Raw
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0

Interface Type: PCI
Bus Number: 0
Version: 0
Revision: 0

Partial Descriptor 0

Resource: Port
Disposition: Device Exclusive
Start: 0x000003b0
Length: 0xc
Type: Port

Partial Descriptor 1

Resource: Port
Disposition: Device Exclusive
Start: 0x000003c0
Length: 0x20
Type: Port

Partial Descriptor 2

Resource: Memory
Disposition: Device Exclusive
Start: 0x000a0000
Length: 0x20000
Type: Read / Write

Partial Descriptor 3

Resource: Memory
Disposition: Device Exclusive
Start: 0xe6000000
Length: 0x2000000
Type: Read / Write

Value 1

Name: \Device\Video0.Translated
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0

Interface Type: PCI
Bus Number: 0
Version: 0
Revision: 0

Partial Descriptor 0

Resource: Port
Disposition: Device Exclusive
Start: 0x000003b0
Length: 0xc
Type: Port

Partial Descriptor 1

Resource: Port
Disposition: Device Exclusive
Start: 0x000003c0
Length: 0x20
Type: Port

Partial Descriptor 2

Resource: Memory
Disposition: Device Exclusive
Start: 0x000a0000
Length: 0x20000
Type: Read / Write

Partial Descriptor 3

Resource: Memory
Disposition: Device Exclusive
Start: 0xe6000000
Length: 0x2000000
Type: Read / Write

Key Name: HARDWARE\RESOURCEMAP\VIDEO\VgaSave
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: HARDWARE\RESOURCEMAP\VIDEO\VgaStart
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:32 PM

Key Name: SYSTEM\CurrentControlSet\Services\NDIS
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: DisplayName
Type: REG_SZ
Data: Microsoft NDIS System Driver

Value 1
Name: ErrorControl
Type: REG_DWORD
Data: 0x1

Value 2
Name: Group
Type: REG_SZ
Data: NDIS

Value 3
Name: Start
Type: REG_DWORD
Data: 0x1

Value 4
Name: Type
Type: REG_DWORD
Data: 0x1

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\Enum
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 12:31 PM
Value 0
Name: 0
Type: REG_SZ
Data: Root\LEGACY_NDIS\0000

Value 1
Name: Count
Type: REG_DWORD
Data: 0x1

Value 2
Name: NextInstance
Type: REG_DWORD
Data: 0x1

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\MediaTypes
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\NetDetect
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\3C592
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x20596d50

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xf0ffffff

Value 2
Name: token
Type: REG_SZ
Data: 3C592

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\3C597
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x70596d50

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xf0ffffff

Value 2
Name: token
Type: REG_SZ
Data: 3C597

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\BONSAI
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x62110e
Value 1
Name: Mask
Type: REG_DWORD
Data: 0xffffffff
Value 2
Name: token
Type: REG_SZ
Data: BONSAI

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\C320TNT
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x32530e
Value 1
Name: Mask
Type: REG_DWORD
Data: 0xffffffff
Value 2
Name: token
Type: REG_SZ
Data: C320TNT

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\DE425
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x5042a310
Value 1
Name: Mask
Type: REG_DWORD
Data: 0xf0ffffff

Value 2
Name: token
Type: REG_SZ
Data: DE425

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\DEC300
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x230a310

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xffffffff

Value 2
Name: token
Type: REG_SZ
Data: DEC300

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\DEC422
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x2042a310

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xf0ffffff

Value 2
Name: token
Type: REG_SZ
Data: DEC422

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\DURANGO
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id

Type: REG_DWORD
Data: 0x260110e

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xffffffff

Value 2
Name: token
Type: REG_SZ
Data: DURANGO

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\ELNK3EISA
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x90506d50

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xf0ffffff

Value 2
Name: token
Type: REG_SZ
Data: ELNK3EISA

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\ES3210
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x12949

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xffffffff

Value 2
Name: token
Type: REG_SZ
Data: ES3210

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\F70XX
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x6690e

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xffffffff

Value 2
Name: token
Type: REG_SZ
Data: F70XX

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\FL32
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x1010d425

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xffffffff

Value 2
Name: token
Type: REG_SZ
Data: FL32

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\FLNK
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x776d50

Value 1
Name: Mask
Type: REG_DWORD

Data: 0xffffffff

Value 2
 Name: token
 Type: REG_SZ
 Data: FLNK

Key Name:
 SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\J2577A
 Class Name: <NO CLASS>
 Last Write Time: 12/12/97 - 2:11 PM

Value 0
 Name: Id
 Type: REG_DWORD
 Data: 0x4019f022

Value 1
 Name: Mask
 Type: REG_DWORD
 Data: 0xf0ffffff

Value 2
 Name: token
 Type: REG_SZ
 Data: J2577A

Key Name:
 SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\MAPLE
 Class Name: <NO CLASS>
 Last Write Time: 10/10/96 - 9:09 AM

Value 0
 Name: Id
 Type: REG_DWORD
 Data: 0x160110e

Value 1
 Name: Mask
 Type: REG_DWORD
 Data: 0xffffffff

Value 2
 Name: token
 Type: REG_SZ
 Data: MAPLE

Key Name:
 SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\NE3200
 Class Name: <NO CLASS>
 Last Write Time: 10/10/96 - 9:09 AM

Value 0

Name: Id
 Type: REG_DWORD
 Data: 0x7cc3a

Value 1
 Name: Mask
 Type: REG_DWORD
 Data: 0xffffffff

Value 2
 Name: token
 Type: REG_SZ
 Data: NE3200

Key Name:
 SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\NETFLEX3
 Class Name: <NO CLASS>
 Last Write Time: 12/12/97 - 2:11 PM

Value 0
 Name: Id
 Type: REG_DWORD
 Data: 0x20f1110e

Value 1
 Name: Mask
 Type: REG_DWORD
 Data: 0xf0ffffff

Value 2
 Name: token
 Type: REG_SZ
 Data: NETFLEX3

Key Name:
 SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\NETFLEX3.1
 Class Name: <NO CLASS>
 Last Write Time: 12/12/97 - 2:11 PM

Value 0
 Name: Id
 Type: REG_DWORD
 Data: 0x40f1110e

Value 1
 Name: Mask
 Type: REG_DWORD
 Data: 0xf0ffffff

Value 2
 Name: token
 Type: REG_SZ
 Data: NETFLEX3

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\NETFLX
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x61110e

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xffffffff

Value 2
Name: token
Type: REG_SZ
Data: NETFLX

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\NF3500
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x84633a

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xffffffff

Value 2
Name: token
Type: REG_SZ
Data: NF3500

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\NPEISA.1
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM

Value 0
Name: id
Type: REG_DWORD
Data: 0x2093a

Value 1
Name: Mask

Type: REG_DWORD
Data: 0xffffffff

Value 2
Name: token
Type: REG_SZ
Data: NPEISA

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\NPEISA.2
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM

Value 0
Name: id
Type: REG_DWORD
Data: 0x3093a

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xffffffff

Value 2
Name: token
Type: REG_SZ
Data: NPEISA

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\P1990
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x604f42

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xffffffff

Value 2
Name: token
Type: REG_SZ
Data: P1990

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\RODAN
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x63110e

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xffffffff

Value 2
Name: token
Type: REG_SZ
Data: RODAN

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\SKETHNT
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x2644d

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xfffff

Value 2
Name: token
Type: REG_SZ
Data: SKETHNT

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\SKFENT
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x1644d

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xfffff

Value 2
Name: token
Type: REG_SZ

Data: SKFENT

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\SMC8232
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x80a34d

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xffffffff

Value 2
Name: token
Type: REG_SZ
Data: SMC8232

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\TLNK3E
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x9c616d50

Value 1
Name: Mask
Type: REG_DWORD
Data: 0xf0ffffff

Value 2
Name: token
Type: REG_SZ
Data: TLNK3E

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\TLNK3EISA
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x90616d50

Value 1

Name: Mask
 Type: REG_DWORD
 Data: 0xf0ffffff

Value 2
 Name: token
 Type: REG_SZ
 Data: TLNK3EISA

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA
 Class Name: <NO CLASS>
 Last Write Time: 10/10/96 - 9:09 AM

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\AT1700
 Class Name: <NO CLASS>
 Last Write Time: 12/12/97 - 2:11 PM

Value 0
 Name: Id
 Type: REG_DWORD
 Data: 0x6413

Value 1
 Name: token
 Type: REG_SZ
 Data: AT1700

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\EE16MC
 Class Name: <NO CLASS>
 Last Write Time: 10/10/96 - 9:09 AM

Value 0
 Name: Id
 Type: REG_DWORD
 Data: 0x628b

Value 1
 Name: token
 Type: REG_SZ
 Data: EE16MC

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\ElINK527
 Class Name: <NO CLASS>
 Last Write Time: 12/12/97 - 2:11 PM

Value 0
 Name: Id
 Type: REG_DWORD
 Data: 0x41

Value 1
 Name: token
 Type: REG_SZ
 Data: ElINK527

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\ELNK3MCA.1
 Class Name: <NO CLASS>
 Last Write Time: 10/10/96 - 9:09 AM

Value 0
 Name: Id
 Type: REG_DWORD
 Data: 0x627c

Value 1
 Name: token
 Type: REG_SZ
 Data: ELNK3MCA

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\ELNK3MCA.2
 Class Name: <NO CLASS>
 Last Write Time: 10/10/96 - 9:09 AM

Value 0
 Name: Id
 Type: REG_DWORD
 Data: 0x627d

Value 1
 Name: token
 Type: REG_SZ
 Data: ELNK3MCA

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\ELNK3MCA.3
 Class Name: <NO CLASS>
 Last Write Time: 10/10/96 - 9:09 AM

Value 0
 Name: Id
 Type: REG_DWORD
 Data: 0x61db

Value 1
 Name: token
 Type: REG_SZ
 Data: ELNK3MCA

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\ELNK3MCA.4

Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x62f6

Value 1
Name: token
Type: REG_SZ
Data: ELNK3MCA

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\ELNK3MCA.5
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x62f7

Value 1
Name: token
Type: REG_SZ
Data: ELNK3MCA

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\ELNKMC
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x6042

Value 1
Name: token
Type: REG_SZ
Data: ELNKMC

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\F30XX
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x70

Value 1

Name: token
Type: REG_SZ
Data: F30XX

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\HPMCA
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x63ca

Value 1
Name: token
Type: REG_SZ
Data: HPMCA

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\IBMENIIN
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0xffe0

Value 1
Name: token
Type: REG_SZ
Data: IBMENIIN

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\IBMTOKA
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0xe000

Value 1
Name: token
Type: REG_SZ
Data: IBMTOKA

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\IBMTOKMC
Class Name: <NO CLASS>

Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0xe001

Value 1
Name: token
Type: REG_SZ
Data: IBMTOKMC

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\IRMAtrac.1
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x5c1c

Value 1
Name: token
Type: REG_SZ
Data: IRMAtrac

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\IRMAtrac.2
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x5c1d

Value 1
Name: token
Type: REG_SZ
Data: IRMAtrac

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\NCR TOK
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x100

Value 1
Name: token

Type: REG_SZ
Data: NCR TOK

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\NPMCA
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x69

Value 1
Name: token
Type: REG_SZ
Data: NPMCA

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\OCTK16.1
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0xa84

Value 1
Name: token
Type: REG_SZ
Data: OCTK16

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\OCTK16.2
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0xa85

Value 1
Name: token
Type: REG_SZ
Data: OCTK16

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\OCTK16.3
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0xa86

Value 1
Name: token
Type: REG_SZ
Data: OCTK16

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\QUADENET.1
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x8f6d

Value 1
Name: token
Type: REG_SZ
Data: QUADENET

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\QUADENET.2
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x8f6a

Value 1
Name: token
Type: REG_SZ
Data: QUADENET

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\SKFMNT.1
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x83

Value 1
Name: token
Type: REG_SZ

Data: SKFMNT

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\SKFMNT.2
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0xab

Value 1
Name: token
Type: REG_SZ
Data: SKFMNT

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\STREAMER.1
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x8fa0

Value 1
Name: token
Type: REG_SZ
Data: STREAMER

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\STREAMER.2
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x8fa2

Value 1
Name: token
Type: REG_SZ
Data: STREAMER

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\STREAMER.3
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0

Name: Id
Type: REG_DWORD
Data: 0x8fa8

Value 1
Name: token
Type: REG_SZ
Data: STREAMER

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\STREAMER.4
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x8faa

Value 1
Name: token
Type: REG_SZ
Data: STREAMER

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\TC\$4046E
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x51

Value 1
Name: token
Type: REG_SZ
Data: TC\$4046E

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\UBPS
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x7012

Value 1
Name: token
Type: REG_SZ
Data: UBPS

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\WAVELAN_MCA
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x6a14

Value 1
Name: token
Type: REG_SZ
Data: WAVELAN_MCA

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\WD8003EA
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x67c0

Value 1
Name: token
Type: REG_SZ
Data: WD8003EA

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\WD8003WA
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x67c2

Value 1
Name: token
Type: REG_SZ
Data: WD8003WA

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\WD8013EPA
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM

Value 0
Name: Id

Type: REG_DWORD
Data: 0x61c8

Value 1
Name: token
Type: REG_SZ
Data: WD8013EPA

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\MCA\WD8013WPA
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x61c9

Value 1
Name: token
Type: REG_SZ
Data: WD8013WPA

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\3C590
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x590010b7

Value 1
Name: token
Type: REG_SZ
Data: 3C590

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\3C595
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x595010b7

Value 1

Name: token
Type: REG_SZ
Data: 3C595

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\3C905
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x905010b7

Value 1
Name: token
Type: REG_SZ
Data: 3C905

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\ALANE0
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x59009004

Value 1
Name: token
Type: REG_SZ
Data: ALANE0

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\AMDPCI
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x20001022

Value 1
Name: token
Type: REG_SZ
Data: AMDPCI

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\DC21040
Class Name: <NO CLASS>

Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x21011

Value 1
Name: token
Type: REG_SZ
Data: DC21040

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\DC21041
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x141011

Value 1
Name: token
Type: REG_SZ
Data: DC21041

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\DC21140
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x91011

Value 1
Name: token
Type: REG_SZ
Data: DC21140

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\DC21142
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x191011

Value 1
Name: token

Type: REG_SZ
Data: DC21142

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\DEFPA
Class Name: <NO CLASS>
Last Write Time: 10/10/96 - 9:09 AM
Value 0
Name: Id
Type: REG_DWORD
Data: 0xf1011

Value 1
Name: token
Type: REG_SZ
Data: DEFPA

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\E100BPCI
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x12298086

Value 1
Name: token
Type: REG_SZ
Data: E100BPCI

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\E10PCI
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x12268086

Value 1
Name: token
Type: REG_SZ
Data: E10PCI

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\LEC
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x100110b6

Value 1
Name: token
Type: REG_SZ
Data: LEC

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\NCPF
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x111bc

Value 1
Name: token
Type: REG_SZ
Data: NCPF

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\NETFLEX3.1
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0xf1300e11

Value 1
Name: token
Type: REG_SZ
Data: NETFLEX3

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\NETFLEX3.2
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0xae320e11

Value 1
Name: token
Type: REG_SZ

Data: NETFLEX3

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\NETFLEX3.3
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0xae340e11

Value 1
Name: token
Type: REG_SZ
Data: NETFLEX3

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\NETFLEX3.4
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0xae350e11

Value 1
Name: token
Type: REG_SZ
Data: NETFLEX3

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\NETFLEX3.5
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0xae430e11

Value 1
Name: token
Type: REG_SZ
Data: NETFLEX3

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\NETFLEX3.6
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0

Name: Id
Type: REG_DWORD
Data: 0xae400e11

Value 1
Name: token
Type: REG_SZ
Data: NETFLEX3

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\NETFLEX3.7
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0xf1500e11

Value 1
Name: token
Type: REG_SZ
Data: NETFLEX3

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\O100PCI
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x11108d

Value 1
Name: token
Type: REG_SZ
Data: O100PCI

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\OCE4XMP
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM
Value 0
Name: Id
Type: REG_DWORD
Data: 0x13108d

Value 1
Name: token
Type: REG_SZ
Data: OCE4XMP

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\OCTK16
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x1108d

Value 1
Name: token
Type: REG_SZ
Data: OCTK16

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\RNSFDDI
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x22001112

Value 1
Name: token
Type: REG_SZ
Data: RNSFDDI

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\RTL8029
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x802910ec

Value 1
Name: token
Type: REG_SZ
Data: RTL8029

Key Name:
SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\SKFPNT
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id

Type: REG_DWORD
Data: 0x40001148

Value 1
Name: token
Type: REG_SZ
Data: SKFPNT

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\SKTOKNT_PCI
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x42001148

Value 1
Name: token
Type: REG_SZ
Data: SKTOKNT_PCI

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\PCI\STREAMER
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:11 PM

Value 0
Name: Id
Type: REG_DWORD
Data: 0x181014

Value 1
Name: token
Type: REG_SZ
Data: STREAMER

Key Name: SYSTEM\CurrentControlSet\Services\NDIS\Parameters
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 9:23 AM

Value 0
Name: ProcessorAffinityMask
Type: REG_DWORD
Data: 0

Key Name: SOFTWARE\Microsoft\MSSQLServer
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:05 PM

Key Name: SOFTWARE\Microsoft\MSSQLServer\Client
Class Name: <NO CLASS>
Last Write Time: 12/12/97 - 2:05 PM

Key Name: SOFTWARE\Microsoft\MSSQLServer\Client\ConnectTo
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:01 PM

Value 0
Name: DSQUERY
Type: REG_SZ
Data: DBNMPNTW

Key Name: SOFTWARE\Microsoft\MSSQLServer\Client\DB-Lib
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:03 PM

Value 0
Name: AutoAnsiToOem
Type: REG_SZ
Data: ON

Value 1
Name: UseIntlSettings
Type: REG_SZ
Data: ON

Key Name: SOFTWARE\Microsoft\MSSQLServer\Client\TDS
Class Name: <NO CLASS>
Last Write Time: 12/18/98 - 2:56 PM

Value 0
Name: <NO NAME>
Type: REG_SZ
Data: 7.0

Value 1
Name: .
Type: REG_SZ
Data: 7.0

Value 2
Name: SPACELAB
Type: REG_SZ
Data: 7.0

Key Name: SOFTWARE\Microsoft\MSSQLServer\MSSQLServer
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:03 PM

Value 0
Name: AuditLevel
Type: REG_DWORD
Data: 0

Value 1
Name: BackupDirectory
Type: REG_SZ
Data: D:\MSSQL7\BACKUP

Value 2
Name: DefaultCompStyle
Type: REG_SZ
Data: 0

Value 3
Name: DefaultDomain
Type: REG_SZ
Data: SPACELAB

Value 4
Name: DefaultLocaleID
Type: REG_SZ
Data: 8200

Value 5
Name: DefaultLogin
Type: REG_SZ
Data: guest

Value 6
Name: DefaultSortID
Type: REG_SZ
Data: 50

Value 7
Name: ListenOn
Type: REG_MULTI_SZ
Data: SSNMPN70,\\.\pipe\sql\query
SSMSSO70,1433

Value 8
Name: LoginMode
Type: REG_DWORD
Data: 0

Value 9
Name: Map#
Type: REG_SZ
Data: -

Value 10
Name: Map\$
Type: REG_SZ
Data:

Value 11
Name: Map_
Type: REG_SZ
Data: \

Value 12
Name: ResourceMgrID
Type: REG_SZ
Data: {28D2E614-BF67-11D2-986C-0800060D682D}

Value 13
Name: RWSListenAddress
Type: REG_SZ
Data:

Value 14
Name: SetHostName
Type: REG_DWORD
Data: 0

Value 15
Name: Tapeloadwaittime
Type: REG_DWORD
Data: 0xffffffff

Key Name:
SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\CurrentVersion
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:01 PM

Value 0
Name: checksum
Type: REG_BINARY
Data:
00000000 37 37 32 32 63 31 35 38 - 61 65 37 64 34 63 64 37
7722c158ae7d4cd7
00000010 35 30 64 61 30 33 34 62 - 36 30 31 34 32 64 30 66
50da034b60142d0f
00000020 30 32 36 33 34 33 66 37 - 64 36 66 35 61 63 31 65
026343f7d6f5ac1e
00000030 39 37 35 65 39 66 66 36 - 32 30 31 34 65 30 32 36
975e9ff62014e026
00000040 34 36 65 34 30 35 35 31 - 64 66 65 37 34 37 32 38
46e40551dfe74728
00000050 33 32 34 30 37 63 64 66 - 38 39 62 66 64 61 35 34
32407cdf89bfda54
00000060 62 34 35 65 38 32 32 35 - 32 38 30 38 38 33 35 35
b45e822528088355
00000070 38 33 34 36 63 63 37 66 - 64 35 62 65 61 36 31 35
8346cc7fd5bea615
00000080 30 38 66 66 36 64 63 37 - 64 62 63 65 61 38 32 61
08ff6dc7dbcea82a

00000090 30 39 33 64 64 35 34 39 - 65 35 30 39 00
093dd549e509.

Value 1
Name: CurrentVersion
Type: REG_SZ
Data: 7.00.623

Value 2
Name: RegisteredOwner
Type: REG_SZ
Data: TPC-C

Value 3
Name: SerialNumber
Type: REG_DWORD
Data: 0x81560040

Key Name: SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\Parameters
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:01 PM

Value 0
Name: SQLArg0
Type: REG_SZ
Data: -dD:\MSSQL7\data\master.mdf

Value 1
Name: SQLArg1
Type: REG_SZ
Data: -eD:\MSSQL7\log\ERRORLOG

Value 2
Name: SQLArg2
Type: REG_SZ
Data: -lD:\MSSQL7\data\mastlog.ldf

Key Name: SOFTWARE\Microsoft\MSSQLServer\MSSQLServer\RPCNetLib
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:01 PM

Value 0
Name: Security
Type: REG_SZ
Data:

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:03 PM

Value 0
Name: AllowInProcess
Type: REG_DWORD

Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers\ADSDSOObject
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:03 PM

Value 0
Name: AllowInProcess
Type: REG_DWORD
Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers\DTSPackageDSO
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:03 PM

Value 0
Name: AllowInProcess
Type: REG_DWORD
Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers\Microsoft.Jet.OLEDB.4.0
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:03 PM

Value 0
Name: AllowInProcess
Type: REG_DWORD
Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers\MSDAORA
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:03 PM

Value 0
Name: AllowInProcess
Type: REG_DWORD
Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers\MSDASQL
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:03 PM

Value 0
Name: AllowInProcess
Type: REG_DWORD
Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers\MSIDX5
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:03 PM

Value 0

Name: AllowInProcess
 Type: REG_DWORD
 Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers\MSQLImpProv
 Class Name: <NO CLASS>
 Last Write Time: 2/8/99 - 4:03 PM
 Value 0
 Name: AllowInProcess
 Type: REG_DWORD
 Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers\MSSEARCHSQL
 Class Name: <NO CLASS>
 Last Write Time: 2/8/99 - 4:03 PM
 Value 0
 Name: AllowInProcess
 Type: REG_DWORD
 Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers\SQLOLEDB
 Class Name: <NO CLASS>
 Last Write Time: 2/8/99 - 4:03 PM
 Value 0
 Name: AllowInProcess
 Type: REG_DWORD
 Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Replication
 Class Name: <NO CLASS>
 Last Write Time: 2/8/99 - 4:03 PM

Key Name: SOFTWARE\Microsoft\MSSQLServer\Replication\MergeReplicationProvider
 Class Name: <NO CLASS>
 Last Write Time: 2/8/99 - 4:03 PM

Key Name: SOFTWARE\Microsoft\MSSQLServer\Replication\MergeReplicationProvider\7.0
 Class Name: <NO CLASS>
 Last Write Time: 2/8/99 - 4:03 PM

Key Name: SOFTWARE\Microsoft\MSSQLServer\Replication\MergeReplicationProvider\7.0\MsJet
 Class Name: <NO CLASS>
 Last Write Time: 2/8/99 - 4:03 PM
 Value 0
 Name: <NO NAME>

Type: REG_SZ
 Data: {f159cf30-0db4-11d1-b272-00aa00b8de95}

Key Name: SOFTWARE\Microsoft\MSSQLServer\Setup
 Class Name: <NO CLASS>
 Last Write Time: 2/8/99 - 4:01 PM
 Value 0
 Name: SourcePath
 Type: REG_SZ
 Data: Z:\SQL70_623

Value 1
 Name: SQLDataRoot
 Type: REG_SZ
 Data: D:\MSSQL7

Value 2
 Name: SQLPath
 Type: REG_SZ
 Data: D:\MSSQL7

Key Name: SOFTWARE\Microsoft\MSSQLServer\SQL Service Manager
 Class Name: <NO CLASS>
 Last Write Time: 2/8/99 - 4:03 PM
 Value 0
 Name: Action Verify
 Type: REG_DWORD
 Data: 0

Value 1
 Name: DefaultSvc
 Type: REG_SZ
 Data: MSSQLServer

Value 2
 Name: Remote
 Type: REG_DWORD
 Data: 0x1

Value 3
 Name: Services
 Type: REG_MULTI_SZ
 Data: MSSQLServer
 SQLServerAgent
 MSDTC

Key Name: SOFTWARE\Microsoft\MSSQLServer\SQLEW
 Class Name: <NO CLASS>
 Last Write Time: 2/8/99 - 4:03 PM

Key Name: SOFTWARE\Microsoft\MSSQLServer\SQLLEW\Replication
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:03 PM
Value 0
Name: PerfmonFile
Type: REG_SZ
Data: D:\MSSQL7\BINN\REPLMON.PMC

Key Name: SOFTWARE\Microsoft\MSSQLServer\SQLLEW\Wizards
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:03 PM
Value 0
Name: Web Assistant
Type: REG_SZ
Data: D:\MSSQL7\BINN\semwebwz.DLL^WebWizardEntry

Key Name: SOFTWARE\Microsoft\MSSQLServer\SQLServerAgent
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:03 PM
Value 0
Name: DownloadedMaxRows
Type: REG_DWORD
Data: 0x64

Value 1
Name: ErrorLogFile
Type: REG_SZ
Data: D:\MSSQL7\LOG\SQLAGENT.OUT

Value 2
Name: ErrorLoggingLevel
Type: REG_DWORD
Data: 0x3

Value 3
Name: JobHistoryMaxRows
Type: REG_DWORD
Data: 0x3e8

Value 4
Name: JobHistoryMaxRowsPerJob
Type: REG_DWORD
Data: 0x64

Value 5
Name: MSXServerName
Type: REG_SZ
Data:

Value 6

Name: NonAlertableErrors
Type: REG_SZ
Data: 1204,4002

Value 7
Name: RestartSQLServer
Type: REG_DWORD
Data: 0x1

Value 8
Name: ServerHost
Type: REG_SZ
Data:

Value 9
Name: WorkingDirectory
Type: REG_SZ
Data: D:\MSSQL7\JOBS

Key Name: SOFTWARE\Microsoft\MSSQLServer\SQLServerAgent\Subsystems
Class Name: <NO CLASS>
Last Write Time: 2/8/99 - 4:03 PM
Value 0
Name: ActiveScripting
Type: REG_SZ
Data: D:\MSSQL7\BINN\SQLATXSS.DLL, NULL, ActiveScriptStart, ActiveScriptEvent, ActiveScriptStop, 10

Value 1
Name: CmdExec
Type: REG_SZ
Data: D:\MSSQL7\BINN\SQLCMDSS.DLL, NULL, CmdExecStart, CmdEvent, CmdExecStop, 10

Value 2
Name: Distribution
Type: REG_SZ
Data: D:\MSSQL7\BINN\SQLREPSS.DLL, D:\MSSQL7\BINN\DISTRIB.EXE, ReplStart, ReplEvent, ReplStop, 100

Value 3
Name: LogReader
Type: REG_SZ
Data: D:\MSSQL7\BINN\SQLREPSS.DLL, D:\MSSQL7\BINN\LOGREAD.EXE, ReplStart, ReplEvent, ReplStop, 25

Value 4
Name: Merge

Type: REG_SZ
 Data: D:\MSSQL7\BINN\SQLREPSS.DLL,D:\MSSQL7\BINN\REPLMERG.EXE,ReplStart,ReplEvent,ReplStop,100

Value 5
 Name: Snapshot
 Type: REG_SZ
 Data: D:\MSSQL7\BINN\SQLREPSS.DLL,D:\MSSQL7\BINN\SNAPSHOT.EXE,ReplStart,ReplEvent,ReplStop,100

Key Name: SOFTWARE\Microsoft\MSSQLServer\Tracking
 Class Name: <NO CLASS>
 Last Write Time: 2/8/99 - 4:03 PM

Value 0
 Name: {E07FDDA4-5A21-11d2-9DAD-00C04F79D434}
 Type: REG_SZ
 Data:

Value 1
 Name: {E07FDDA8-5A21-11d2-9DAD-00C04F79D434}
 Type: REG_SZ
 Data:

Value 2
 Name: {E07FDDA9-5A21-11d2-9DAD-00C04F79D434}
 Type: REG_SZ
 Data:

Value 3
 Name: {E07FDDAA-5A21-11d2-9DAD-00C04F79D434}
 Type: REG_SZ
 Data:

Value 4
 Name: {E07FDDAC-5A21-11d2-9DAD-00C04F79D434}
 Type: REG_SZ
 Data:

Value 5
 Name: {E07FDDAD-5A21-11d2-9DAD-00C04F79D434}
 Type: REG_SZ
 Data:

Value 6
 Name: {E07FDDAF-5A21-11d2-9DAD-00C04F79D434}
 Type: REG_SZ
 Data:

Value 7
 Name: {E07FDDBE-5A21-11d2-9DAD-00C04F79D434}

Type: REG_SZ
 Data:

Value 8
 Name: {E07FDDBF-5A21-11d2-9DAD-00C04F79D434}
 Type: REG_SZ
 Data:

Value 9
 Name: {E07FDDC0-5A21-11d2-9DAD-00C04F79D434}
 Type: REG_SZ
 Data:

Value 10
 Name: {E07FDDC8-5A21-11d2-9DAD-00C04F79D434}
 Type: REG_SZ
 Data:

This section discloses the Windows NT 4.0 Enterprise Edition registry parameters used on the Primergy 470 client systems.

Key Name: SOFTWARE\Microsoft\TPCC
 Class Name: <NO CLASS>
 Last Write Time: 12/9/97 - 9:21 AM

Value 0
 Name: LOG
 Type: REG_SZ
 Data: OFF

Value 1
 Name: MaxConnections
 Type: REG_SZ
 Data: 10000

Value 2
 Name: MaximumWarehouses
 Type: REG_SZ
 Data: 2000

Value 3
 Name: PATH
 Type: REG_SZ
 Data: C:\InetPub\wwwroot\

```

Key Name:      HARDWARE
Class Name:    <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Key Name:      HARDWARE\DESCRIPTION
Class Name:    <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Key Name:      HARDWARE\DESCRIPTION\System
Class Name:    System
Last Write Time: 3/9/99 - 3:58 PM
Value 0
  Name:        Component Information
  Type:        REG_BINARY
  Data:        00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00
  .....

Value 1
  Name:        Configuration Data
  Type:        REG_FULL_RESOURCE_DESCRIPTOR
  Interface Type:  Invalid
  Bus Number:    -1
  Version:       0
  Revision:      0
  Partial Descriptor 0
  Resource:      Device Specific
  Disposition:   Undetermined
  Reserved1:     0x00000000
  Reserved2:     0x00000000
  Data:         00000000 80 00 fe 03 00 00 3f 00 - 83 00 01 00
  .....?.....

Value 2
  Name:        Identifier
  Type:        REG_SZ
  Data:        AT/AT COMPATIBLE

Value 3
  Name:        SystemBiosDate
  Type:        REG_SZ
  Data:        09/04/98

Value 4
  Name:        SystemBiosVersion
  Type:        REG_MULTI_SZ
  Data:        PhoenixBIOS Version 4.06 Rev. 1.01.1031
  PhoenixBIOS Version 4.06 Rev. 1.01.1031
  PhoenixBIOS Version 4.06 Rev. 1.01.1031

```

```

Value 5
  Name:        VideoBiosDate
  Type:        REG_SZ
  Data:        05/21/97

Value 6
  Name:        VideoBiosVersion
  Type:        REG_MULTI_SZ
  Data:        CL-GD5446 PCI VGA BIOS Version 1.33
  Rel. 1.00
  Version 4.06 Rev. 1.01.1031
  PhoenixBIOS Version 4.06 Rev. 1.01.1031

```

```

Key Name:          HARDWARE\DESCRIPTION\System\CentralProcessor
Class Name:        Processor
Last Write Time:   3/9/99 - 3:58 PM

Key Name:          HARDWARE\DESCRIPTION\System\CentralProcessor\0
Class Name:        Processor
Last Write Time:   3/9/99 - 3:58 PM
Value 0
  Name:            Component Information
  Type:            REG_BINARY
  Data:            00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 01 00 00 00
  .....
```

```

Value 1
  Name:            Configuration Data
  Type:            REG_FULL_RESOURCE_DESCRIPTOR
                  Interface Type:   Invalid
                  Bus Number:      -1
                  Version:         0
                  Revision:        0

Value 2
  Name:            FeatureSet
  Type:            REG_DWORD
  Data:            0x3ff

Value 3
  Name:            Identifier
  Type:            REG_SZ
  Data:            x86 Family 6 Model 5 Stepping 2

Value 4
  Name:            Update Signature
  Type:            REG_BINARY
```

```

Data:
00000000 00 00 00 00 14 00 00 00 - .....
```

```

Value 5
  Name:            Update Status
  Type:            REG_DWORD
  Data:            0x6

Value 6
  Name:            VendorIdentifier
  Type:            REG_SZ
  Data:            GenuineIntel

Value 7
  Name:            ~MHz
  Type:            REG_DWORD
  Data:            0x1be

Key Name:          HARDWARE\DESCRIPTION\System\CentralProcessor\1
Class Name:        Processor
Last Write Time:   3/9/99 - 3:58 PM
Value 0
  Name:            Component Information
  Type:            REG_BINARY
  Data:            00000000 00 00 00 00 00 00 00 00 - 01 00 00 00 02 00 00 00
  .....
```

```

Value 1
  Name:            Configuration Data
  Type:            REG_FULL_RESOURCE_DESCRIPTOR
                  Interface Type:   Invalid
                  Bus Number:      -1
                  Version:         0
                  Revision:        0

Value 2
  Name:            FeatureSet
  Type:            REG_DWORD
  Data:            0x3ff

Value 3
  Name:            Identifier
  Type:            REG_SZ
  Data:            x86 Family 6 Model 5 Stepping 2

Value 4
  Name:            Update Signature
  Type:            REG_BINARY
  Data:            00000000 00 00 00 00 14 00 00 00 - .....
```

```

Value 5
Name: Update Status
Type: REG_DWORD
Data: 0x6

Value 6
Name: VendorIdentifier
Type: REG_SZ
Data: GenuineIntel

Value 7
Name: ~MHz
Type: REG_DWORD
Data: 0x1be

Key Name: HARDWARE\DESCRIPTION\System\FloatingPointProcessor
Class Name: Processor
Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\DESCRIPTION\System\FloatingPointProcessor\0
Class Name: Processor
Last Write Time: 3/9/99 - 3:58 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 01 00 00 00
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Invalid
Bus Number: -1
Version: 0
Revision: 0

Value 2
Name: Identifier
Type: REG_SZ
Data: x86 Family 6 Model 5 Stepping 2

Key Name: HARDWARE\DESCRIPTION\System\FloatingPointProcessor\1
Class Name: Processor
Last Write Time: 3/9/99 - 3:58 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data:

```

```

00000000 00 00 00 00 00 00 00 00 - 01 00 00 00 02 00 00 00
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Invalid
Bus Number: -1
Version: 0
Revision: 0

Value 2
Name: Identifier
Type: REG_SZ
Data: x86 Family 6 Model 5 Stepping 2

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter
Class Name: Adapter
Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\0
Class Name: Adapter
Last Write Time: 3/9/99 - 3:58 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: PCI
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Device Specific
Disposition: Undetermined
Reserved1: 0x00000000
Reserved2: 0x00000000
Data:
00000000 02 10 03 11 .....

Value 2
Name: Identifier
Type: REG_SZ
Data: PCI

```

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\1
 Class Name: Adapter
 Last Write Time: 3/9/99 - 3:58 PM
 Value 0
 Name: Component Information
 Type: REG_BINARY
 Data:
 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff

Value 1
 Name: Configuration Data
 Type: REG_FULL_RESOURCE_DESCRIPTOR
 Interface Type: PCI
 Bus Number: 1
 Version: 0
 Revision: 0

Value 2
 Name: Identifier
 Type: REG_SZ
 Data: PCI

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\2
 Class Name: Adapter
 Last Write Time: 3/9/99 - 3:58 PM
 Value 0
 Name: Component Information
 Type: REG_BINARY
 Data:
 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff

Value 1
 Name: Configuration Data
 Type: REG_FULL_RESOURCE_DESCRIPTOR
 Interface Type: PCI
 Bus Number: 2
 Version: 0
 Revision: 0

Value 2
 Name: Identifier
 Type: REG_SZ
 Data: PCI

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\3

Class Name: Adapter
 Last Write Time: 3/9/99 - 3:58 PM
 Value 0
 Name: Component Information
 Type: REG_BINARY
 Data:
 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff

Value 1
 Name: Configuration Data
 Type: REG_FULL_RESOURCE_DESCRIPTOR
 Interface Type: Internal
 Bus Number: 0
 Version: 0
 Revision: 0
 Partial Descriptor 0
 Resource: Device Specific
 Disposition: Undetermined
 Reserved1: 0x00000000
 Reserved2: 0x00000000
 Data:

00000000 24 50 6e 50 10 21 00 00 - 10 00 04 00 00 6e a6 00
 \$PnP.!.....n..
 00000010 f0 8c a6 00 00 0f 00 00 - 00 00 00 40 00 00 04 00
@....
 00000020 00 1e 00 00 41 d0 0c 02 - 08 80 00 03 00 86 09 00
A.....
 00000030 00 00 00 e0 fe 00 10 00 - 00 79 00 79 00 79 00 46
y.y.y.F
 00000040 00 01 41 d0 0c 02 08 80 - 00 03 00 47 01 80 00 80
 ..A.....G....
 00000050 00 01 01 47 01 72 00 72 - 00 01 02 47 01 92 00 92
 ...G.r.r...G....
 00000060 00 01 01 47 01 b2 00 b2 - 00 01 02 47 01 ea 00 ea
 ...G.....G....
 00000070 00 01 02 86 09 00 00 00 - 00 fe ff 00 00 02 00 79
y
 00000080 00 79 00 79 00 36 00 02 - 41 d0 0c 01 05 00 00 03
 .y.y.6..A.....
 00000090 00 86 09 00 01 00 00 00 - 00 00 00 0a 00 86 09 00

 000000a0 60 00 80 0e 00 00 80 01 - 00 86 09 00 00 00 00 10
 \.....
 000000b0 00 00 00 f0 17 79 00 79 - 00 79 00 2d 00 03 41 d0
y.y.y.-..A.
 000000c0 02 00 08 01 01 03 00 47 - 01 00 00 00 00 01 10 47
G.....G
 000000d0 01 81 00 81 00 01 0f 47 - 01 c0 00 c0 00 01 20 2a
G.....*
 000000e0 10 01 79 00 79 00 79 00 - 25 00 04 41 d0 00 00 08
 ..y.y.y.%.A....

```

000000f0 00 01 03 00 47 01 20 00 - 20 00 01 02 47 01 a0 00 ....G. .
...G...
00000100 a0 00 01 02 22 04 00 79 - 00 79 00 79 00 1d 00 05
....".y.y.y....
00000110 41 d0 01 00 08 02 01 03 - 00 47 01 40 00 40 00 01
A.....G.@.@..
00000120 04 22 01 00 79 00 79 00 - 79 00 1d 00 06 41 d0 0b
.."y.y.y....A..
00000130 00 08 03 01 03 00 47 01 - 70 00 70 00 01 02 22 00
.....G.p.p...".
00000140 01 79 00 79 00 79 00 25 - 00 07 41 d0 03 03 09 00
.y.y.y.%..A....
00000150 00 03 00 47 01 60 00 60 - 00 01 01 47 01 64 00 64
...G.'...'G.d.d
00000160 00 01 01 22 02 00 79 00 - 79 00 79 00 1d 00 08 41
..."y.y.y....A
00000170 d0 0c 04 0b 80 00 03 00 - 47 01 f0 00 f0 00 01 10
.....G.....
00000180 22 00 20 79 00 79 00 79 - 00 1a 00 09 41 d0 08 00 ".
y.y.y....A...
00000190 04 01 00 03 00 47 01 61 - 00 61 00 01 01 79 00 79
.....G.a.a...y.y
000001a0 00 79 00 1a 00 0a 41 d0 - 0a 03 06 04 00 03 00 47
.y....A.....G
000001b0 01 f8 0c f8 0c 01 08 79 - 00 79 00 79 00 2a 00 0b
.....y.y.y.*..
000001c0 41 d0 0c 02 06 01 00 03 - 00 47 01 d0 04 d0 04 01
A.....G.....
000001d0 02 47 01 c0 f0 c0 f0 01 - 40 47 01 b0 f0 b0 f0 01
.G.....@G.....
000001e0 10 79 00 79 00 79 00 1e - 00 0d 41 d0 0c 02 05 00
.y.y.y....A....
000001f0 00 03 00 86 09 00 20 00 - b8 0c 00 00 08 00 00 79 .....
.....y
00000200 00 79 00 79 00 18 00 0e - 41 d0 0f 13 09 02 00 88
.y.y....A.....
00000210 00 22 00 10 79 00 22 00 - 10 79 00 79 00 57 00 11
.."y."..y.y.W..
00000220 41 d0 07 00 01 02 00 90 - 00 47 01 f0 03 f0 03 08
A.....G.....
00000230 06 47 01 f7 03 f7 03 01 - 01 22 40 00 2a 04 00 79
.G....."@.*.y
00000240 00 30 47 01 f0 03 f0 03 - 08 06 47 01 f7 03 f7 03
.OG.....G.....
00000250 01 01 22 40 00 2a 04 00 - 30 47 01 70 03 70 03 08
.."@.*..OG.p.p..
00000260 06 47 01 77 03 77 03 01 - 01 22 40 00 2a 04 00 38
.G.w.w..."@.*.8
00000270 79 00 79 00
y.y.

```

```

Name: Identifier
Type: REG_SZ
Data: PNP BIOS

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4
Class Name: Adapter
Last Write Time: 3/9/99 - 3:58 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0

Value 2
Name: Identifier
Type: REG_SZ
Data: ISA

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\DiskController
Class Name: Controller
Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\DiskController\0
Class Name: Controller
Last Write Time: 3/9/99 - 3:58 PM
Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 64 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
d.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0

```

Value 2

Partial Descriptor 0
Resource: Port
Disposition: Device Exclusive
Start: 0x000003f0
Length: 0x8
Type: Port

Partial Descriptor 1
Resource: Interrupt
Disposition: Undetermined
Vector: 6
Level: 6
Affinity: 0xffffffff
Type: Latched

Partial Descriptor 2
Resource: DMA
Disposition: Undetermined
Channel: 2
Port: 0

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\DiskController\0\DiskPeripheral
Class Name: Peripheral
Last Write Time: 3/9/99 - 3:58 PM

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\DiskController\0\DiskPeripheral\0
Class Name: Peripheral
Last Write Time: 3/9/99 - 3:58 PM

Value 0
Name: Component Information
Type: REG_BINARY
Data:
00000000 60 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
\.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Device Specific
Disposition: Undetermined
Reserved1: 0x00000000

Reserved2: 0x00000000
Data:
00000000 00 02 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00
.....

Value 2
Name: Identifier
Type: REG_SZ
Data: 3bd3f252-21df7336-A

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\DiskController\0\FloppyDiskPeripheral
Class Name: Peripheral
Last Write Time: 3/9/99 - 3:58 PM

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\DiskController\0\FloppyDiskPeripheral\0
Class Name: Peripheral
Last Write Time: 3/9/99 - 3:58 PM

Value 0
Name: Component Information
Type: REG_BINARY
Data:
00000000 00 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Device Specific
Disposition: Undetermined
Reserved1: 0x00000000
Reserved2: 0x00000000
Data:
00000000 02 00 00 00 00 00 00 00 - 00 00 00 00 a0 05 00 00
.....
00000010 00 00 00 00 df 02 25 02 - 12 1b ff 6c f6 0f 05 4f
.....%...l...O
00000020 00

Value 2

Name: Identifier
Type: REG_SZ
Data: FLOPPY1

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\KeyboardController
Class Name: Controller
Last Write Time: 3/9/99 - 3:58 PM

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\KeyboardController\0
Class Name: Controller
Last Write Time: 3/9/99 - 3:58 PM

Value 0
Name: Component Information
Type: REG_BINARY
Data:
00000000 28 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
(.....)

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Port
Disposition: Device Exclusive
Start: 0x00000060
Length: 0x1
Type: Port
Partial Descriptor 1
Resource: Port
Disposition: Device Exclusive
Start: 0x00000064
Length: 0x1
Type: Port
Partial Descriptor 2
Resource: Interrupt
Disposition: Undetermined
Vector: 1
Level: 1
Affinity: 0xffffffff
Type: Latched

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\KeyboardController\0\KeyboardPeripheral
Class Name: Peripheral
Last Write Time: 3/9/99 - 3:58 PM

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\KeyboardController\0\KeyboardPeripheral\0
Class Name: Peripheral
Last Write Time: 3/9/99 - 3:58 PM

Value 0
Name: Component Information
Type: REG_BINARY
Data:
00000000 28 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
(.....)

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Device Specific
Disposition: Undetermined
Reserved1: 0x00000000
Reserved2: 0x00000000
Data:
00000000 00 00 00 00 04 00 20 00 -

Value 2
Name: Identifier
Type: REG_SZ
Data: PCAT_ENHANCED

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\PointerController
Class Name: Controller
Last Write Time: 3/9/99 - 3:58 PM

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\PointerController\0
Class Name: Controller
Last Write Time: 3/9/99 - 3:58 PM

Value 0
Name: Component Information
Type: REG_BINARY

Data:
00000000 20 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Interrupt
Disposition: Undetermined
Vector: 12
Level: 12
Affinity: 0xffffffff
Type: Latched

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\PointerController\0\PointerPeripheral
Class Name: Peripheral
Last Write Time: 3/9/99 - 3:58 PM

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\PointerController\0\PointerPeripheral\0
Class Name: Peripheral
Last Write Time: 3/9/99 - 3:58 PM

Value 0
Name: Component Information
Type: REG_BINARY
Data:
00000000 20 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff
.....

Value 1
Name: Configuration Data
Type: REG_FULL_RESOURCE_DESCRIPTOR
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0

Value 2
Name: Identifier
Type: REG_SZ
Data: MICROSOFT PS2 MOUSE

Key Name: HARDWARE\DEVICEMAP
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\DEVICEMAP\KeyboardClass
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM
Value 0
Name: \Device\KeyboardClass0
Type: REG_SZ
Data:
\REGISTRY\Machine\System\ControlSet001\Services\Kbdclass

Key Name: HARDWARE\DEVICEMAP\KeyboardPort
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM
Value 0
Name: \Device\KeyboardPort0
Type: REG_SZ
Data:
\REGISTRY\Machine\System\ControlSet001\Services\i8042prt

Key Name: HARDWARE\DEVICEMAP\PointerClass
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM
Value 0
Name: \Device\PointerClass0
Type: REG_SZ
Data:
\REGISTRY\Machine\System\ControlSet001\Services\Mouclass

Key Name: HARDWARE\DEVICEMAP\PointerPort
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM
Value 0
Name: \Device\PointerPort0
Type: REG_SZ
Data:
\REGISTRY\Machine\System\ControlSet001\Services\i8042prt

Key Name: HARDWARE\DEVICEMAP\Scsi
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Value 0
 Name: DMAEnabled
 Type: REG_DWORD
 Data: 0x1

Value 1
 Name: Driver
 Type: REG_SZ
 Data: symc8XX

Value 2
 Name: Interrupt
 Type: REG_DWORD
 Data: 0x7

Value 3
 Name: IOAddress
 Type: REG_DWORD
 Data: 0xd800

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Initiator Id 7
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 0\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 3:58 PM

Value 0
 Name: Identifier
 Type: REG_SZ
 Data: WDIGTL ENTERPRISE 1.91

Value 1
 Name: Type
 Type: REG_SZ
 Data: DiskPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 5
 Class Name: <NO CLASS>

Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 5\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 3:58 PM

Value 0
 Name: Identifier
 Type: REG_SZ
 Data: TOSHIBA CD-ROM XM-6201TA1400

Value 1
 Name: Type
 Type: REG_SZ
 Data: CdRomPeripheral

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 8
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 8\Logical Unit Id 0
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 3:58 PM

Value 0
 Name: Identifier
 Type: REG_SZ
 Data: SDR GEM200 2

Value 1
 Name: Type
 Type: REG_SZ
 Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\VIDEO
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 3:59 PM

Value 0
 Name: \Device\Video0
 Type: REG_SZ
 Data: \REGISTRY\Machine\System\ControlSet001\Services\cirrus\Device0

Value 1
 Name: VgaCompatible
 Type: REG_SZ
 Data: \Device\Video0

Key Name: HARDWARE\OWNERMAP

Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 3:59 PM
 Value 0
 Name: PCI_0_c
 Type: REG_SZ
 Data: \Device\E100B1

 Value 1
 Name: PCI_0_d
 Type: REG_SZ
 Data: \Device\E100B2

 Value 2
 Name: PCI_0_e
 Type: REG_SZ
 Data: \Device\E100B3

 Value 3
 Name: PCI_2_0
 Type: REG_SZ
 Data: \Device\Video0

 Value 4
 Name: PCI_2_1
 Type: REG_SZ
 Data: \Device\ScsiPort0

 Value 5
 Name: PCI_2_8
 Type: REG_SZ
 Data: \Device\E100B4

 Value 6
 Name: PCI_2_9
 Type: REG_SZ
 Data: \Device\E100B5

 Value 7
 Name: PCI_2_a
 Type: REG_SZ
 Data: \Device\E100B6

Key Name: HARDWARE\RESOURCEMAP
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\RESOURCEMAP\Hardware Abstraction Layer
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\RESOURCEMAP\Hardware Abstraction Layer\MPS
 1.4 - APIC platform

Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 3:58 PM
 Value 0
 Name: .Raw
 Type: REG_RESOURCE_LIST
 Data:

Full Resource Descriptor 0
 Interface Type: Isa
 Bus Number: 0
 Version: 0
 Revision: 0
 Partial Descriptor 0
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 8
 Level: 8
 Affinity: 0x00000003
 Type: Level Sensitive

Full Resource Descriptor 1
 Interface Type: Internal
 Bus Number: 0
 Version: 0
 Revision: 0
 Partial Descriptor 0
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 0
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 1
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 1
 Level: 1
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 2
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 2
 Level: 2
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 3
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 3

Level: 3
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 4
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 4
Level: 4
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 5
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 5
Level: 5
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 6
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 6
Level: 6
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 7
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 7
Level: 7
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 8
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 8
Level: 8
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 9
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 9
Level: 9
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 10

Resource: Interrupt
Disposition: Driver Exclusive
Vector: 10
Level: 10
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 11
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 11
Level: 11
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 12
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 12
Level: 12
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 13
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 13
Level: 13
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 14
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 14
Level: 14
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 15
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 15
Level: 15
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 16
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 16
Level: 16
Affinity: 0x00000003

Type: Level Sensitive

Partial Descriptor 17
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 17
 Level: 17
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 18
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 18
 Level: 18
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 19
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 19
 Level: 19
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 20
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 20
 Level: 20
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 21
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 21
 Level: 21
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 22
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 22
 Level: 22
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 23
 Resource: Interrupt
 Disposition: Driver Exclusive

Vector: 23
 Level: 23
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 24
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 24
 Level: 24
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 25
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 25
 Level: 25
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 26
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 26
 Level: 26
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 27
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 27
 Level: 27
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 28
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 28
 Level: 28
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 29
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 29
 Level: 29
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 30
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 30
Level: 30
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 31
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 31
Level: 31
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 32
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 32
Level: 32
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 33
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 33
Level: 33
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 34
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 34
Level: 34
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 35
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 35
Level: 35
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 36
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 36
Level: 36

Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 37
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 37
Level: 37
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 38
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 38
Level: 38
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 39
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 39
Level: 39
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 40
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 40
Level: 40
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 41
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 41
Level: 41
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 42
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 42
Level: 42
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 43
Resource: Interrupt

Disposition: Driver Exclusive
 Vector: 43
 Level: 43
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 44
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 44
 Level: 44
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 45
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 45
 Level: 45
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 46
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 46
 Level: 46
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 47
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 47
 Level: 47
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 48
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 61
 Level: 61
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 49
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 65
 Level: 65
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 50
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 80
 Level: 80
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 51
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 193
 Level: 193
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 52
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 225
 Level: 225
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 53
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 253
 Level: 253
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 54
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 254
 Level: 254
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 55
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 255
 Level: 255
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 56
 Resource: Port
 Disposition: Driver Exclusive
 Start: 0x00000000

```

Length:      0x10
Type:       Port

Partial Descriptor 57
Resource:   Port
Disposition: Driver Exclusive
Start:     0x00000020
Length:    0x2
Type:      Port

Partial Descriptor 58
Resource:   Port
Disposition: Driver Exclusive
Start:     0x00000040
Length:    0x4
Type:      Port

Partial Descriptor 59
Resource:   Port
Disposition: Driver Exclusive
Start:     0x00000048
Length:    0x4
Type:      Port

Partial Descriptor 60
Resource:   Port
Disposition: Driver Exclusive
Start:     0x00000061
Length:    0x1
Type:      Port

Partial Descriptor 61
Resource:   Port
Disposition: Driver Exclusive
Start:     0x00000070
Length:    0x2
Type:      Port

Partial Descriptor 62
Resource:   Port
Disposition: Driver Exclusive
Start:     0x00000080
Length:    0x10
Type:      Port

Partial Descriptor 63
Resource:   Port
Disposition: Driver Exclusive
Start:     0x00000092
Length:    0x1
Type:      Port

Partial Descriptor 64

```

```

Value 1
Name:
Type:
Data:

```

```

Resource:    Port
Disposition: Driver Exclusive
Start:      0x000000a0
Length:     0x2
Type:       Port

Partial Descriptor 65
Resource:   Port
Disposition: Driver Exclusive
Start:     0x000000c0
Length:    0x10
Type:      Port

Partial Descriptor 66
Resource:   Port
Disposition: Driver Exclusive
Start:     0x000000f0
Length:    0x10
Type:      Port

Partial Descriptor 67
Resource:   Memory
Disposition: Driver Exclusive
Start:     0xfec00000
Length:    0x400
Type:      Read / Write

Partial Descriptor 68
Resource:   Memory
Disposition: Driver Exclusive
Start:     0xfef00000
Length:    0x400
Type:      Read / Write

.Translated
REG_RESOURCE_LIST

Full Resource Descriptor 0
Interface Type:  Isa
Bus Number:     0
Version:        0
Revision:       0
Partial Descriptor 0
Resource:       Interrupt
Disposition:    Driver Exclusive
Vector:        209
Level:         28
Affinity:      0x00000003
Type:          Level Sensitive

```

Full Resource Descriptor 1
Interface Type: Internal
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 0
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 1
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 1
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 2
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 2
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 3
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 3
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 4
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 4
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 5
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 5
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 6
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 6
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 7
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 7
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 8
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 8
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 9
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 9
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 10
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 10
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 11
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 11
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 12
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 12

Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 13
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 13
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 14
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 14
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 15
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 15
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 16
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 16
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 17
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 17
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 18
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 18
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 19

Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 19
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 20
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 20
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 21
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 21
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 22
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 22
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 23
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 23
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 24
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 24
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 25
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 25
 Level: 0
 Affinity: 0x00000003

Type: Level Sensitive

Partial Descriptor 26
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 26
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 27
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 27
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 28
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 28
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 29
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 29
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 30
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 30
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 31
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 31
 Level: 31
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 32
 Resource: Interrupt
 Disposition: Driver Exclusive

Vector: 32
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 33
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 33
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 34
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 34
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 35
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 35
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 36
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 36
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 37
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 37
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 38
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 38
 Level: 0
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 39
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 39
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 40
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 40
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 41
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 41
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 42
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 42
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 43
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 43
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 44
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 44
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 45
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 45
Level: 0

Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 46
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 46
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 47
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 47
Level: 0
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 48
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 61
Level: 1
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 49
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 65
Level: 2
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 50
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 80
Level: 255
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 51
Resource: Interrupt
Disposition: Driver Exclusive
Vector: 193
Level: 27
Affinity: 0x00000003
Type: Level Sensitive

Partial Descriptor 52
Resource: Interrupt

Disposition: Driver Exclusive
 Vector: 225
 Level: 29
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 53
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 253
 Level: 30
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 54
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 254
 Level: 30
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 55
 Resource: Interrupt
 Disposition: Driver Exclusive
 Vector: 255
 Level: 31
 Affinity: 0x00000003
 Type: Level Sensitive

Partial Descriptor 56
 Resource: Port
 Disposition: Driver Exclusive
 Start: 0x00000000
 Length: 0x10
 Type: Port

Partial Descriptor 57
 Resource: Port
 Disposition: Driver Exclusive
 Start: 0x00000020
 Length: 0x2
 Type: Port

Partial Descriptor 58
 Resource: Port
 Disposition: Driver Exclusive
 Start: 0x00000040
 Length: 0x4
 Type: Port

Partial Descriptor 59
 Resource: Port

Disposition: Driver Exclusive
 Start: 0x00000048
 Length: 0x4
 Type: Port

Partial Descriptor 60
 Resource: Port
 Disposition: Driver Exclusive
 Start: 0x00000061
 Length: 0x1
 Type: Port

Partial Descriptor 61
 Resource: Port
 Disposition: Driver Exclusive
 Start: 0x00000070
 Length: 0x2
 Type: Port

Partial Descriptor 62
 Resource: Port
 Disposition: Driver Exclusive
 Start: 0x00000080
 Length: 0x10
 Type: Port

Partial Descriptor 63
 Resource: Port
 Disposition: Driver Exclusive
 Start: 0x00000092
 Length: 0x1
 Type: Port

Partial Descriptor 64
 Resource: Port
 Disposition: Driver Exclusive
 Start: 0x000000a0
 Length: 0x2
 Type: Port

Partial Descriptor 65
 Resource: Port
 Disposition: Driver Exclusive
 Start: 0x000000c0
 Length: 0x10
 Type: Port

Partial Descriptor 66
 Resource: Port
 Disposition: Driver Exclusive
 Start: 0x000000f0
 Length: 0x10
 Type: Port

Partial Descriptor 67
Resource: Memory
Disposition: Driver Exclusive
Start: 0xfec00000
Length: 0x400
Type: Read / Write

Partial Descriptor 68
Resource: Memory
Disposition: Driver Exclusive
Start: 0xfe000000
Length: 0x400
Type: Read / Write

Key Name: HARDWARE\RESOURCEMAP\KeyboardPort/PointerPort
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\RESOURCEMAP\KeyboardPort/PointerPort\i8042prt
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Value 0
Name: \Device\KeyboardPort0.Raw
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Interrupt
Disposition: Device Exclusive
Vector: 1
Level: 1
Affinity: 0xffffffff
Type: Latched

Partial Descriptor 1
Resource: Interrupt
Disposition: Device Exclusive
Vector: 12
Level: 12
Affinity: 0xffffffff
Type: Latched

Partial Descriptor 2
Resource: Port
Disposition: Driver Exclusive

Start: 0x00000060
Length: 0x1
Type: Port

Partial Descriptor 3
Resource: Port
Disposition: Driver Exclusive
Start: 0x00000064
Length: 0x1
Type: Port

Value 1
Name: \Device\KeyboardPort0.Translated
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Interrupt
Disposition: Device Exclusive
Vector: 97
Level: 5
Affinity: 0x00000003
Type: Latched

Partial Descriptor 1
Resource: Interrupt
Disposition: Device Exclusive
Vector: 113
Level: 6
Affinity: 0x00000003
Type: Latched

Partial Descriptor 2
Resource: Port
Disposition: Driver Exclusive
Start: 0x00000060
Length: 0x1
Type: Port

Partial Descriptor 3
Resource: Port
Disposition: Driver Exclusive
Start: 0x00000064
Length: 0x1
Type: Port

Key Name: HARDWARE\RESOURCEMAP\OtherDrivers
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\RESOURCEMAP\OtherDrivers\E100B
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:59 PM

Value 0
Name: \Device\E100B1.Raw
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: PCI
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Memory
Disposition: Device Exclusive
Start: 0xfecfd000
Length: 0x1e
Type: Read / Write

Partial Descriptor 1
Resource: Port
Disposition: Device Exclusive
Start: 0x0000e400
Length: 0x1e
Type: Port

Partial Descriptor 2
Resource: Interrupt
Disposition: Shared
Vector: 3
Level: 3
Affinity: 0x00000000
Type: Level Sensitive

Value 1
Name: \Device\E100B1.Translated
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: PCI
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Memory

Disposition: Device Exclusive
Start: 0xfecfd000
Length: 0x1e
Type: Read / Write

Partial Descriptor 1
Resource: Port
Disposition: Device Exclusive
Start: 0x0000e400
Length: 0x1e
Type: Port

Partial Descriptor 2
Resource: Interrupt
Disposition: Shared
Vector: 177
Level: 10
Affinity: 0x00000003
Type: Level Sensitive

Value 2
Name: \Device\E100B2.Raw
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: PCI
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Memory
Disposition: Device Exclusive
Start: 0xfecfc000
Length: 0x1e
Type: Read / Write

Partial Descriptor 1
Resource: Port
Disposition: Device Exclusive
Start: 0x0000e000
Length: 0x1e
Type: Port

Partial Descriptor 2
Resource: Interrupt
Disposition: Shared
Vector: 4
Level: 4
Affinity: 0x00000000
Type: Level Sensitive

Value 3
Name:
Type:
Data:

```
\Device\E100B2.Translated
REG_RESOURCE_LIST

Full Resource Descriptor 0
  Interface Type:  PCI
  Bus Number:      0
  Version:         0
  Revision:        0
  Partial Descriptor 0
    Resource:       Memory
    Disposition:    Device Exclusive
    Start:          0xfecfc000
    Length:         0x1e
    Type:           Read / Write

  Partial Descriptor 1
    Resource:       Port
    Disposition:    Device Exclusive
    Start:          0x0000e000
    Length:         0x1e
    Type:           Port

  Partial Descriptor 2
    Resource:       Interrupt
    Disposition:    Shared
    Vector:         161
    Level:          9
    Affinity:       0x00000003
    Type:           Level Sensitive
```

Value 4
Name:
Type:
Data:

```
\Device\E100B3.Raw
REG_RESOURCE_LIST

Full Resource Descriptor 0
  Interface Type:  PCI
  Bus Number:      0
  Version:         0
  Revision:        0
  Partial Descriptor 0
    Resource:       Memory
    Disposition:    Device Exclusive
    Start:          0xfecfe000
    Length:         0x1e
    Type:           Read / Write

  Partial Descriptor 1
    Resource:       Port
```

```
Disposition:  Device Exclusive
Start:        0x0000e800
Length:       0x1e
Type:         Port
```

```
Partial Descriptor 2
Resource:     Interrupt
Disposition:  Shared
Vector:       10
Level:        10
Affinity:     0x00000000
Type:         Level Sensitive
```

Value 5
Name:
Type:
Data:

```
\Device\E100B3.Translated
REG_RESOURCE_LIST

Full Resource Descriptor 0
  Interface Type:  PCI
  Bus Number:      0
  Version:         0
  Revision:        0
  Partial Descriptor 0
    Resource:       Memory
    Disposition:    Device Exclusive
    Start:          0xfecfe000
    Length:         0x1e
    Type:           Read / Write

  Partial Descriptor 1
    Resource:       Port
    Disposition:    Device Exclusive
    Start:          0x0000e800
    Length:         0x1e
    Type:           Port

  Partial Descriptor 2
    Resource:       Interrupt
    Disposition:    Shared
    Vector:         178
    Level:          10
    Affinity:       0x00000003
    Type:           Level Sensitive
```

Value 6
Name:
Type:
Data:

```
\Device\E100B4.Raw
REG_RESOURCE_LIST

Full Resource Descriptor 0
```

```

Interface Type:  PCI
Bus Number:     2
Version:        0
Revision:       0
Partial Descriptor 0
  Resource:      Memory
  Disposition:   Device Exclusive
  Start:         0xf7ffe000
  Length:        0x1e
  Type:          Read / Write

Partial Descriptor 1
  Resource:      Port
  Disposition:   Device Exclusive
  Start:         0x0000d400
  Length:        0x1e
  Type:          Port

Partial Descriptor 2
  Resource:      Interrupt
  Disposition:   Shared
  Vector:        15
  Level:         15
  Affinity:      0x00000000
  Type:          Level Sensitive

```

```

Value 8
Name:
Type:
Data:

```

```

Disposition:    Shared
Vector:         162
Level:          9
Affinity:       0x00000003
Type:           Level Sensitive

```

```

Name:           \Device\E100B5.Raw
Type:           REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
  Interface Type:  PCI
  Bus Number:     2
  Version:        0
  Revision:       0
  Partial Descriptor 0
    Resource:      Memory
    Disposition:   Device Exclusive
    Start:         0xf7ffd000
    Length:        0x1e
    Type:          Read / Write

  Partial Descriptor 1
    Resource:      Port
    Disposition:   Device Exclusive
    Start:         0x0000d000
    Length:        0x1e
    Type:          Port

  Partial Descriptor 2
    Resource:      Interrupt
    Disposition:   Shared
    Vector:        5
    Level:         5
    Affinity:      0x00000000
    Type:          Level Sensitive

```

```

Value 7
Name:
Type:
Data:

```

```

Name:           \Device\E100B4.Translated
Type:           REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
  Interface Type:  PCI
  Bus Number:     2
  Version:        0
  Revision:       0
  Partial Descriptor 0
    Resource:      Memory
    Disposition:   Device Exclusive
    Start:         0xf7ffe000
    Length:        0x1e
    Type:          Read / Write

  Partial Descriptor 1
    Resource:      Port
    Disposition:   Device Exclusive
    Start:         0x0000d400
    Length:        0x1e
    Type:          Port

  Partial Descriptor 2
    Resource:      Interrupt

```

```

Value 9
Name:
Type:
Data:

```

```

Name:           \Device\E100B5.Translated
Type:           REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
  Interface Type:  PCI
  Bus Number:     2
  Version:        0
  Revision:       0
  Partial Descriptor 0
    Resource:      Memory
    Disposition:   Device Exclusive

```

Start: 0xf7ffd000
Length: 0x1e
Type: Read / Write

Partial Descriptor 1
Resource: Port
Disposition: Device Exclusive
Start: 0x0000d000
Length: 0x1e
Type: Port

Partial Descriptor 2
Resource: Interrupt
Disposition: Shared
Vector: 146
Level: 8
Affinity: 0x00000003
Type: Level Sensitive

Value 10
Name:
Type:
Data:

\Device\E100B6.Raw
REG_RESOURCE_LIST

Full Resource Descriptor 0
Interface Type: PCI
Bus Number: 2
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Memory
Disposition: Device Exclusive
Start: 0xf7ffc000
Length: 0x1e
Type: Read / Write

Partial Descriptor 1
Resource: Port
Disposition: Device Exclusive
Start: 0x0000c800
Length: 0x1e
Type: Port

Partial Descriptor 2
Resource: Interrupt
Disposition: Shared
Vector: 14
Level: 14
Affinity: 0x00000000
Type: Level Sensitive

Value 11
Name:
Type:
Data:

\Device\E100B6.Translated
REG_RESOURCE_LIST

Full Resource Descriptor 0
Interface Type: PCI
Bus Number: 2
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Memory
Disposition: Device Exclusive
Start: 0xf7ffc000
Length: 0x1e
Type: Read / Write

Partial Descriptor 1
Resource: Port
Disposition: Device Exclusive
Start: 0x0000c800
Length: 0x1e
Type: Port

Partial Descriptor 2
Resource: Interrupt
Disposition: Shared
Vector: 145
Level: 8
Affinity: 0x00000003
Type: Level Sensitive

Key Name: HARDWARE\RESOURCEMAP\OtherDrivers\Floppy
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Value 0

Name: .Raw
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Port
Disposition: Shared
Start: 0x000003f0
Length: 0x6
Type: Port

Partial Descriptor 1
Resource: Port
Disposition: Shared
Start: 0x000003f7
Length: 0x1
Type: Port

Partial Descriptor 2
Resource: DMA
Disposition: Shared
Channel: 2
Port: 0

Partial Descriptor 3
Resource: Interrupt
Disposition: Shared
Vector: 6
Level: 6
Affinity: 0x00000000
Type: Latched

Value 1
Name: .Translated
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: Isa
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Port
Disposition: Shared
Start: 0x000003f0
Length: 0x6
Type: Port
Partial Descriptor 1
Resource: Port
Disposition: Shared
Start: 0x000003f7
Length: 0x1
Type: Port
Partial Descriptor 2
Resource: DMA
Disposition: Shared
Channel: 2
Port: 0

Partial Descriptor 3
Resource: Interrupt
Disposition: Shared
Vector: 129
Level: 7
Affinity: 0x00000003
Type: Latched

Key Name: HARDWARE\RESOURCEMAP\ScsiAdapter
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\RESOURCEMAP\ScsiAdapter\symc8XX
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM
Value 0

Name: \Device\ScsiPort0.Raw
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: PCI
Bus Number: 2
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Interrupt
Disposition: Shared
Vector: 7
Level: 7
Affinity: 0x00000000
Type: Level Sensitive

Partial Descriptor 1
Resource: Port
Disposition: Device Exclusive
Start: 0x0000d800
Length: 0x100
Type: Port

Partial Descriptor 2
Resource: Memory
Disposition: Device Exclusive
Start: 0xfe8ff800
Length: 0x100
Type: Read / Write

Partial Descriptor 3
Resource: Memory
Disposition: Device Exclusive
Start: 0xfe8fd000

Length: 0x1000
Type: Read / Write

Value 0
Name: .Translated
Type: REG_RESOURCE_LIST
Data:

Value 1
Name:
Type:
Data:

\Device\ScsiPort0.Translated
REG_RESOURCE_LIST
Full Resource Descriptor 0
Interface Type: PCI
Bus Number: 2
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Interrupt
Disposition: Shared
Vector: 81
Level: 4
Affinity: 0x00000003
Type: Level Sensitive
Partial Descriptor 1
Resource: Port
Disposition: Device Exclusive
Start: 0x0000d800
Length: 0x100
Type: Port
Partial Descriptor 2
Resource: Memory
Disposition: Device Exclusive
Start: 0xfe8ff800
Length: 0x100
Type: Read / Write
Partial Descriptor 3
Resource: Memory
Disposition: Device Exclusive
Start: 0xfe8fd000
Length: 0x1000
Type: Read / Write

Full Resource Descriptor 0
Interface Type: Internal
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Memory
Disposition: Device Exclusive
Start: 0x00001000
Length: 0x9e000
Type: Read / Write
Partial Descriptor 1
Resource: Memory
Disposition: Device Exclusive
Start: 0x00100000
Length: 0xeff000
Type: Read / Write
Partial Descriptor 2
Resource: Memory
Disposition: Device Exclusive
Start: 0x01000000
Length: 0x17000000
Type: Read / Write

Key Name: HARDWARE\RESOURCEMAP\System Resources\Reserved
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM
Value 0
Name: .Translated
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: Internal
Bus Number: 0
Version: 0
Revision: 0
Partial Descriptor 0
Resource: Memory
Disposition: Device Exclusive
Start: 0x0009f000
Length: 0x1000
Type: Read / Write
Partial Descriptor 1

Key Name: HARDWARE\RESOURCEMAP\System Resources
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\RESOURCEMAP\System Resources\Physical Memory
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Resource: Memory
Disposition: Device Exclusive
Start: 0x000f0000
Length: 0x10000
Type: Read / Write

Partial Descriptor 2

Resource: Memory
Disposition: Device Exclusive
Start: 0x00fff000
Length: 0x1000
Type: Read / Write

Key Name: HARDWARE\RESOURCEMAP\VIDEO
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\RESOURCEMAP\VIDEO\cirrus
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Value 0
Name: \Device\Video0.Raw
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0

Interface Type: PCI
Bus Number: 2
Version: 0
Revision: 0

Partial Descriptor 0

Resource: Port
Disposition: Device Exclusive
Start: 0x000003b0
Length: 0xc
Type: Port

Partial Descriptor 1

Resource: Port
Disposition: Device Exclusive
Start: 0x000003c0
Length: 0x20
Type: Port

Partial Descriptor 2

Resource: Memory
Disposition: Device Exclusive
Start: 0x000a0000
Length: 0x20000
Type: Read / Write

Partial Descriptor 3

Resource: Memory
Disposition: Device Exclusive
Start: 0xf4000000
Length: 0x2000000
Type: Read / Write

Value 1

Name: \Device\Video0.Translated
Type: REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0

Interface Type: PCI
Bus Number: 2
Version: 0
Revision: 0

Partial Descriptor 0

Resource: Port
Disposition: Device Exclusive
Start: 0x000003b0
Length: 0xc
Type: Port

Partial Descriptor 1

Resource: Port
Disposition: Device Exclusive
Start: 0x000003c0
Length: 0x20
Type: Port

Partial Descriptor 2

Resource: Memory
Disposition: Device Exclusive
Start: 0x000a0000
Length: 0x20000
Type: Read / Write

Partial Descriptor 3

Resource: Memory
Disposition: Device Exclusive
Start: 0xf4000000
Length: 0x2000000
Type: Read / Write

Key Name: HARDWARE\RESOURCEMAP\VIDEO\VgaSave
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Key Name: HARDWARE\RESOURCEMAP\VIDEO\VgaStart
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Key Name: SYSTEM\CurrentControlSet\Services\InetInfo
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:01 PM

Key Name: SYSTEM\CurrentControlSet\Services\InetInfo\Parameters
Class Name: <NO CLASS>
Last Write Time: 12/11/97 - 2:22 PM

Value 0
Name: BandwidthLevel
Type: REG_DWORD
Data: 0xffffffff

Value 1
Name: ListenBackLog
Type: REG_DWORD
Data: 0x1400

Value 2
Name: PoolThreadLimit
Type: REG_DWORD
Data: 0xc8

Value 3
Name: PoolThreadsLimit
Type: REG_DWORD
Data: 0x2fe

Value 4
Name: ThreadTimeout
Type: REG_DWORD
Data: 0x1c20

Key Name: SYSTEM\CurrentControlSet\Services\InetInfo\Parameters\Filter
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:01 PM

Value 0
Name: FilterType
Type: REG_DWORD
Data: 0

Value 1
Name: NumDenySites
Type: REG_DWORD
Data: 0

Value 2
Name: NumGrantSites
Type: REG_DWORD
Data: 0

Key Name: SYSTEM\CurrentControlSet\Services\InetInfo\Parameters\MimeMap
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:01 PM

Value 0
Name: application/envoy,envy,,5
Type: REG_SZ
Data:

Value 1
Name: application/mac-binhex40,hqx,,4
Type: REG_SZ
Data:

Value 2
Name: application/msword,doc,,5
Type: REG_SZ
Data:

Value 3
Name: application/msword,dot,,5
Type: REG_SZ
Data:

Value 4
Name: application/octet-stream,*,,5
Type: REG_SZ
Data:

Value 5
Name: application/octet-stream,bin,,5
Type: REG_SZ
Data:

Value 6
Name: application/octet-stream,exe,,5
Type: REG_SZ
Data:

Value 7
Name: application/oda,oda,,5
Type: REG_SZ
Data:

Value 8
Name: application/pdf,pdf,,5
Type: REG_SZ

Data:		Name:	application/x-director,dxr,,5
Value 9		Type:	REG_SZ
Name:	application/postscript,ai,,5	Data:	
Type:	REG_SZ	Value 20	
Data:		Name:	application/x-dvi,dvi,,5
Value 10		Type:	REG_SZ
Name:	application/postscript,eps,,5	Data:	
Type:	REG_SZ	Value 21	
Data:		Name:	application/x-gtar,gtar,,9
Value 11		Type:	REG_SZ
Name:	application/postscript,ps,,5	Data:	
Type:	REG_SZ	Value 22	
Data:		Name:	application/x-hdf,hdf,,5
Value 12		Type:	REG_SZ
Name:	application/rtf,rtf,,5	Data:	
Type:	REG_SZ	Value 23	
Data:		Name:	application/x-latex,latex,,5
Value 13		Type:	REG_SZ
Name:	application/winhelp,hlp,,5	Data:	
Type:	REG_SZ	Value 24	
Data:		Name:	application/x-msaccess,mdb,,5
Value 14		Type:	REG_SZ
Name:	application/x-bcpio,bcpio,,5	Data:	
Type:	REG_SZ	Value 25	
Data:		Name:	application/x-mscardfile,crd,,5
Value 15		Type:	REG_SZ
Name:	application/x-cpio,cpio,,5	Data:	
Type:	REG_SZ	Value 26	
Data:		Name:	application/x-msclip,clip,,5
Value 16		Type:	REG_SZ
Name:	application/x-csh,csh,,5	Data:	
Type:	REG_SZ	Value 27	
Data:		Name:	application/x-msexcel,xla,,5
Value 17		Type:	REG_SZ
Name:	application/x-director,dcr,,5	Data:	
Type:	REG_SZ	Value 28	
Data:		Name:	application/x-msexcel,xlc,,5
Value 18		Type:	REG_SZ
Name:	application/x-director,dir,,5	Data:	
Type:	REG_SZ	Value 29	
Data:		Name:	application/x-msexcel,xlm,,5
Value 19		Type:	REG_SZ
		Data:	

Value 30
 Name: application/x-msexcel,xls,,5
 Type: REG_SZ
 Data:

Value 31
 Name: application/x-msexcel,xlt,,5
 Type: REG_SZ
 Data:

Value 32
 Name: application/x-msexcel,xlw,,5
 Type: REG_SZ
 Data:

Value 33
 Name: application/x-msmediaview,m13,,5
 Type: REG_SZ
 Data:

Value 34
 Name: application/x-msmediaview,m14,,5
 Type: REG_SZ
 Data:

Value 35
 Name: application/x-msmetafile,wmf,,5
 Type: REG_SZ
 Data:

Value 36
 Name: application/x-msmoney,mny,,5
 Type: REG_SZ
 Data:

Value 37
 Name: application/x-mspowerpoint,ppt,,5
 Type: REG_SZ
 Data:

Value 38
 Name: application/x-msproject,mpp,,5
 Type: REG_SZ
 Data:

Value 39
 Name: application/x-mspublisher,pub,,5
 Type: REG_SZ
 Data:

Value 40
 Name: application/x-msterminal,term,,5

Type: REG_SZ
 Data:

Value 41
 Name: application/x-msworks,wks,,5
 Type: REG_SZ
 Data:

Value 42
 Name: application/x-mswrite,wri,,5
 Type: REG_SZ
 Data:

Value 43
 Name: application/x-netcdf,cdf,,5
 Type: REG_SZ
 Data:

Value 44
 Name: application/x-netcdf,nc,,5
 Type: REG_SZ
 Data:

Value 45
 Name: application/x-perfmon,pma,,5
 Type: REG_SZ
 Data:

Value 46
 Name: application/x-perfmon,pmc,,5
 Type: REG_SZ
 Data:

Value 47
 Name: application/x-perfmon,pml,,5
 Type: REG_SZ
 Data:

Value 48
 Name: application/x-perfmon,pmr,,5
 Type: REG_SZ
 Data:

Value 49
 Name: application/x-perfmon,pmw,,5
 Type: REG_SZ
 Data:

Value 50
 Name: application/x-sh,sh,,5
 Type: REG_SZ
 Data:

Value 51
 Name: application/x-shar,shar,,5
 Type: REG_SZ
 Data:

Value 52
 Name: application/x-sv4cpio,sv4cpio,,5
 Type: REG_SZ
 Data:

Value 53
 Name: application/x-sv4crc,sv4crc,,5
 Type: REG_SZ
 Data:

Value 54
 Name: application/x-tar,tar,,5
 Type: REG_SZ
 Data:

Value 55
 Name: application/x-tcl,tcl,,5
 Type: REG_SZ
 Data:

Value 56
 Name: application/x-tex,tex,,5
 Type: REG_SZ
 Data:

Value 57
 Name: application/x-texinfo,texi,,5
 Type: REG_SZ
 Data:

Value 58
 Name: application/x-texinfo,texinfo,,5
 Type: REG_SZ
 Data:

Value 59
 Name: application/x-troff,roff,,5
 Type: REG_SZ
 Data:

Value 60
 Name: application/x-troff,t,,5
 Type: REG_SZ
 Data:

Value 61
 Name: application/x-troff,tr,,5
 Type: REG_SZ

Data:

Value 62
 Name: application/x-troff-man,man,,5
 Type: REG_SZ
 Data:

Value 63
 Name: application/x-troff-me,me,,5
 Type: REG_SZ
 Data:

Value 64
 Name: application/x-troff-ms,ms,,5
 Type: REG_SZ
 Data:

Value 65
 Name: application/x-ustar,ustar,,5
 Type: REG_SZ
 Data:

Value 66
 Name: application/x-wais-source,src,,7
 Type: REG_SZ
 Data:

Value 67
 Name: application/zip,zip,,9
 Type: REG_SZ
 Data:

Value 68
 Name: audio/basic,au,,<
 Type: REG_SZ
 Data:

Value 69
 Name: audio/basic,snd,,<
 Type: REG_SZ
 Data:

Value 70
 Name: audio/x-aiff,aif,,<
 Type: REG_SZ
 Data:

Value 71
 Name: audio/x-aiff,aifc,,<
 Type: REG_SZ
 Data:

Value 72

Name: audio/x-aiff,aiff,,<
Type: REG_SZ
Data:

Value 73
Name: audio/x-pn-realaudio,ram,,<
Type: REG_SZ
Data:

Value 74
Name: audio/x-wav,wav,,<
Type: REG_SZ
Data:

Value 75
Name: image/bmp,bmp,,:
Type: REG_SZ
Data:

Value 76
Name: image/cis-cod,cod,,5
Type: REG_SZ
Data:

Value 77
Name: image/gif,gif,,g
Type: REG_SZ
Data:

Value 78
Name: image/ief,ief,,:
Type: REG_SZ
Data:

Value 79
Name: image/jpeg,jpe,,:
Type: REG_SZ
Data:

Value 80
Name: image/jpeg,jpeg,,:
Type: REG_SZ
Data:

Value 81
Name: image/jpeg,jpg,,:
Type: REG_SZ
Data:

Value 82
Name: image/tiff,tif,,:
Type: REG_SZ
Data:

Value 83
Name: image/tiff,tiff,,:
Type: REG_SZ
Data:

Value 84
Name: image/x-cmu-raster,ras,,:
Type: REG_SZ
Data:

Value 85
Name: image/x-cmx,cmx,,5
Type: REG_SZ
Data:

Value 86
Name: image/x-portable-anymap,pnm,,:
Type: REG_SZ
Data:

Value 87
Name: image/x-portable-bitmap,pbm,,:
Type: REG_SZ
Data:

Value 88
Name: image/x-portable-graymap,pgm,,:
Type: REG_SZ
Data:

Value 89
Name: image/x-portable-pixmap,ppm,,:
Type: REG_SZ
Data:

Value 90
Name: image/x-rgb,rgb,,:
Type: REG_SZ
Data:

Value 91
Name: image/x-xbitmap,xbm,,:
Type: REG_SZ
Data:

Value 92
Name: image/x-ypixmap,xpm,,:
Type: REG_SZ
Data:

Value 93
Name: image/x-xwindowdump,xwd,,:

Type: REG_SZ
Data:

Value 94
Name: text/html,htm,,h
Type: REG_SZ
Data:

Value 95
Name: text/html,html,,h
Type: REG_SZ
Data:

Value 96
Name: text/html,stm,,h
Type: REG_SZ
Data:

Value 97
Name: text/plain,bas,,0
Type: REG_SZ
Data:

Value 98
Name: text/plain,c,,0
Type: REG_SZ
Data:

Value 99
Name: text/plain,h,,0
Type: REG_SZ
Data:

Value 100
Name: text/plain,txt,,0
Type: REG_SZ
Data:

Value 101
Name: text/richtext,rtx,,0
Type: REG_SZ
Data:

Value 102
Name: text/tab-separated-values,tsv,,0
Type: REG_SZ
Data:

Value 103
Name: text/x-setext,etx,,0
Type: REG_SZ
Data:

Value 104
Name: video/mpeg,mpe,,;
Type: REG_SZ
Data:

Value 105
Name: video/mpeg,mpeg,,;
Type: REG_SZ
Data:

Value 106
Name: video/mpeg,mpg,,;
Type: REG_SZ
Data:

Value 107
Name: video/quicktime,mov,,;
Type: REG_SZ
Data:

Value 108
Name: video/quicktime,qt,,;
Type: REG_SZ
Data:

Value 109
Name: video/x-msvideo,avi,,<
Type: REG_SZ
Data:

Value 110
Name: video/x-sgi-movie,movie,,<
Type: REG_SZ
Data:

Value 111
Name: x-world/x-vrml,flr,,5
Type: REG_SZ
Data:

Value 112
Name: x-world/x-vrml,wrl,,5
Type: REG_SZ
Data:

Value 113
Name: x-world/x-vrml,wrz,,5
Type: REG_SZ
Data:

Value 114
Name: x-world/x-vrml,xaf,,5
Type: REG_SZ

Data:

Value 115
 Name: x-world/x-vrml,xof,,5
 Type: REG_SZ
 Data:

Key Name: SYSTEM\CurrentControlSet\Services\InetInfo\Performance
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:01 PM

Value 0
 Name: Close
 Type: REG_SZ
 Data: CloseINFOPerformanceData

Value 1
 Name: Collect
 Type: REG_SZ
 Data: CollectINFOPerformanceData

Value 2
 Name: First Counter
 Type: REG_DWORD
 Data: 0x738

Value 3
 Name: First Help
 Type: REG_DWORD
 Data: 0x739

Value 4
 Name: Last Counter
 Type: REG_DWORD
 Data: 0x756

Value 5
 Name: Last Help
 Type: REG_DWORD
 Data: 0x757

Value 6
 Name: Library
 Type: REG_SZ
 Data: infoctrs.DLL

Value 7
 Name: Open
 Type: REG_SZ
 Data: OpenINFOPerformanceData

Key Name: SYSTEM\CurrentControlSet\Services\Tcpip
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:01 PM

Value 0
 Name: DisplayName
 Type: REG_SZ
 Data: TCP/IP Service

Value 1
 Name: ErrorControl
 Type: REG_DWORD
 Data: 0x1

Value 2
 Name: Group
 Type: REG_SZ
 Data: PNP_TDI

Value 3
 Name: ImagePath
 Type: REG_EXPAND_SZ
 Data: \SystemRoot\System32\drivers\tcpip.sys

Value 4
 Name: Start
 Type: REG_DWORD
 Data: 0x2

Value 5
 Name: Type
 Type: REG_DWORD
 Data: 0x1

Key Name: SYSTEM\CurrentControlSet\Services\Tcpip\Enum
 Class Name: <NO CLASS>
 Last Write Time: 3/9/99 - 3:58 PM

Value 0
 Name: 0
 Type: REG_SZ
 Data: Root\LEGACY_TCPIP\0000

Value 1
 Name: Count
 Type: REG_DWORD
 Data: 0x1

Value 2
 Name: NextInstance
 Type: REG_DWORD
 Data: 0x1

Key Name: SYSTEM\CurrentControlSet\Services\Tcpip\Linkage
Class Name: GenericClass
Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name: Bind
Type: REG_MULTI_SZ
Data: \Device\E100B1
\Device\E100B3
\Device\E100B4
\Device\E100B5
\Device\E100B6
\Device\E100B2

Value 1
Name: Export
Type: REG_MULTI_SZ
Data: \Device\Tcpip\E100B1
\Device\Tcpip\E100B3
\Device\Tcpip\E100B4
\Device\Tcpip\E100B5
\Device\Tcpip\E100B6
\Device\Tcpip\E100B2

Value 2
Name: Route
Type: REG_MULTI_SZ
Data: "E100B" "E100B1"
"E100B" "E100B3"
"E100B" "E100B4"
"E100B" "E100B5"
"E100B" "E100B6"
"E100B" "E100B2"

Key Name: SYSTEM\CurrentControlSet\Services\Tcpip\Linkage\Disabled
Class Name: GenericClass
Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name: Bind
Type: REG_MULTI_SZ
Data:
Value 1
Name: Export
Type: REG_MULTI_SZ
Data:
Value 2
Name: Route

Type: REG_MULTI_SZ
Data:

Key Name: SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
Class Name: GenericClass
Last Write Time: 12/11/97 - 2:23 PM
Value 0
Name: DataBasePath
Type: REG_EXPAND_SZ
Data: %SystemRoot%\System32\drivers\etc

Value 1
Name: Domain
Type: REG_SZ
Data:

Value 2
Name: EnableSecurityFilters
Type: REG_DWORD
Data: 0

Value 3
Name: ForwardBroadcasts
Type: REG_DWORD
Data: 0

Value 4
Name: Hostname
Type: REG_SZ
Data: weiss

Value 5
Name: IPEnableRouter
Type: REG_DWORD
Data: 0

Value 6
Name: MaxUserPort
Type: REG_DWORD
Data: 0xffff

Value 7
Name: NameServer
Type: REG_SZ
Data:

Value 8
Name: SearchList
Type: REG_SZ
Data:

Key Name:
SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\PersistentRoutes
Class Name: GenericClass
Last Write Time: 12/7/97 - 4:01 PM

Key Name:
SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Winsock
Class Name: GenericClass
Last Write Time: 12/7/97 - 4:01 PM
Value 0
Name: HelperDllName
Type: REG_EXPAND_SZ
Data: %SystemRoot%\System32\wshtcpip.dll

Value 1
Name: Mapping
Type: REG_BINARY
Data:
00000000 0b 00 00 00 03 00 00 00 - 02 00 00 00 01 00 00 00
.....
00000010 06 00 00 00 02 00 00 00 - 01 00 00 00 00 00 00 00
.....
00000020 02 00 00 00 00 00 00 00 - 06 00 00 00 00 00 00 00
.....
00000030 00 00 00 00 06 00 00 00 - 00 00 00 00 01 00 00 00
.....
00000040 06 00 00 00 02 00 00 00 - 02 00 00 00 11 00 00 00
.....
00000050 02 00 00 00 02 00 00 00 - 00 00 00 00 02 00 00 00
.....
00000060 00 00 00 00 11 00 00 00 - 00 00 00 00 00 00 00 00
.....
00000070 11 00 00 00 00 00 00 00 - 02 00 00 00 11 00 00 00
.....
00000080 02 00 00 00 03 00 00 00 - 00 00 00 00
.....

Value 2
Name: MaxSockAddrLength
Type: REG_DWORD
Data: 0x10

Value 3
Name: MinSockAddrLength
Type: REG_DWORD
Data: 0x10

Key Name: SYSTEM\CurrentControlSet\Services\Tcpip\Performance
Class Name: GenericClass
Last Write Time: 12/7/97 - 4:01 PM
Value 0
Name: Close

Type: REG_SZ
Data: CloseTcpIpPerformanceData

Value 1
Name: Collect
Type: REG_SZ
Data: CollectTcpIpPerformanceData

Value 2
Name: Library
Type: REG_SZ
Data: Perfctrs.dll

Value 3
Name: Open
Type: REG_SZ
Data: OpenTcpIpPerformanceData

Key Name: SYSTEM\CurrentControlSet\Services\Tcpip\Security
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:01 PM

Value 0
Name: Security
Type: REG_BINARY
Data:
00000000 01 00 14 80 c0 00 00 00 - cc 00 00 00 14 00 00 00
.....
00000010 34 00 00 00 02 00 20 00 - 01 00 00 00 02 80 18 00 4.....
.....
00000020 ff 01 0f 00 01 01 00 00 - 00 00 00 01 00 00 00 00
.....
00000030 20 02 00 00 02 00 8c 00 - 05 00 00 00 00 00 18 00
.....
00000040 8d 01 02 00 01 01 00 00 - 00 00 00 01 00 00 00 00
.....
00000050 6d 00 00 00 00 00 1c 00 - fd 01 02 00 01 02 00 00
m.....
00000060 00 00 00 05 20 00 00 00 - 23 02 00 00 53 00 65 00
...#...S.e.
00000070 00 00 1c 00 ff 01 0f 00 - 01 02 00 00 00 00 00 05
.....
00000080 20 00 00 00 20 02 00 00 - 53 00 65 00 00 00 1c 00 ...
...S.e.....
00000090 ff 01 0f 00 01 02 00 00 - 00 00 00 05 20 00 00 00
.....
000000a0 25 02 00 00 53 00 65 00 - 00 00 18 00 fd 01 02 00
%...S.e.....
000000b0 01 01 00 00 00 00 00 05 - 12 00 00 00 25 02 00 00
.....%...
000000c0 01 01 00 00 00 00 00 05 - 12 00 00 00 01 01 00 00
.....
000000d0 00 00 00 05 12 00 00 00 -

Key Name: SYSTEM\CurrentControlSet\Services\Tcpip\ServiceProvider
 Class Name: GenericClass
 Last Write Time: 12/7/97 - 4:01 PM
 Value 0
 Name: Class
 Type: REG_DWORD
 Data: 0x8
 Value 1
 Name: DnsPriority
 Type: REG_DWORD
 Data: 0x7d0
 Value 2
 Name: HostsPriority
 Type: REG_DWORD
 Data: 0x1f4
 Value 3
 Name: LocalPriority
 Type: REG_DWORD
 Data: 0x1f3
 Value 4
 Name: Name
 Type: REG_SZ
 Data: TCP/IP
 Value 5
 Name: NetbtPriority
 Type: REG_DWORD
 Data: 0x7d1
 Value 6
 Name: ProviderPath
 Type: REG_EXPAND_SZ
 Data: %SystemRoot%\System32\wsock32.dll

 Key Name: SOFTWARE\BEA Systems
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:00 PM

 Key Name: SOFTWARE\BEA Systems\TUXEDO
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:00 PM

 Key Name: SOFTWARE\BEA Systems\TUXEDO\6.4
 Class Name: <NO CLASS>

Last Write Time: 12/8/97 - 1:46 PM
 Value 0
 Name: Company_Name
 Type: REG_SZ
 Data: Siemens
 Value 1
 Name: Install_Date
 Type: REG_SZ
 Data: 12-8-1997
 Value 2
 Name: License-Token
 Type: REG_DWORD
 Data: 0
 Value 3
 Name: Major_Version
 Type: REG_DWORD
 Data: 0x6
 Value 4
 Name: Minor_Version
 Type: REG_DWORD
 Data: 0x4
 Value 5
 Name: Serial_Number
 Type: REG_DWORD
 Data: 0
 Value 6
 Name: User_Name
 Type: REG_SZ
 Data: tpcc
 Value 7
 Name: Volume_Number
 Type: REG_DWORD
 Data: 0x1

 Key Name: SOFTWARE\BEA Systems\TUXEDO\6.4\Developer
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:00 PM

 Key Name: SOFTWARE\BEA Systems\TUXEDO\6.4\Developer\Libraries
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:00 PM

 Key Name: SOFTWARE\BEA Systems\TUXEDO\6.4\Developer\Libraries\All
 Class Name: <NO CLASS>

Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\All\libfml.lib
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\All\libfml32.lib
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\All\libgp.lib
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\Client
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\Client\libbuft.lib
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\Client\libtux.lib
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\Client\libtux2.lib
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\Server
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\Server\libbuft.lib
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\Server\libtux.lib
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\Server\libtux2.lib
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\Workstation
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\Workstation\libbuft.lib
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\Workstation\libnwi.lib
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\Workstation\libnws.lib
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
Systems\TUXEDO\6.4\Developer\Libraries\Workstation\libwsc.lib
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA Systems\TUXEDO\6.4\Environment
Class Name: <NO CLASS>
Last Write Time: 3/3/99 - 11:53 AM
Value 0
Name: NLSPATH
Type: REG_SZ
Data: C:\TUXEDO\locale\C

Value 1
Name: TUXDIR
Type: REG_SZ
Data: C:\TUXEDO

Value 2
Name: TUXIPC_MSG_BYTES
Type: REG_DWORD
Data: 0x10000

Value 3
Name: TUXIPC_MSG_HDRS
Type: REG_DWORD
Data: 0x1fc0

Value 4
 Name: TUXIPC_MSG_QUEUE_BYTES
 Type: REG_DWORD
 Data: 0x10000

Value 5
 Name: TUXIPC_MSG_QUEUES
 Type: REG_DWORD
 Data: 0x400

Value 6
 Name: TUXIPC_MSG_SEG_BYTES
 Type: REG_DWORD
 Data: 0x40

Value 7
 Name: TUXIPC_MSG_SEGS
 Type: REG_DWORD
 Data: 0x7fff

Value 8
 Name: TUXIPC_PROC
 Type: REG_DWORD
 Data: 0x400

Value 9
 Name: TUXIPC_SEM
 Type: REG_DWORD
 Data: 0x800

Value 10
 Name: TUXIPC_SEM_IDS
 Type: REG_DWORD
 Data: 0x800

Value 11
 Name: TUXIPC_SEM_UNDO
 Type: REG_DWORD
 Data: 0x800

Value 12
 Name: TUXIPC_SHM_PROCS
 Type: REG_DWORD
 Data: 0x800

Value 13
 Name: TUXIPC_SHM_SEGS
 Type: REG_DWORD
 Data: 0x32

Value 14
 Name: ULOGDIR
 Type: REG_SZ

Data: C:\TUXEDO

Value 15
 Name: ULOGOUT
 Type: REG_DWORD
 Data: 0x2

Value 16
 Name: ULOGPFX
 Type: REG_SZ
 Data: C:\ULOG

Key Name: SOFTWARE\BEA Systems\TUXEDO\6.4\Environment\Services
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA
 Systems\TUXEDO\6.4\Environment\Services\3050
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:00 PM

Key Name: SOFTWARE\BEA Systems\TUXEDO\6.4\IPCResources
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:00 PM

Value 0
 Name: CurrentResource
 Type: REG_SZ
 Data: tpcc

Key Name: SOFTWARE\BEA Systems\TUXEDO\6.4\IPCResources\tpcc
 Class Name: <NO CLASS>
 Last Write Time: 3/3/99 - 11:53 AM

Value 0
 Name: TUXIPC_MSG_BYTES
 Type: REG_DWORD
 Data: 0x10000

Value 1
 Name: TUXIPC_MSG_HDRS
 Type: REG_DWORD
 Data: 0x1fc0

Value 2
 Name: TUXIPC_MSG_QUEUE_BYTES
 Type: REG_DWORD
 Data: 0x10000

Value 3
 Name: TUXIPC_MSG_QUEUES
 Type: REG_DWORD
 Data: 0x400

Value 4
 Name: TUXIPC_MSG_SEG_BYTES
 Type: REG_DWORD
 Data: 0x40

Value 5
 Name: TUXIPC_MSG_SEGS
 Type: REG_DWORD
 Data: 0x7fff

Value 6
 Name: TUXIPC_PROC
 Type: REG_DWORD
 Data: 0x400

Value 7
 Name: TUXIPC_SEM
 Type: REG_DWORD
 Data: 0x800

Value 8
 Name: TUXIPC_SEM_IDS
 Type: REG_DWORD
 Data: 0x800

Value 9
 Name: TUXIPC_SEM_UNDO
 Type: REG_DWORD
 Data: 0x800

Value 10
 Name: TUXIPC_SHM_PROCS
 Type: REG_DWORD
 Data: 0x800

Value 11
 Name: TUXIPC_SHM_SEGS
 Type: REG_DWORD
 Data: 0x32

Key Name: SOFTWARE\BEA Systems\TUXEDO\6.4\SECURITY
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:00 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:01 PM

Value 0
 Name: DependOnGroup

Type: REG_MULTI_SZ
 Data:

Value 1
 Name: DependOnService
 Type: REG_MULTI_SZ
 Data: RPCSS
 NTLMSSP

Value 2
 Name: DisplayName
 Type: REG_SZ
 Data: World Wide Web Publishing Service

Value 3
 Name: ErrorControl
 Type: REG_DWORD
 Data: 0

Value 4
 Name: ImagePath
 Type: REG_EXPAND_SZ
 Data: C:\WINNT\System32\inetssrv\inetinfo.exe

Value 5
 Name: ObjectName
 Type: REG_SZ
 Data: LocalSystem

Value 6
 Name: Start
 Type: REG_DWORD
 Data: 0x2

Value 7
 Name: Type
 Type: REG_DWORD
 Data: 0x20

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\ASP
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:01 PM

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\ASP\Parameters
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:01 PM

Value 0
 Name: AllowOutOfProcCmpnts
 Type: REG_DWORD
 Data: 0

Value 1
 Name: AllowSessionState
 Type: REG_DWORD
 Data: 0x1

Value 2
 Name: BufferingOn
 Type: REG_DWORD
 Data: 0

Value 3
 Name: CheckForNestedVroots
 Type: REG_DWORD
 Data: 0x1

Value 4
 Name: DefaultScriptLanguage
 Type: REG_SZ
 Data: VBScript

Value 5
 Name: EnableParentPaths
 Type: REG_DWORD
 Data: 0x1

Value 6
 Name: EventLogDirection
 Type: REG_DWORD
 Data: 0

Value 7
 Name: LogErrorRequests
 Type: REG_DWORD
 Data: 0x1

Value 8
 Name: MemFreeFactor
 Type: REG_DWORD
 Data: 0x32

Value 9
 Name: MinUsedBlocks
 Type: REG_DWORD
 Data: 0xa

Value 10
 Name: NumInitialThreads
 Type: REG_DWORD
 Data: 0x2

Value 11
 Name: ProcessorThreadMax
 Type: REG_DWORD

Data: 0xa

Value 12
 Name: RequestQueueMax
 Type: REG_DWORD
 Data: 0x1f4

Value 13
 Name: ScriptEngineCacheMax
 Type: REG_DWORD
 Data: 0x1e

Value 14
 Name: ScriptErrorMessage
 Type: REG_SZ
 Data: An error occurred on the server when processing the URL. Please contact the system administrator.

Value 15
 Name: ScriptErrorsSentToBrowser
 Type: REG_DWORD
 Data: 0x1

Value 16
 Name: ScriptFileCacheSize
 Type: REG_DWORD
 Data: 0xffffffff

Value 17
 Name: ScriptFileCacheTTL
 Type: REG_DWORD
 Data: 0x12c

Value 18
 Name: ScriptTimeout
 Type: REG_DWORD
 Data: 0x5a

Value 19
 Name: SessionTimeout
 Type: REG_DWORD
 Data: 0x14

Value 20
 Name: StartConnectionPool
 Type: REG_DWORD
 Data: 0

Value 21
 Name: ThreadCreationThreshold
 Type: REG_DWORD
 Data: 0x5

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Enum
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Value 0
Name: 0
Type: REG_SZ
Data: Root\LEGACY_W3SVC\0000

Value 1
Name: Count
Type: REG_DWORD
Data: 0x1

Value 2
Name: NextInstance
Type: REG_DWORD
Data: 0x1

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Parameters
Class Name: <NO CLASS>
Last Write Time: 12/11/97 - 2:23 PM

Value 0
Name: AcceptExOutstanding
Type: REG_DWORD
Data: 0x1400

Value 1
Name: AccessDeniedMessage
Type: REG_SZ
Data: Error: Access is Denied.

Value 2
Name: AdminEmail
Type: REG_SZ
Data: Admin@corp.com

Value 3
Name: AdminName
Type: REG_SZ
Data: Administrator

Value 4
Name: AnonymousUserName
Type: REG_SZ
Data: IUSR_GELB

Value 5
Name: Authorization
Type: REG_DWORD
Data: 0x5

Value 6
Name: CacheExtensions
Type: REG_DWORD
Data: 0x1

Value 7
Name: CheckForWAISDB
Type: REG_DWORD
Data: 0

Value 8
Name: ConnectionTimeOut
Type: REG_DWORD
Data: 0x258

Value 9
Name: DebugFlags
Type: REG_DWORD
Data: 0x8

Value 10
Name: Default Load File
Type: REG_SZ
Data: Default.htm

Value 11
Name: Dir Browse Control
Type: REG_DWORD
Data: 0x4000001e

Value 12
Name: Filter DLLs
Type: REG_SZ
Data: C:\WINNT\System32\inet_srv\sspifilt.dll

Value 13
Name: GlobalExpire
Type: REG_DWORD
Data: 0xffffffff

Value 14
Name: InstallPath
Type: REG_SZ
Data: C:\WINNT\System32\inet_srv

Value 15
Name: LogFileDirectory
Type: REG_EXPAND_SZ
Data: %SystemRoot%\System32\LogFiles

Value 16
Name: LogFileFormat
Type: REG_DWORD

Data: 0

Value 17
 Name: LogFilePeriod
 Type: REG_DWORD
 Data: 0x1

Value 18
 Name: LogFileTruncateSize
 Type: REG_DWORD
 Data: 0x1388000

Value 19
 Name: LogSqlDataSource
 Type: REG_SZ
 Data: HTTPLOG

Value 20
 Name: LogSqlPassword
 Type: REG_SZ
 Data: sqllog

Value 21
 Name: LogSqlTableName
 Type: REG_SZ
 Data: Internetlog

Value 22
 Name: LogSqlUserName
 Type: REG_SZ
 Data: InternetAdmin

Value 23
 Name: LogType
 Type: REG_DWORD
 Data: 0

Value 24
 Name: MajorVersion
 Type: REG_DWORD
 Data: 0x2

Value 25
 Name: MaxConnections
 Type: REG_DWORD
 Data: 0x186a0

Value 26
 Name: MinorVersion
 Type: REG_DWORD
 Data: 0

Value 27

Name: NTAAuthenticationProviders
 Type: REG_SZ
 Data: NTLM

Value 28
 Name: ScriptTimeout
 Type: REG_DWORD
 Data: 0x258

Value 29
 Name: SecurePort
 Type: REG_DWORD
 Data: 0x1bb

Value 30
 Name: ServerSideIncludesEnabled
 Type: REG_DWORD
 Data: 0x1

Value 31
 Name: ServerSideIncludesExtension
 Type: REG_SZ
 Data: .stm

Key Name:
 SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:01 PM

Key Name:
 SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedData
 Factory
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:01 PM

Key Name:
 SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSServer.Da
 taFactory
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:01 PM

Key Name:
 SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map
 Class Name: <NO CLASS>
 Last Write Time: 12/7/97 - 4:01 PM

Value 0
 Name: .ASA
 Type: REG_SZ
 Data: C:\WINNT\System32\inet_srv\ASP\ASP.dll

Value 1
 Name: .ASP

Type: REG_SZ
Data: C:\WINNT\System32\inetsrv\ASP\ASP.dll

Value 2
Name: .ida
Type: REG_SZ
Data: C:\WINNT\System32\idq.dll

Value 3
Name: .idc
Type: REG_SZ
Data: C:\WINNT\System32\inetsrv\httpodbc.dll

Value 4
Name: .idq
Type: REG_SZ
Data: C:\WINNT\System32\idq.dll

Value 5
Name: .STM
Type: REG_SZ
Data: C:\WINNT\System32\inetsrv\ssinc.dll

Key Name:
SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:01 PM

Value 0
Name: /,
Type: REG_SZ
Data: C:\InetPub\wwwroot,,5

Value 1
Name: /AdvWorks,
Type: REG_SZ
Data: C:\inetpub\ASPSamp\AdvWorks,,5

Value 2
Name: /ASPSamp,
Type: REG_SZ
Data: C:\inetpub\ASPSamp,,5

Value 3
Name: /cgi-bin,
Type: REG_SZ
Data: C:\TUXEDO\udataobj\webgui\cgi-bin,,5

Value 4
Name: /doc,
Type: REG_SZ
Data: C:\TUXEDO\DOC,,5

Value 5
Name: /IASDocs,
Type: REG_SZ
Data: C:\WINNT\System32\inetsrv\Docs,,5

Value 6
Name: /iisadmin,
Type: REG_SZ
Data: C:\WINNT\System32\inetsrv\iisadmin,,1

Value 7
Name: /java,
Type: REG_SZ
Data: C:\TUXEDO\udataobj\webgui\java,,5

Value 8
Name: /MSADC,
Type: REG_SZ
Data: C:\Program Files\Common Files\System\MSADC,,5

Value 9
Name: /Scripts,
Type: REG_SZ
Data: C:\InetPub\scripts,,4

Value 10
Name: /srchadm,
Type: REG_SZ
Data: C:\InetPub\wwwroot\srchadm,,1

Value 11
Name: /tuxedo,
Type: REG_SZ
Data: C:\TUXEDO\udataobj\webgui,,5

Key Name: SYSTEM\CurrentControlSet\Services\W3SVC\Performance
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:01 PM

Value 0
Name: Close
Type: REG_SZ
Data: CloseW3PerformanceData

Value 1
Name: Collect
Type: REG_SZ
Data: CollectW3PerformanceData

Value 2
Name: First Counter
Type: REG_DWORD
Data: 0x758


```

Value 3
Name:      First Help
Type:      REG_DWORD
Data:      0x759

Value 4
Name:      Last Counter
Type:      REG_DWORD
Data:      0x790

Value 5
Name:      Last Help
Type:      REG_DWORD
Data:      0x791

Value 6
Name:      Library
Type:      REG_SZ
Data:      w3ctrs.DLL

Value 7
Name:      Open
Type:      REG_SZ
Data:      OpenW3PerformanceData

Key Name:  SYSTEM\CurrentControlSet\Services\W3SVC\Security
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:01 PM

Value 0
Name:      Security
Type:      REG_BINARY
Data:
00000000  01 00 14 80 c0 00 00 00 - cc 00 00 00 14 00 00 00
.....
00000010  34 00 00 00 02 00 20 00 - 01 00 00 00 02 80 18 00  4.....
.....
00000020  ff 01 0f 00 01 01 00 00 - 00 00 00 01 00 00 00 00
.....
00000030  20 02 00 00 02 00 8c 00 - 05 00 00 00 00 00 18 00
.....
00000040  8d 01 02 00 01 01 00 00 - 00 00 00 01 00 00 00 00
.....
00000050  6d 00 63 00 00 00 1c 00 - fd 01 02 00 01 02 00 00
m.c.....
00000060  00 00 00 05 20 00 00 00 - 23 02 00 00 69 00 61 00  ....
...#...i.a.
00000070  00 00 1c 00 ff 01 0f 00 - 01 02 00 00 00 00 05
.....
00000080  20 00 00 00 20 02 00 00 - 69 00 61 00 00 00 1c 00  ...
...i.a....

```

```

00000090  ff 01 0f 00 01 02 00 00 - 00 00 00 05 20 00 00 00
.....
000000a0  25 02 00 00 69 00 61 00 - 00 00 18 00 fd 01 02 00
%...i.a.....
000000b0  01 01 00 00 00 00 00 05 - 12 00 00 00 25 02 00 00
.....%...
000000c0  01 01 00 00 00 00 00 05 - 12 00 00 00 01 01 00 00
.....
000000d0  00 00 00 05 12 00 00 00 - .....

Key Name:      SYSTEM\CurrentControlSet\Services\W3SVC\W3SAMP
Class Name:    <NO CLASS>
Last Write Time: 12/7/97 - 4:01 PM

Key Name:      SYSTEM\CurrentControlSet\Services\E100B
Class Name:    <NO CLASS>
Last Write Time: 3/9/99 - 3:59 PM

Value 0
Name:          DisplayName
Type:          REG_SZ
Data:          Intel(R) PRO NDIS Driver

Value 1
Name:          ErrorControl
Type:          REG_DWORD
Data:          0x1

Value 2
Name:          Group
Type:          REG_SZ
Data:          NDIS

Value 3
Name:          ImagePath
Type:          REG_EXPAND_SZ
Data:          \SystemRoot\System32\drivers\e100bnt.sys

Value 4
Name:          RequestedSystemResources
Type:          REG_RESOURCE_REQUIREMENTS_LIST
Data:
Interface Type:      Internal
Bus Number:          0
Slot Number:         0
List 0
Descriptor 0
Resource:            Interrupt
Option:              0x00000000
Disposition:         Shared
Type:                Level Sensitive

```

Minimum Vector: 0x4
 Maximum Vector: 0x4

Descriptor 1
 Resource: Memory
 Option: 0x00000001
 Disposition: Device Exclusive
 Type: Write Only
 Length: 0x1000
 Alignment: 0x1000
 Minimum Address: 0xfecfc000
 Maximum Address: 0xfecfcfff

Descriptor 2
 Resource: Memory
 Option: 0x00000009
 Disposition: Device Exclusive
 Type: Write Only
 Length: 0x1000
 Alignment: 0x1000
 Minimum Address: 0xfecfc000
 Maximum Address: 0xfecfcfff

Descriptor 3
 Resource: Memory
 Option: 0x00000008
 Disposition: Device Exclusive
 Type: Write Only
 Length: 0x1000
 Alignment: 0x1000
 Minimum Address: 0xfe900000
 Maximum Address: 0xfe9fffff

Descriptor 4
 Resource: Port
 Option: 0x00000001
 Disposition: Device Exclusive
 Type: Port
 Length: 0x20
 Alignment: 0x20
 Minimum Address: 0x0000e000
 Maximum Address: 0x0000e01f

Descriptor 5
 Resource: Port
 Option: 0x00000008
 Disposition: Device Exclusive
 Type: Port
 Length: 0x20
 Alignment: 0x20
 Minimum Address: 0x0000e000
 Maximum Address: 0x0000e01f

Descriptor 6
 Resource: Memory
 Option: 0x00000001
 Disposition: Device Exclusive
 Type: Read / Write
 Length: 0x100000
 Alignment: 0x100000
 Minimum Address: 0xfe900000
 Maximum Address: 0xfe9fffff

Descriptor 7
 Resource: Memory
 Option: 0x00000008
 Disposition: Device Exclusive
 Type: Read / Write
 Length: 0x100000
 Alignment: 0x100000
 Minimum Address: 0xfe900000
 Maximum Address: 0xfe9fffff

Value 5
 Name: Start
 Type: REG_DWORD
 Data: 0x2

Value 6
 Name: Type
 Type: REG_DWORD
 Data: 0x1

Key Name: SYSTEM\CurrentControlSet\Services\E100B\Enum
Class Name: <NO CLASS>
Last Write Time: 3/9/99 - 3:58 PM

Value 0
 Name: 0
 Type: REG_SZ
 Data: Root\LEGACY_E100B\0000

Value 1
 Name: Count
 Type: REG_DWORD
 Data: 0x1

Value 2
 Name: NextInstance
 Type: REG_DWORD
 Data: 0x1

Key Name: SYSTEM\CurrentControlSet\Services\E100B\Linkage

Class Name: GenericClass
Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name: Bind
Type: REG_MULTI_SZ
Data: \\Device\E100B1
\\Device\E100B3
\\Device\E100B4
\\Device\E100B5
\\Device\E100B6
\\Device\E100B2

Value 1
Name: Export
Type: REG_MULTI_SZ
Data: \\Device\E100B1
\\Device\E100B3
\\Device\E100B4
\\Device\E100B5
\\Device\E100B6
\\Device\E100B2

Value 2
Name: Route
Type: REG_MULTI_SZ
Data: "E100B1"
"E100B3"
"E100B4"
"E100B5"
"E100B6"
"E100B2"

Key Name: SYSTEM\CurrentControlSet\Services\E100B\Linkage\Disabled
Class Name: GenericClass
Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name: Bind
Type: REG_MULTI_SZ
Data:

Value 1
Name: Export
Type: REG_MULTI_SZ
Data:

Value 2
Name: Route
Type: REG_MULTI_SZ

Data:
Key Name: SYSTEM\CurrentControlSet\Services\E100B\Parameters
Class Name: GenericClass
Last Write Time: 12/7/97 - 4:01 PM

Key Name: SYSTEM\CurrentControlSet\Services\E100B\Security
Class Name: <NO CLASS>
Last Write Time: 12/7/97 - 4:01 PM
Value 0
Name: Security
Type: REG_BINARY
Data:

00000000 01 00 14 80 c0 00 00 00 - cc 00 00 00 14 00 00 00
.....
00000010 34 00 00 00 02 00 20 00 - 01 00 00 00 02 80 18 00 4.....
.....
00000020 ff 01 0f 00 01 01 00 00 - 00 00 00 01 00 00 00 00
.....
00000030 20 02 00 00 02 00 8c 00 - 05 00 00 00 00 00 18 00
.....
00000040 8d 01 02 00 01 01 00 00 - 00 00 00 01 00 00 00 00
.....
00000050 6d 00 63 00 00 00 1c 00 - fd 01 02 00 01 02 00 00
m.c.....
00000060 00 00 00 05 20 00 00 00 - 23 02 00 00 69 00 61 00
...#.i.a.
00000070 00 00 1c 00 ff 01 0f 00 - 01 02 00 00 00 00 00 05
.....
00000080 20 00 00 00 20 02 00 00 - 69 00 61 00 00 00 1c 00 ...
...i.a.....
00000090 ff 01 0f 00 01 02 00 00 - 00 00 00 05 20 00 00 00
.....
000000a0 25 02 00 00 69 00 61 00 - 00 00 18 00 fd 01 02 00
%.i.a.....
000000b0 01 01 00 00 00 00 00 05 - 12 00 00 00 25 02 00 00
.....%...
000000c0 01 01 00 00 00 00 00 05 - 12 00 00 00 01 01 00 00
.....
000000d0 00 00 00 05 12 00 00 00 -

Key Name: SYSTEM\CurrentControlSet\Services\E100B1
Class Name: GenericClass
Last Write Time: 12/7/97 - 4:01 PM
Value 0
Name: ErrorControl
Type: REG_DWORD
Data: 0x1

Value 1
Name: Start

Type: REG_DWORD
Data: 0x3

Value 2
Name: Type
Type: REG_DWORD
Data: 0x4

Key Name: SYSTEM\CurrentControlSet\Services\E100B1\Linkage
Class Name: GenericClass
Last Write Time: 3/9/99 - 3:59 PM

Value 0
Name: Bind
Type: REG_MULTI_SZ
Data: \Device\E100B1

Value 1
Name: Export
Type: REG_MULTI_SZ
Data: \Device\E100B1

Value 2
Name: Route
Type: REG_MULTI_SZ
Data: "E100B1"

Key Name: SYSTEM\CurrentControlSet\Services\E100B1\Linkage\Disabled
Class Name: GenericClass
Last Write Time: 12/7/97 - 4:01 PM

Key Name: SYSTEM\CurrentControlSet\Services\E100B1\Parameters
Class Name: GenericClass
Last Write Time: 3/8/99 - 10:10 AM

Value 0
Name: Adaptive_IFS
Type: REG_DWORD
Data: 0x1

Value 1
Name: BoardHasBridge
Type: REG_DWORD

Data: 0

Value 2
Name: BusNumber
Type: REG_DWORD
Data: 0

Value 3
Name: BusType
Type: REG_DWORD
Data: 0x5

Value 4
Name: BusTypeLocal
Type: REG_DWORD
Data: 0x5

Value 5
Name: Coalesce
Type: REG_DWORD
Data: 0

Value 6
Name: CPUSaver
Type: REG_DWORD
Data: 0x600

Value 7
Name: ForceDpx
Type: REG_DWORD
Data: 0x2

Value 8
Name: Location
Type: REG_SZ
Data: 6000

Value 9
Name: MediaType
Type: REG_DWORD
Data: 0x1

Value 10
Name: MWIEnable
Type: REG_DWORD
Data: 0

Value 11
Name: NetworkAddress
Type: REG_SZ
Data: 0

Value 12

Name: NumCoalesce
 Type: REG_DWORD
 Data: 0x20

Value 13
 Name: NumRfd
 Type: REG_DWORD
 Data: 0x80

Value 14
 Name: NumTbdPerTcb
 Type: REG_DWORD
 Data: 0xc

Value 15
 Name: NumTcb
 Type: REG_DWORD
 Data: 0x40

Value 16
 Name: Pcnic
 Type: REG_DWORD
 Data: 0x1

Value 17
 Name: RxDmaCount
 Type: REG_DWORD
 Data: 0

Value 18
 Name: RxFifo
 Type: REG_DWORD
 Data: 0x8

Value 19
 Name: SlotNumber
 Type: REG_DWORD
 Data: 0xc

Value 20
 Name: Speed
 Type: REG_DWORD
 Data: 0x64

Value 21
 Name: Threshold
 Type: REG_DWORD
 Data: 0x10

Value 22
 Name: TxDmaCount
 Type: REG_DWORD
 Data: 0

Value 23
 Name: TxFifo
 Type: REG_DWORD
 Data: 0x8

Value 24
 Name: Txmitwait
 Type: REG_DWORD
 Data: 0x1

Value 25
 Name: UcodeSW
 Type: REG_DWORD
 Data: 0x1

Value 26
 Name: UnderrunRetry
 Type: REG_DWORD
 Data: 0x1

Value 27
 Name: UseIo
 Type: REG_DWORD
 Data: 0x2

Value 28
 Name: UseManualPCIAssign
 Type: REG_DWORD
 Data: 0

Value 29
 Name: VLanMode
 Type: REG_DWORD
 Data: 0

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B1\Parameters\Tcpip
 Class Name: GenericClass
 Last Write Time: 3/9/99 - 3:59 PM

Value 0
 Name: DefaultGateway
 Type: REG_MULTI_SZ
 Data:

Value 1
 Name: EnableDHCP
 Type: REG_DWORD
 Data: 0

Value 2

Name: IPAddress
Type: REG_MULTI_SZ
Data: 129.103.181.123

Value 3
Name: IPInterfaceContext
Type: REG_DWORD
Data: 0x6

Value 4
Name: IPInterfaceContextMax
Type: REG_DWORD
Data: 0x6

Value 5
Name: LLInterface
Type: REG_SZ
Data:

Value 6
Name: PPTPFiltering
Type: REG_DWORD
Data: 0

Value 7
Name: RawIPAllowedProtocols
Type: REG_MULTI_SZ
Data: 0

Value 8
Name: SubnetMask
Type: REG_MULTI_SZ
Data: 255.255.255.0

Value 9
Name: TCPAllowedPorts
Type: REG_MULTI_SZ
Data: 0

Value 10
Name: UDPAllowedPorts
Type: REG_MULTI_SZ
Data: 0

Value 11
Name: UseZeroBroadcast
Type: REG_DWORD
Data: 0

Key Name: SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:24 PM

Key Name: SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:24 PM

Key Name: SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\Adaptive_IFS
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 1

Value 2
Name: Max
Type: REG_SZ
Data: 255

Value 3
Name: Min
Type: REG_SZ
Data: 0

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc
Type: REG_SZ
Data: Adaptive Inter-Frame Spacing

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7
Name: Step
Type: REG_SZ

Data: 1

Value 8
 Name: Type
 Type: REG_SZ
 Data: int

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\Coalesce
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: Default
 Type: REG_SZ
 Data: 0

Value 1
 Name: MiniHelp
 Type: REG_SZ
 Data:

Value 2
 Name: ParamDesc
 Type: REG_SZ
 Data: PCI Bus Efficiency

Value 3
 Name: Type
 Type: REG_SZ
 Data: enum

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\Coalesce\Enum
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: 0
 Type: REG_SZ
 Data: Disabled

Value 1
 Name: 1
 Type: REG_SZ
 Data: Enabled

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\CPUSaver
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0

Name: Default
 Type: REG_SZ
 Data: 1536

Value 1
 Name: LeftLabel
 Type: REG_SZ
 Data: Adapter Bandwidth

Value 2
 Name: MiniHelp
 Type: REG_SZ
 Data: Sets optimal point of CPU/Adapter performance for this system. See help.

Value 3
 Name: ParamDesc
 Type: REG_SZ
 Data: Adaptive Performance Tuning

Value 4
 Name: RightLabel
 Type: REG_SZ
 Data: CPU Utilization

Value 5
 Name: Type
 Type: REG_SZ
 Data: slider

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\CPUSaver\Values
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: 0
 Type: REG_SZ
 Data: 0

Value 1
 Name: 1
 Type: REG_SZ
 Data: 256

Value 2
 Name: 10
 Type: REG_SZ
 Data: 2560

Value 3
 Name: 11
 Type: REG_SZ

Data: 2816

Value 4
 Name: 12
 Type: REG_SZ
 Data: 3072

Value 5
 Name: 13
 Type: REG_SZ
 Data: 3328

Value 6
 Name: 14
 Type: REG_SZ
 Data: 3584

Value 7
 Name: 15
 Type: REG_SZ
 Data: 3840

Value 8
 Name: 16
 Type: REG_SZ
 Data: 4096

Value 9
 Name: 2
 Type: REG_SZ
 Data: 512

Value 10
 Name: 3
 Type: REG_SZ
 Data: 768

Value 11
 Name: 4
 Type: REG_SZ
 Data: 1024

Value 12
 Name: 5
 Type: REG_SZ
 Data: 1280

Value 13
 Name: 6
 Type: REG_SZ
 Data: 1536

Value 14

Name: 7
 Type: REG_SZ
 Data: 1792

Value 15
 Name: 8
 Type: REG_SZ
 Data: 2048

Value 16
 Name: 9
 Type: REG_SZ
 Data: 2304

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\ForceDpx
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: Default
 Type: REG_SZ
 Data: 0

Value 1
 Name: MiniHelp
 Type: REG_SZ
 Data:

Value 2
 Name: ParamDesc
 Type: REG_SZ
 Data: Duplex

Value 3
 Name: Type
 Type: REG_SZ
 Data: enum

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\ForceDpx\Enum
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: 0
 Type: REG_SZ
 Data: Auto Detect

Value 1
 Name: 1
 Type: REG_SZ
 Data: Half-Duplex

Value 2
Name: 2
Type: REG_SZ
Data: Full-Duplex

Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\NetworkAddress
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Default
Type: REG_SZ
Data:

Value 1
Name: MiniHelp
Type: REG_SZ
Data:

Value 2
Name: ParamDesc
Type: REG_SZ
Data: Locally Administered Address

Value 3
Name: Type
Type: REG_SZ
Data: edit

Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\NumCoalesce
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 8

Value 2
Name: Max
Type: REG_SZ
Data: 32

Value 3
Name: Min

Type: REG_SZ
Data: 1

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc
Type: REG_SZ
Data: Coalesce Buffers

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7
Name: Step
Type: REG_SZ
Data: 1

Value 8
Name: Type
Type: REG_SZ
Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\NumRfd
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 48

Value 2
Name: Max
Type: REG_SZ
Data: 1024

Value 3
Name: Min
Type: REG_SZ
Data: 1

```

Value 4
  Name:      MiniHelp
  Type:      REG_SZ
  Data:

Value 5
  Name:      ParamDesc
  Type:      REG_SZ
  Data:      Receive Buffers

Value 6
  Name:      Scale
  Type:      REG_SZ
  Data:      1

Value 7
  Name:      Step
  Type:      REG_SZ
  Data:      1

Value 8
  Name:      Type
  Type:      REG_SZ
  Data:      int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\NumTcb
Class Name:      <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM
Value 0
  Name:      Base
  Type:      REG_SZ
  Data:      10

Value 1
  Name:      Default
  Type:      REG_SZ
  Data:      32

Value 2
  Name:      Max
  Type:      REG_SZ
  Data:      80

Value 3
  Name:      Min
  Type:      REG_SZ
  Data:      1

Value 4
  Name:      MiniHelp
  Type:      REG_SZ

```

```

Data:

Value 5
  Name:      ParamDesc
  Type:      REG_SZ
  Data:      Transmit Control Blocks

Value 6
  Name:      Scale
  Type:      REG_SZ
  Data:      1

Value 7
  Name:      Step
  Type:      REG_SZ
  Data:      1

Value 8
  Name:      Type
  Type:      REG_SZ
  Data:      int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\Speed
Class Name:      <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM
Value 0
  Name:      Default
  Type:      REG_SZ
  Data:      0

Value 1
  Name:      MiniHelp
  Type:      REG_SZ
  Data:

Value 2
  Name:      ParamDesc
  Type:      REG_SZ
  Data:      Speed

Value 3
  Name:      Type
  Type:      REG_SZ
  Data:      enum

Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\Speed\Enum
Class Name:      <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM
Value 0

```

Name: 0
Type: REG_SZ
Data: Auto Detect

Value 1
Name: 10
Type: REG_SZ
Data: 10 Mbps

Value 2
Name: 100
Type: REG_SZ
Data: 100 Mbps

Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\Threshold
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 16

Value 2
Name: Max
Type: REG_SZ
Data: 200

Value 3
Name: Min
Type: REG_SZ
Data: 0

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc
Type: REG_SZ
Data: Adaptive Transmit Threshold

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7
Name: Step
Type: REG_SZ
Data: 1

Value 8
Name: Type
Type: REG_SZ
Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\UcodeSW
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Default
Type: REG_SZ
Data: 1

Value 1
Name: MiniHelp
Type: REG_SZ
Data:

Value 2
Name: ParamDesc
Type: REG_SZ
Data: Adaptive Technology

Value 3
Name: Type
Type: REG_SZ
Data: enum

Key Name:
SYSTEM\CurrentControlSet\Services\E100B1\ProsetNdi\Params\UcodeSW\Enum
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: 0
Type: REG_SZ
Data: Off

Value 1
Name: 1
Type: REG_SZ
Data: On

Key Name: SYSTEM\CurrentControlSet\Services\E100B2

Class Name: GenericClass
Last Write Time: 12/7/97 - 5:49 PM
Value 0
Name: ErrorControl
Type: REG_DWORD
Data: 0x1

Value 1
Name: Start
Type: REG_DWORD
Data: 0x3

Value 2
Name: Type
Type: REG_DWORD
Data: 0x4

Key Name: SYSTEM\CurrentControlSet\Services\E100B2\Linkage
Class Name: GenericClass
Last Write Time: 3/9/99 - 3:59 PM
Value 0
Name: Bind
Type: REG_MULTI_SZ
Data: \Device\E100B2

Value 1
Name: Export
Type: REG_MULTI_SZ
Data: \Device\E100B2

Value 2
Name: Route
Type: REG_MULTI_SZ
Data: "E100B2"

Key Name: SYSTEM\CurrentControlSet\Services\E100B2\Linkage\Disabled
Class Name: GenericClass
Last Write Time: 12/7/97 - 5:49 PM

Key Name: SYSTEM\CurrentControlSet\Services\E100B2\Parameters
Class Name: GenericClass

Last Write Time: 3/8/99 - 10:10 AM

Value 0
Name: Adaptive_IFS
Type: REG_DWORD
Data: 0x1

Value 1
Name: BoardHasBridge
Type: REG_DWORD
Data: 0

Value 2
Name: BusNumber
Type: REG_DWORD
Data: 0

Value 3
Name: BusType
Type: REG_DWORD
Data: 0x5

Value 4
Name: BusTypeLocal
Type: REG_DWORD
Data: 0x5

Value 5
Name: Coalesce
Type: REG_DWORD
Data: 0

Value 6
Name: CPUSaver
Type: REG_DWORD
Data: 0x600

Value 7
Name: ForceDpx
Type: REG_DWORD
Data: 0x1

Value 8
Name: Location
Type: REG_SZ
Data: 6800

Value 9
Name: MediaType
Type: REG_DWORD
Data: 0x1

Value 10
Name: MWIEnable

Type: REG_DWORD
Data: 0

Value 11
Name: NetworkAddress
Type: REG_SZ
Data: 0

Value 12
Name: NumCoalesce
Type: REG_DWORD
Data: 0x20

Value 13
Name: NumRfd
Type: REG_DWORD
Data: 0x80

Value 14
Name: NumTbdPerTcb
Type: REG_DWORD
Data: 0xc

Value 15
Name: NumTcb
Type: REG_DWORD
Data: 0x40

Value 16
Name: PcNic
Type: REG_DWORD
Data: 0x1

Value 17
Name: RxDmaCount
Type: REG_DWORD
Data: 0

Value 18
Name: RxFifo
Type: REG_DWORD
Data: 0x8

Value 19
Name: SlotNumber
Type: REG_DWORD
Data: 0xd

Value 20
Name: Speed
Type: REG_DWORD
Data: 0xa

Value 21
Name: Threshold
Type: REG_DWORD
Data: 0x10

Value 22
Name: TxDmaCount
Type: REG_DWORD
Data: 0

Value 23
Name: TxFifo
Type: REG_DWORD
Data: 0x8

Value 24
Name: Txmitwait
Type: REG_DWORD
Data: 0x1

Value 25
Name: UcodesW
Type: REG_DWORD
Data: 0x1

Value 26
Name: UnderrunRetry
Type: REG_DWORD
Data: 0x1

Value 27
Name: UseIo
Type: REG_DWORD
Data: 0x2

Value 28
Name: UseManualPciAssign
Type: REG_DWORD
Data: 0

Value 29
Name: VLanMode
Type: REG_DWORD
Data: 0

Key Name:
SYSTEM\CurrentControlSet\Services\E100B2\Parameters\Tcpip
Class Name: GenericClass
Last Write Time: 3/9/99 - 3:59 PM

Value 0
Name: DefaultGateway
Type: REG_MULTI_SZ

Data:

Value 1
 Name: EnableDHCP
 Type: REG_DWORD
 Data: 0

Value 2
 Name: IPAddress
 Type: REG_MULTI_SZ
 Data: 129.103.120.2

Value 3
 Name: IPInterfaceContext
 Type: REG_DWORD
 Data: 0x1

Value 4
 Name: IPInterfaceContextMax
 Type: REG_DWORD
 Data: 0x1

Value 5
 Name: LLInterface
 Type: REG_SZ
 Data:

Value 6
 Name: PPTPFiltering
 Type: REG_DWORD
 Data: 0

Value 7
 Name: RawIPAllowedProtocols
 Type: REG_MULTI_SZ
 Data: 0

Value 8
 Name: SubnetMask
 Type: REG_MULTI_SZ
 Data: 255.255.255.0

Value 9
 Name: TCPAllowedPorts
 Type: REG_MULTI_SZ
 Data: 0

Value 10

Name: UDPAIlowedPorts
 Type: REG_MULTI_SZ
 Data: 0

Value 11
 Name: UseZeroBroadcast
 Type: REG_DWORD
 Data: 0

Key Name: SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:24 PM

Key Name: SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:24 PM

Key Name: SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\Adaptive_IFS
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: Base
 Type: REG_SZ
 Data: 10

Value 1
 Name: Default
 Type: REG_SZ
 Data: 1

Value 2
 Name: Max
 Type: REG_SZ
 Data: 255

Value 3
 Name: Min
 Type: REG_SZ
 Data: 0

Value 4
 Name: MiniHelp
 Type: REG_SZ
 Data:

Value 5
 Name: ParamDesc
 Type: REG_SZ
 Data: Adaptive Inter-Frame Spacing

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7
Name: Step
Type: REG_SZ
Data: 1

Value 8
Name: Type
Type: REG_SZ
Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\Coalesce
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Default
Type: REG_SZ
Data: 0

Value 1
Name: MiniHelp
Type: REG_SZ
Data:

Value 2
Name: ParamDesc
Type: REG_SZ
Data: PCI Bus Efficiency

Value 3
Name: Type
Type: REG_SZ
Data: enum

Key Name:
SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\Coalesce\Enum
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: 0
Type: REG_SZ
Data: Disabled

Value 1
Name: 1

Type: REG_SZ
Data: Enabled

Key Name:
SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\CPUSaver
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Default
Type: REG_SZ
Data: 1536

Value 1
Name: LeftLabel
Type: REG_SZ
Data: Adapter Bandwidth

Value 2
Name: MiniHelp
Type: REG_SZ
Data: Sets optimal point of CPU/Adapter performance for this system. See help.

Value 3
Name: ParamDesc
Type: REG_SZ
Data: Adaptive Performance Tuning

Value 4
Name: RightLabel
Type: REG_SZ
Data: CPU Utilization

Value 5
Name: Type
Type: REG_SZ
Data: slider

Key Name:
SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\CPUSaver\Values
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: 0
Type: REG_SZ
Data: 0

Value 1
Name: 1
Type: REG_SZ
Data: 256

Value 2
 Name: 10
 Type: REG_SZ
 Data: 2560

Value 3
 Name: 11
 Type: REG_SZ
 Data: 2816

Value 4
 Name: 12
 Type: REG_SZ
 Data: 3072

Value 5
 Name: 13
 Type: REG_SZ
 Data: 3328

Value 6
 Name: 14
 Type: REG_SZ
 Data: 3584

Value 7
 Name: 15
 Type: REG_SZ
 Data: 3840

Value 8
 Name: 16
 Type: REG_SZ
 Data: 4096

Value 9
 Name: 2
 Type: REG_SZ
 Data: 512

Value 10
 Name: 3
 Type: REG_SZ
 Data: 768

Value 11
 Name: 4
 Type: REG_SZ
 Data: 1024

Value 12
 Name: 5

Type: REG_SZ
 Data: 1280

Value 13
 Name: 6
 Type: REG_SZ
 Data: 1536

Value 14
 Name: 7
 Type: REG_SZ
 Data: 1792

Value 15
 Name: 8
 Type: REG_SZ
 Data: 2048

Value 16
 Name: 9
 Type: REG_SZ
 Data: 2304

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\ForceDpx
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: Default
 Type: REG_SZ
 Data: 0

Value 1
 Name: MiniHelp
 Type: REG_SZ
 Data:

Value 2
 Name: ParamDesc
 Type: REG_SZ
 Data: Duplex

Value 3
 Name: Type
 Type: REG_SZ
 Data: enum

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\ForceDpx\Enum
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: 0
Type: REG_SZ
Data: Auto Detect

Value 1
Name: 1
Type: REG_SZ
Data: Half-Duplex

Value 2
Name: 2
Type: REG_SZ
Data: Full-Duplex

Key Name:
SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\NetworkAddress
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Default
Type: REG_SZ
Data:

Value 1
Name: MiniHelp
Type: REG_SZ
Data:

Value 2
Name: ParamDesc
Type: REG_SZ
Data: Locally Administered Address

Value 3
Name: Type
Type: REG_SZ
Data: edit

Key Name:
SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\NumCoalesce
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ

Data: 8

Value 2
Name: Max
Type: REG_SZ
Data: 32

Value 3
Name: Min
Type: REG_SZ
Data: 1

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc
Type: REG_SZ
Data: Coalesce Buffers

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7
Name: Step
Type: REG_SZ
Data: 1

Value 8
Name: Type
Type: REG_SZ
Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\NumRfd
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 48

Value 2

Name: Max
Type: REG_SZ
Data: 1024

Value 3
Name: Min
Type: REG_SZ
Data: 1

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc
Type: REG_SZ
Data: Receive Buffers

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7
Name: Step
Type: REG_SZ
Data: 1

Value 8
Name: Type
Type: REG_SZ
Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\NumTcb
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 32

Value 2
Name: Max
Type: REG_SZ
Data: 80

Value 3
Name: Min
Type: REG_SZ
Data: 1

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc
Type: REG_SZ
Data: Transmit Control Blocks

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7
Name: Step
Type: REG_SZ
Data: 1

Value 8
Name: Type
Type: REG_SZ
Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\Speed
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Default
Type: REG_SZ
Data: 0

Value 1
Name: MiniHelp
Type: REG_SZ
Data:

Value 2
Name: ParamDesc
Type: REG_SZ
Data: Speed

Value 3
Name: Type

Type: REG_SZ
Data: enum

Key Name:
SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\Speed\Enum
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: 0
Type: REG_SZ
Data: Auto Detect

Value 1
Name: 10
Type: REG_SZ
Data: 10 Mbps

Value 2
Name: 100
Type: REG_SZ
Data: 100 Mbps

Key Name:
SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\Threshold
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 16

Value 2
Name: Max
Type: REG_SZ
Data: 200

Value 3
Name: Min
Type: REG_SZ
Data: 0

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc
Type: REG_SZ
Data: Adaptive Transmit Threshold

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7
Name: Step
Type: REG_SZ
Data: 1

Value 8
Name: Type
Type: REG_SZ
Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\UcodeSW
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Default
Type: REG_SZ
Data: 1

Value 1
Name: MiniHelp
Type: REG_SZ
Data:

Value 2
Name: ParamDesc
Type: REG_SZ
Data: Adaptive Technology

Value 3
Name: Type
Type: REG_SZ
Data: enum

Key Name:
SYSTEM\CurrentControlSet\Services\E100B2\ProsetNdi\Params\UcodeSW\Enum
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: 0
Type: REG_SZ

Data: Off

Value 1
 Name: 1
 Type: REG_SZ
 Data: On

Key Name: SYSTEM\CurrentControlSet\Services\E100B3
 Class Name: GenericClass
 Last Write Time: 12/7/97 - 5:49 PM

Value 0
 Name: ErrorControl
 Type: REG_DWORD
 Data: 0x1

Value 1
 Name: Start
 Type: REG_DWORD
 Data: 0x3

Value 2
 Name: Type
 Type: REG_DWORD
 Data: 0x4

Key Name: SYSTEM\CurrentControlSet\Services\E100B3\Linkage
 Class Name: GenericClass
 Last Write Time: 3/9/99 - 3:59 PM

Value 0
 Name: Bind
 Type: REG_MULTI_SZ
 Data: \Device\E100B3

Value 1
 Name: Export
 Type: REG_MULTI_SZ
 Data: \Device\E100B3

Value 2
 Name: Route
 Type: REG_MULTI_SZ
 Data: "E100B3"

Key Name: SYSTEM\CurrentControlSet\Services\E100B3\Linkage\Disabled
 Class Name: GenericClass
 Last Write Time: 12/7/97 - 5:49 PM

Key Name: SYSTEM\CurrentControlSet\Services\E100B3\Parameters
 Class Name: GenericClass
 Last Write Time: 3/8/99 - 10:10 AM

Value 0
 Name: Adaptive_IFS
 Type: REG_DWORD
 Data: 0x1

Value 1
 Name: BoardHasBridge
 Type: REG_DWORD
 Data: 0

Value 2
 Name: BusNumber
 Type: REG_DWORD
 Data: 0

Value 3
 Name: BusType
 Type: REG_DWORD
 Data: 0x5

Value 4
 Name: BusTypeLocal
 Type: REG_DWORD
 Data: 0x5

Value 5
 Name: Coalesce
 Type: REG_DWORD
 Data: 0

Value 6
 Name: CPUSaver
 Type: REG_DWORD
 Data: 0x600

Value 7
 Name: ForceDpx
 Type: REG_DWORD
 Data: 0x1

Value 8
 Name: Location
 Type: REG_SZ

Data: 7000

Value 9
 Name: MediaType
 Type: REG_DWORD
 Data: 0x1

Value 10
 Name: MWIEnable
 Type: REG_DWORD
 Data: 0

Value 11
 Name: NetworkAddress
 Type: REG_SZ
 Data: 0

Value 12
 Name: NumCoalesce
 Type: REG_DWORD
 Data: 0x20

Value 13
 Name: NumRfd
 Type: REG_DWORD
 Data: 0x80

Value 14
 Name: NumTbdPerTcb
 Type: REG_DWORD
 Data: 0xc

Value 15
 Name: NumTcb
 Type: REG_DWORD
 Data: 0x40

Value 16
 Name: PcNic
 Type: REG_DWORD
 Data: 0x1

Value 17
 Name: RxDmaCount
 Type: REG_DWORD
 Data: 0

Value 18
 Name: RxFifo
 Type: REG_DWORD
 Data: 0x8

Value 19

Name: SlotNumber
 Type: REG_DWORD
 Data: 0xe

Value 20
 Name: Speed
 Type: REG_DWORD
 Data: 0xa

Value 21
 Name: Threshold
 Type: REG_DWORD
 Data: 0x10

Value 22
 Name: TxDmaCount
 Type: REG_DWORD
 Data: 0

Value 23
 Name: TxFifo
 Type: REG_DWORD
 Data: 0x8

Value 24
 Name: Txmitwait
 Type: REG_DWORD
 Data: 0x1

Value 25
 Name: UcodeSW
 Type: REG_DWORD
 Data: 0x1

Value 26
 Name: UnderrunRetry
 Type: REG_DWORD
 Data: 0x1

Value 27
 Name: UseIo
 Type: REG_DWORD
 Data: 0x2

Value 28
 Name: UseManualPciAssign
 Type: REG_DWORD
 Data: 0

Value 29
 Name: VLanMode
 Type: REG_DWORD
 Data: 0

Key Name: SYSTEM\CurrentControlSet\Services\E100B3\Parameters\Tcpip
 Class Name: GenericClass
 Last Write Time: 3/9/99 - 3:59 PM
 Value 0
 Name: DefaultGateway
 Type: REG_MULTI_SZ
 Data:

 Value 1
 Name: EnableDHCP
 Type: REG_DWORD
 Data: 0

 Value 2
 Name: IPAddress
 Type: REG_MULTI_SZ
 Data: 129.103.121.2

 Value 3
 Name: IPInterfaceContext
 Type: REG_DWORD
 Data: 0x5

 Value 4
 Name: IPInterfaceContextMax
 Type: REG_DWORD
 Data: 0x5

 Value 5
 Name: LLInterface
 Type: REG_SZ
 Data:

 Value 6
 Name: PPTPFiltering
 Type: REG_DWORD
 Data: 0

 Value 7
 Name: RawIPAllowedProtocols
 Type: REG_MULTI_SZ
 Data: 0

 Value 8
 Name: SubnetMask
 Type: REG_MULTI_SZ
 Data: 255.255.255.0

Value 9
 Name: TCPAllowedPorts
 Type: REG_MULTI_SZ
 Data: 0

 Value 10
 Name: UDPAllowedPorts
 Type: REG_MULTI_SZ
 Data: 0

 Value 11
 Name: UseZeroBroadcast
 Type: REG_DWORD
 Data: 0

 Key Name: SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:24 PM

 Key Name: SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:24 PM

 Key Name: SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\Adaptive_IFS
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM
 Value 0
 Name: Base
 Type: REG_SZ
 Data: 10

 Value 1
 Name: Default
 Type: REG_SZ
 Data: 1

 Value 2
 Name: Max
 Type: REG_SZ
 Data: 255

 Value 3
 Name: Min
 Type: REG_SZ
 Data: 0

Value 4
 Name: MiniHelp
 Type: REG_SZ
 Data:

Value 5
 Name: ParamDesc
 Type: REG_SZ
 Data: Adaptive Inter-Frame Spacing

Value 6
 Name: Scale
 Type: REG_SZ
 Data: 1

Value 7
 Name: Step
 Type: REG_SZ
 Data: 1

Value 8
 Name: Type
 Type: REG_SZ
 Data: int

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\Coalesce
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: Default
 Type: REG_SZ
 Data: 0

Value 1
 Name: MiniHelp
 Type: REG_SZ
 Data:

Value 2
 Name: ParamDesc
 Type: REG_SZ
 Data: PCI Bus Efficiency

Value 3
 Name: Type
 Type: REG_SZ
 Data: enum

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\Coalesce\Enum

Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: 0
 Type: REG_SZ
 Data: Disabled

Value 1
 Name: 1
 Type: REG_SZ
 Data: Enabled

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\CPUSaver
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: Default
 Type: REG_SZ
 Data: 1536

Value 1
 Name: LeftLabel
 Type: REG_SZ
 Data: Adapter Bandwidth

Value 2
 Name: MiniHelp
 Type: REG_SZ
 Data: Sets optimal point of CPU/Adapter performance for this system. See help.

Value 3
 Name: ParamDesc
 Type: REG_SZ
 Data: Adaptive Performance Tuning

Value 4
 Name: RightLabel
 Type: REG_SZ
 Data: CPU Utilization

Value 5
 Name: Type
 Type: REG_SZ
 Data: slider

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\CPUSaver\Values
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: 0
 Type: REG_SZ
 Data: 0

Value 1
 Name: 1
 Type: REG_SZ
 Data: 256

Value 2
 Name: 10
 Type: REG_SZ
 Data: 2560

Value 3
 Name: 11
 Type: REG_SZ
 Data: 2816

Value 4
 Name: 12
 Type: REG_SZ
 Data: 3072

Value 5
 Name: 13
 Type: REG_SZ
 Data: 3328

Value 6
 Name: 14
 Type: REG_SZ
 Data: 3584

Value 7
 Name: 15
 Type: REG_SZ
 Data: 3840

Value 8
 Name: 16
 Type: REG_SZ
 Data: 4096

Value 9
 Name: 2
 Type: REG_SZ
 Data: 512

Value 10
 Name: 3
 Type: REG_SZ

Data: 768

Value 11
 Name: 4
 Type: REG_SZ
 Data: 1024

Value 12
 Name: 5
 Type: REG_SZ
 Data: 1280

Value 13
 Name: 6
 Type: REG_SZ
 Data: 1536

Value 14
 Name: 7
 Type: REG_SZ
 Data: 1792

Value 15
 Name: 8
 Type: REG_SZ
 Data: 2048

Value 16
 Name: 9
 Type: REG_SZ
 Data: 2304

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\ForceDpx
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: Default
 Type: REG_SZ
 Data: 0

Value 1
 Name: MiniHelp
 Type: REG_SZ
 Data:

Value 2
 Name: ParamDesc
 Type: REG_SZ
 Data: Duplex

Value 3

Name: Type
Type: REG_SZ
Data: enum

Key Name:
SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\ForceDpx\Enum
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: 0
Type: REG_SZ
Data: Auto Detect

Value 1
Name: 1
Type: REG_SZ
Data: Half-Duplex

Value 2
Name: 2
Type: REG_SZ
Data: Full-Duplex

Key Name:
SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\NetworkAddress
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Default
Type: REG_SZ
Data:

Value 1
Name: MiniHelp
Type: REG_SZ
Data:

Value 2
Name: ParamDesc
Type: REG_SZ
Data: Locally Administered Address

Value 3
Name: Type
Type: REG_SZ
Data: edit

Key Name:
SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\NumCoalesce
Class Name: <NO CLASS>

Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 8

Value 2
Name: Max
Type: REG_SZ
Data: 32

Value 3
Name: Min
Type: REG_SZ
Data: 1

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc
Type: REG_SZ
Data: Coalesce Buffers

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7
Name: Step
Type: REG_SZ
Data: 1

Value 8
Name: Type
Type: REG_SZ
Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\NumRfd
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base

```

Type:          REG_SZ
Data:          10

Value 1
Name:          Default
Type:          REG_SZ
Data:          48

Value 2
Name:          Max
Type:          REG_SZ
Data:          1024

Value 3
Name:          Min
Type:          REG_SZ
Data:          1

Value 4
Name:          MiniHelp
Type:          REG_SZ
Data:

Value 5
Name:          ParamDesc
Type:          REG_SZ
Data:          Receive Buffers

Value 6
Name:          Scale
Type:          REG_SZ
Data:          1

Value 7
Name:          Step
Type:          REG_SZ
Data:          1

Value 8
Name:          Type
Type:          REG_SZ
Data:          int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\NumTcb
Class Name:    <NO CLASS>
Last Write Time:  2/19/99 - 12:25 PM

Value 0
Name:          Base
Type:          REG_SZ
Data:          10

```

```

Value 1
Name:          Default
Type:          REG_SZ
Data:          32

Value 2
Name:          Max
Type:          REG_SZ
Data:          80

Value 3
Name:          Min
Type:          REG_SZ
Data:          1

Value 4
Name:          MiniHelp
Type:          REG_SZ
Data:

Value 5
Name:          ParamDesc
Type:          REG_SZ
Data:          Transmit Control Blocks

Value 6
Name:          Scale
Type:          REG_SZ
Data:          1

Value 7
Name:          Step
Type:          REG_SZ
Data:          1

Value 8
Name:          Type
Type:          REG_SZ
Data:          int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\Speed
Class Name:    <NO CLASS>
Last Write Time:  2/19/99 - 12:25 PM

Value 0
Name:          Default
Type:          REG_SZ
Data:          0

Value 1
Name:          MiniHelp
Type:          REG_SZ

```

Data:

Value 2
 Name: ParamDesc
 Type: REG_SZ
 Data: Speed

Value 3
 Name: Type
 Type: REG_SZ
 Data: enum

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\Speed\Enum
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: 0
 Type: REG_SZ
 Data: Auto Detect

Value 1
 Name: 10
 Type: REG_SZ
 Data: 10 Mbps

Value 2
 Name: 100
 Type: REG_SZ
 Data: 100 Mbps

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\Threshold
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: Base
 Type: REG_SZ
 Data: 10

Value 1
 Name: Default
 Type: REG_SZ
 Data: 16

Value 2
 Name: Max
 Type: REG_SZ
 Data: 200

Value 3

Name: Min
 Type: REG_SZ
 Data: 0

Value 4
 Name: MiniHelp
 Type: REG_SZ
 Data:

Value 5
 Name: ParamDesc
 Type: REG_SZ
 Data: Adaptive Transmit Threshold

Value 6
 Name: Scale
 Type: REG_SZ
 Data: 1

Value 7
 Name: Step
 Type: REG_SZ
 Data: 1

Value 8
 Name: Type
 Type: REG_SZ
 Data: int

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\UcodeSW
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: Default
 Type: REG_SZ
 Data: 1

Value 1
 Name: MiniHelp
 Type: REG_SZ
 Data:

Value 2
 Name: ParamDesc
 Type: REG_SZ
 Data: Adaptive Technology

Value 3
 Name: Type
 Type: REG_SZ
 Data: enum

Key Name: SYSTEM\CurrentControlSet\Services\E100B3\ProsetNdi\Params\UcodeSW\Enum
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM
 Value 0
 Name: 0
 Type: REG_SZ
 Data: Off
 Value 1
 Name: 1
 Type: REG_SZ
 Data: On
 Key Name: SYSTEM\CurrentControlSet\Services\E100B4
 Class Name: GenericClass
 Last Write Time: 12/7/97 - 5:49 PM
 Value 0
 Name: ErrorControl
 Type: REG_DWORD
 Data: 0x1
 Value 1
 Name: Start
 Type: REG_DWORD
 Data: 0x3
 Value 2
 Name: Type
 Type: REG_DWORD
 Data: 0x4
 Key Name: SYSTEM\CurrentControlSet\Services\E100B4\Linkage
 Class Name: GenericClass
 Last Write Time: 3/9/99 - 3:59 PM
 Value 0
 Name: Bind
 Type: REG_MULTI_SZ
 Data: \Device\E100B4
 Value 1
 Name: Export
 Type: REG_MULTI_SZ
 Data: \Device\E100B4
 Value 2
 Name: Route

Type: REG_MULTI_SZ
 Data: "E100B4"

Key Name: SYSTEM\CurrentControlSet\Services\E100B4\Linkage\Disabled
 Class Name: GenericClass
 Last Write Time: 12/7/97 - 5:49 PM

Key Name: SYSTEM\CurrentControlSet\Services\E100B4\Parameters
 Class Name: GenericClass
 Last Write Time: 3/8/99 - 10:10 AM

Value 0
 Name: Adaptive_IFS
 Type: REG_DWORD
 Data: 0x1

Value 1
 Name: BoardHasBridge
 Type: REG_DWORD
 Data: 0

Value 2
 Name: BusNumber
 Type: REG_DWORD
 Data: 0x2

Value 3
 Name: BusType
 Type: REG_DWORD
 Data: 0x5

Value 4
 Name: BusTypeLocal
 Type: REG_DWORD
 Data: 0x5

Value 5
 Name: Coalesce
 Type: REG_DWORD
 Data: 0

Value 6
 Name: CPUSaver
 Type: REG_DWORD
 Data: 0x600

Value 7
 Name: ForcedDpx
 Type: REG_DWORD
 Data: 0x1

Value 8
 Name: Location
 Type: REG_SZ
 Data: 401800

Value 9
 Name: MediaType
 Type: REG_DWORD
 Data: 0x1

Value 10
 Name: MWIEnable
 Type: REG_DWORD
 Data: 0

Value 11
 Name: NetworkAddress
 Type: REG_SZ
 Data: 0

Value 12
 Name: NumCoalesce
 Type: REG_DWORD
 Data: 0x20

Value 13
 Name: NumRfd
 Type: REG_DWORD
 Data: 0x80

Value 14
 Name: NumTbdPerTcb
 Type: REG_DWORD
 Data: 0xc

Value 15
 Name: NumTcb
 Type: REG_DWORD
 Data: 0x40

Value 16
 Name: PcNic
 Type: REG_DWORD
 Data: 0x1

Value 17
 Name: RxDmaCount

Type: REG_DWORD
 Data: 0

Value 18
 Name: Rx Fifo
 Type: REG_DWORD
 Data: 0x8

Value 19
 Name: SlotNumber
 Type: REG_DWORD
 Data: 0x8

Value 20
 Name: Speed
 Type: REG_DWORD
 Data: 0xa

Value 21
 Name: Threshold
 Type: REG_DWORD
 Data: 0x10

Value 22
 Name: TxDmaCount
 Type: REG_DWORD
 Data: 0

Value 23
 Name: Tx Fifo
 Type: REG_DWORD
 Data: 0x8

Value 24
 Name: Txmitwait
 Type: REG_DWORD
 Data: 0x1

Value 25
 Name: UcodesW
 Type: REG_DWORD
 Data: 0x1

Value 26
 Name: UnderrunRetry
 Type: REG_DWORD
 Data: 0x1

Value 27
 Name: UseIo
 Type: REG_DWORD
 Data: 0x2

Value 28
Name: UseManualPciAssign
Type: REG_DWORD
Data: 0

Value 29
Name: VlanMode
Type: REG_DWORD
Data: 0

Key Name:
SYSTEM\CurrentControlSet\Services\E100B4\Parameters\Tcpip
Class Name: GenericClass
Last Write Time: 3/9/99 - 3:59 PM

Value 0
Name: DefaultGateway
Type: REG_MULTI_SZ
Data:

Value 1
Name: EnableDHCP
Type: REG_DWORD
Data: 0

Value 2
Name: IPAddress
Type: REG_MULTI_SZ
Data: 129.103.122.2

Value 3
Name: IPInterfaceContext
Type: REG_DWORD
Data: 0x4

Value 4
Name: IPInterfaceContextMax
Type: REG_DWORD
Data: 0x4

Value 5
Name: LLInterface
Type: REG_SZ
Data:

Value 6
Name: PPTPFiltering
Type: REG_DWORD
Data: 0

Value 7

Name: RawIPAllowedProtocols
Type: REG_MULTI_SZ
Data: 0

Value 8
Name: SubnetMask
Type: REG_MULTI_SZ
Data: 255.255.255.0

Value 9
Name: TCPAllowedPorts
Type: REG_MULTI_SZ
Data: 0

Value 10
Name: UDPAllowedPorts
Type: REG_MULTI_SZ
Data: 0

Value 11
Name: UseZeroBroadcast
Type: REG_DWORD
Data: 0

Key Name: SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:24 PM

Key Name: SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:24 PM

Key Name: SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\Adaptive_IFS
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 1

Value 2

Name: Max
 Type: REG_SZ
 Data: 255

Value 3
 Name: Min
 Type: REG_SZ
 Data: 0

Value 4
 Name: MiniHelp
 Type: REG_SZ
 Data:

Value 5
 Name: ParamDesc
 Type: REG_SZ
 Data: Adaptive Inter-Frame Spacing

Value 6
 Name: Scale
 Type: REG_SZ
 Data: 1

Value 7
 Name: Step
 Type: REG_SZ
 Data: 1

Value 8
 Name: Type
 Type: REG_SZ
 Data: int

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\Coalesce
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: Default
 Type: REG_SZ
 Data: 0

Value 1
 Name: MiniHelp
 Type: REG_SZ
 Data:

Value 2
 Name: ParamDesc
 Type: REG_SZ
 Data: PCI Bus Efficiency

Value 3
 Name: Type
 Type: REG_SZ
 Data: enum

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\Coalesce\Enum
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: 0
 Type: REG_SZ
 Data: Disabled

Value 1
 Name: 1
 Type: REG_SZ
 Data: Enabled

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\CPUSaver
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: Default
 Type: REG_SZ
 Data: 1536

Value 1
 Name: LeftLabel
 Type: REG_SZ
 Data: Adapter Bandwidth

Value 2
 Name: MiniHelp
 Type: REG_SZ
 Data: Sets optimal point of CPU/Adapter performance for this system. See help.

Value 3
 Name: ParamDesc
 Type: REG_SZ
 Data: Adaptive Performance Tuning

Value 4
 Name: RightLabel
 Type: REG_SZ
 Data: CPU Utilization

Value 5

Name: Type
Type: REG_SZ
Data: slider

Key Name:
SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\CPUSaver\Values
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: 0
Type: REG_SZ
Data: 0

Value 1
Name: 1
Type: REG_SZ
Data: 256

Value 2
Name: 10
Type: REG_SZ
Data: 2560

Value 3
Name: 11
Type: REG_SZ
Data: 2816

Value 4
Name: 12
Type: REG_SZ
Data: 3072

Value 5
Name: 13
Type: REG_SZ
Data: 3328

Value 6
Name: 14
Type: REG_SZ
Data: 3584

Value 7
Name: 15
Type: REG_SZ
Data: 3840

Value 8
Name: 16
Type: REG_SZ
Data: 4096

Value 9
Name: 2
Type: REG_SZ
Data: 512

Value 10
Name: 3
Type: REG_SZ
Data: 768

Value 11
Name: 4
Type: REG_SZ
Data: 1024

Value 12
Name: 5
Type: REG_SZ
Data: 1280

Value 13
Name: 6
Type: REG_SZ
Data: 1536

Value 14
Name: 7
Type: REG_SZ
Data: 1792

Value 15
Name: 8
Type: REG_SZ
Data: 2048

Value 16
Name: 9
Type: REG_SZ
Data: 2304

Key Name:
SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\ForceDpx
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Default
Type: REG_SZ
Data: 0

Value 1
Name: MiniHelp


```

Type:          REG_SZ
Data:

Value 2
Name:          ParamDesc
Type:          REG_SZ
Data:          Duplex

Value 3
Name:          Type
Type:          REG_SZ
Data:          enum

Key Name:
SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\ForceDpx\Enum
Class Name:    <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name:          0
Type:          REG_SZ
Data:          Auto Detect

Value 1
Name:          1
Type:          REG_SZ
Data:          Half-Duplex

Value 2
Name:          2
Type:          REG_SZ
Data:          Full-Duplex

Key Name:
SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\NetworkAddress
Class Name:    <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name:          Default
Type:          REG_SZ
Data:

Value 1
Name:          MiniHelp
Type:          REG_SZ
Data:

Value 2
Name:          ParamDesc
Type:          REG_SZ
Data:          Locally Administered Address

```

```

Value 3
Name:          Type
Type:          REG_SZ
Data:          edit

Key Name:
SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\NumCoalesce
Class Name:    <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name:          Base
Type:          REG_SZ
Data:          10

Value 1
Name:          Default
Type:          REG_SZ
Data:          8

Value 2
Name:          Max
Type:          REG_SZ
Data:          32

Value 3
Name:          Min
Type:          REG_SZ
Data:          1

Value 4
Name:          MiniHelp
Type:          REG_SZ
Data:

Value 5
Name:          ParamDesc
Type:          REG_SZ
Data:          Coalesce Buffers

Value 6
Name:          Scale
Type:          REG_SZ
Data:          1

Value 7
Name:          Step
Type:          REG_SZ
Data:          1

Value 8
Name:          Type
Type:          REG_SZ

```

Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\NumRfd
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 48

Value 2
Name: Max
Type: REG_SZ
Data: 1024

Value 3
Name: Min
Type: REG_SZ
Data: 1

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc
Type: REG_SZ
Data: Receive Buffers

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7
Name: Step
Type: REG_SZ
Data: 1

Value 8
Name: Type
Type: REG_SZ
Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\NumTcb
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 32

Value 2
Name: Max
Type: REG_SZ
Data: 80

Value 3
Name: Min
Type: REG_SZ
Data: 1

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc
Type: REG_SZ
Data: Transmit Control Blocks

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7
Name: Step
Type: REG_SZ
Data: 1

Value 8
Name: Type
Type: REG_SZ
Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\Speed
Class Name: <NO CLASS>

Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Default
Type: REG_SZ
Data: 0

Value 1
Name: MiniHelp
Type: REG_SZ
Data:

Value 2
Name: ParamDesc
Type: REG_SZ
Data: Speed

Value 3
Name: Type
Type: REG_SZ
Data: enum

Key Name:
SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\Speed\Enum

Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name: 0
Type: REG_SZ
Data: Auto Detect

Value 1
Name: 10
Type: REG_SZ
Data: 10 Mbps

Value 2
Name: 100
Type: REG_SZ
Data: 100 Mbps

Key Name:
SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\Threshold

Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default

Type: REG_SZ
Data: 16

Value 2
Name: Max
Type: REG_SZ
Data: 200

Value 3
Name: Min
Type: REG_SZ
Data: 0

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc
Type: REG_SZ
Data: Adaptive Transmit Threshold

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7
Name: Step
Type: REG_SZ
Data: 1

Value 8
Name: Type
Type: REG_SZ
Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\UcodeSW

Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Default
Type: REG_SZ
Data: 1

Value 1
Name: MiniHelp
Type: REG_SZ
Data:

Value 2
Name: ParamDesc
Type: REG_SZ
Data: Adaptive Technology

Value 3
Name: Type
Type: REG_SZ
Data: enum

Key Name:
SYSTEM\CurrentControlSet\Services\E100B4\ProsetNdi\Params\UcodeSW\Enum
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: 0
Type: REG_SZ
Data: Off

Value 1
Name: 1
Type: REG_SZ
Data: On

Key Name: SYSTEM\CurrentControlSet\Services\E100B5
Class Name: GenericClass
Last Write Time: 12/7/97 - 5:49 PM

Value 0
Name: ErrorControl
Type: REG_DWORD
Data: 0x1

Value 1
Name: Start
Type: REG_DWORD
Data: 0x3

Value 2
Name: Type
Type: REG_DWORD
Data: 0x4

Key Name: SYSTEM\CurrentControlSet\Services\E100B5\Linkage
Class Name: GenericClass
Last Write Time: 3/9/99 - 3:59 PM

Value 0
Name: Bind
Type: REG_MULTI_SZ
Data: \Device\E100B5

Value 1
Name: Export
Type: REG_MULTI_SZ
Data: \Device\E100B5

Value 2
Name: Route
Type: REG_MULTI_SZ
Data: "E100B5"

Key Name:
SYSTEM\CurrentControlSet\Services\E100B5\Linkage\Disabled
Class Name: GenericClass
Last Write Time: 12/7/97 - 5:49 PM

Key Name: SYSTEM\CurrentControlSet\Services\E100B5\Parameters
Class Name: GenericClass
Last Write Time: 3/8/99 - 10:10 AM

Value 0
Name: Adaptive_IFS
Type: REG_DWORD
Data: 0x1

Value 1
Name: BoardHasBridge
Type: REG_DWORD
Data: 0

Value 2
Name: BusNumber
Type: REG_DWORD
Data: 0x2

Value 3
Name: BusType
Type: REG_DWORD
Data: 0x5

Value 4
Name: BusTypeLocal
Type: REG_DWORD
Data: 0x5

Value 5	Name: Coalesce	Data: 0x40
	Type: REG_DWORD	
	Data: 0	
Value 6	Name: CPUSaver	Data: 0x1
	Type: REG_DWORD	
	Data: 0x600	
Value 7	Name: ForceDpx	Data: 0x1
	Type: REG_DWORD	
	Data: 0x1	
Value 8	Name: Location	Data: 481800
	Type: REG_SZ	
	Data: 481800	
Value 9	Name: MediaType	Data: 0x1
	Type: REG_DWORD	
	Data: 0x1	
Value 10	Name: MWIEnable	Data: 0
	Type: REG_DWORD	
	Data: 0	
Value 11	Name: NetworkAddress	Data: 0
	Type: REG_SZ	
	Data: 0	
Value 12	Name: NumCoalesce	Data: 0x20
	Type: REG_DWORD	
	Data: 0x20	
Value 13	Name: NumRfd	Data: 0x80
	Type: REG_DWORD	
	Data: 0x80	
Value 14	Name: NumTbdPerTcb	Data: 0xc
	Type: REG_DWORD	
	Data: 0xc	
Value 15	Name: NumTcb	Data: 0x1
	Type: REG_DWORD	
	Data: 0x1	
Value 16	Name: PcNic	Data: 0x1
	Type: REG_DWORD	
	Data: 0x1	
Value 17	Name: RxDmaCount	Data: 0
	Type: REG_DWORD	
	Data: 0	
Value 18	Name: RxFifo	Data: 0x8
	Type: REG_DWORD	
	Data: 0x8	
Value 19	Name: SlotNumber	Data: 0x9
	Type: REG_DWORD	
	Data: 0x9	
Value 20	Name: Speed	Data: 0xa
	Type: REG_DWORD	
	Data: 0xa	
Value 21	Name: Threshold	Data: 0x10
	Type: REG_DWORD	
	Data: 0x10	
Value 22	Name: TxDmaCount	Data: 0
	Type: REG_DWORD	
	Data: 0	
Value 23	Name: TxFifo	Data: 0x8
	Type: REG_DWORD	
	Data: 0x8	
Value 24	Name: Txmitwait	Data: 0x1
	Type: REG_DWORD	
	Data: 0x1	
Value 25	Name: UcodeSW	Data: 0x1
	Type: REG_DWORD	
	Data: 0x1	
Value 26		

Name: UnderrunRetry
 Type: REG_DWORD
 Data: 0x1

Value 27
 Name: UseIo
 Type: REG_DWORD
 Data: 0x2

Value 28
 Name: UseManualPciAssign
 Type: REG_DWORD
 Data: 0

Value 29
 Name: VLanMode
 Type: REG_DWORD
 Data: 0

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B5\Parameters\Tcpip
 Class Name: GenericClass
 Last Write Time: 3/9/99 - 3:59 PM

Value 0
 Name: DefaultGateway
 Type: REG_MULTI_SZ
 Data:

Value 1
 Name: EnableDHCP
 Type: REG_DWORD
 Data: 0

Value 2
 Name: IPAddress
 Type: REG_MULTI_SZ
 Data: 129.103.123.2

Value 3
 Name: IPInterfaceContext
 Type: REG_DWORD
 Data: 0x3

Value 4
 Name: IPInterfaceContextMax
 Type: REG_DWORD
 Data: 0x3

Value 5
 Name: LLInterface

Type: REG_SZ
 Data:

Value 6
 Name: PPTPFiltering
 Type: REG_DWORD
 Data: 0

Value 7
 Name: RawIPAllowedProtocols
 Type: REG_MULTI_SZ
 Data: 0

Value 8
 Name: SubnetMask
 Type: REG_MULTI_SZ
 Data: 255.255.255.0

Value 9
 Name: TCPAllowedPorts
 Type: REG_MULTI_SZ
 Data: 0

Value 10
 Name: UDPAllowedPorts
 Type: REG_MULTI_SZ
 Data: 0

Value 11
 Name: UseZeroBroadcast
 Type: REG_DWORD
 Data: 0

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:24 PM

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:24 PM

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\Adaptive_IFS
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: Base

```

Type:          REG_SZ
Data:          10

Value 1
Name:          Default
Type:          REG_SZ
Data:          1

Value 2
Name:          Max
Type:          REG_SZ
Data:          255

Value 3
Name:          Min
Type:          REG_SZ
Data:          0

Value 4
Name:          MiniHelp
Type:          REG_SZ
Data:

Value 5
Name:          ParamDesc
Type:          REG_SZ
Data:          Adaptive Inter-Frame Spacing

Value 6
Name:          Scale
Type:          REG_SZ
Data:          1

Value 7
Name:          Step
Type:          REG_SZ
Data:          1

Value 8
Name:          Type
Type:          REG_SZ
Data:          int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\Coalesce
Class Name:    <NO CLASS>
Last Write Time:  2/19/99 - 12:25 PM
Value 0
Name:          Default
Type:          REG_SZ
Data:          0

```

```

Value 1
Name:          MiniHelp
Type:          REG_SZ
Data:

Value 2
Name:          ParamDesc
Type:          REG_SZ
Data:          PCI Bus Efficiency

Value 3
Name:          Type
Type:          REG_SZ
Data:          enum

Key Name:
SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\Coalesce\Enum
Class Name:    <NO CLASS>
Last Write Time:  2/19/99 - 12:25 PM
Value 0
Name:          0
Type:          REG_SZ
Data:          Disabled

Value 1
Name:          1
Type:          REG_SZ
Data:          Enabled

Key Name:
SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\CPUSaver
Class Name:    <NO CLASS>
Last Write Time:  2/19/99 - 12:25 PM
Value 0
Name:          Default
Type:          REG_SZ
Data:          1536

Value 1
Name:          LeftLabel
Type:          REG_SZ
Data:          Adapter Bandwidth

Value 2
Name:          MiniHelp
Type:          REG_SZ
Data:          Sets optimal point of CPU/Adapter performance for this
system. See help.

Value 3
Name:          ParamDesc

```

Type: REG_SZ
Data: Adaptive Performance Tuning

Value 4
Name: RightLabel
Type: REG_SZ
Data: CPU Utilization

Value 5
Name: Type
Type: REG_SZ
Data: slider

Key Name:
SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\CPUSaver\Values
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: 0
Type: REG_SZ
Data: 0

Value 1
Name: 1
Type: REG_SZ
Data: 256

Value 2
Name: 10
Type: REG_SZ
Data: 2560

Value 3
Name: 11
Type: REG_SZ
Data: 2816

Value 4
Name: 12
Type: REG_SZ
Data: 3072

Value 5
Name: 13
Type: REG_SZ
Data: 3328

Value 6
Name: 14
Type: REG_SZ
Data: 3584

Value 7
Name: 15
Type: REG_SZ
Data: 3840

Value 8
Name: 16
Type: REG_SZ
Data: 4096

Value 9
Name: 2
Type: REG_SZ
Data: 512

Value 10
Name: 3
Type: REG_SZ
Data: 768

Value 11
Name: 4
Type: REG_SZ
Data: 1024

Value 12
Name: 5
Type: REG_SZ
Data: 1280

Value 13
Name: 6
Type: REG_SZ
Data: 1536

Value 14
Name: 7
Type: REG_SZ
Data: 1792

Value 15
Name: 8
Type: REG_SZ
Data: 2048

Value 16
Name: 9
Type: REG_SZ
Data: 2304

Key Name:
SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\ForceDpx

Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name: Default
Type: REG_SZ
Data: 0

Value 1
Name: MiniHelp
Type: REG_SZ
Data:

Value 2
Name: ParamDesc
Type: REG_SZ
Data: Duplex

Value 3
Name: Type
Type: REG_SZ
Data: enum

Key Name:
SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\ForceDpx\Enum
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name: 0
Type: REG_SZ
Data: Auto Detect

Value 1
Name: 1
Type: REG_SZ
Data: Half-Duplex

Value 2
Name: 2
Type: REG_SZ
Data: Full-Duplex

Key Name:
SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\NetworkAddress
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name: Default
Type: REG_SZ
Data:

Value 1

Name: MiniHelp
Type: REG_SZ
Data:
Value 2
Name: ParamDesc
Type: REG_SZ
Data: Locally Administered Address
Value 3
Name: Type
Type: REG_SZ
Data: edit

Key Name:
SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\NumCoalesce
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 8

Value 2
Name: Max
Type: REG_SZ
Data: 32

Value 3
Name: Min
Type: REG_SZ
Data: 1

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc
Type: REG_SZ
Data: Coalesce Buffers

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7

Name: Step
Type: REG_SZ
Data: 1

Value 8

Name: Type
Type: REG_SZ
Data: int

Key Name:

SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\NumRfd
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0

Name: Base
Type: REG_SZ
Data: 10

Value 1

Name: Default
Type: REG_SZ
Data: 48

Value 2

Name: Max
Type: REG_SZ
Data: 1024

Value 3

Name: Min
Type: REG_SZ
Data: 1

Value 4

Name: MiniHelp
Type: REG_SZ
Data:

Value 5

Name: ParamDesc
Type: REG_SZ
Data: Receive Buffers

Value 6

Name: Scale
Type: REG_SZ
Data: 1

Value 7

Name: Step

Type: REG_SZ
Data: 1

Value 8

Name: Type
Type: REG_SZ
Data: int

Key Name:

SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\NumTcb
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0

Name: Base
Type: REG_SZ
Data: 10

Value 1

Name: Default
Type: REG_SZ
Data: 32

Value 2

Name: Max
Type: REG_SZ
Data: 80

Value 3

Name: Min
Type: REG_SZ
Data: 1

Value 4

Name: MiniHelp
Type: REG_SZ
Data:

Value 5

Name: ParamDesc
Type: REG_SZ
Data: Transmit Control Blocks

Value 6

Name: Scale
Type: REG_SZ
Data: 1

Value 7

Name: Step
Type: REG_SZ
Data: 1

Value 8

Name: Type
Type: REG_SZ
Data: int

Key Name:

SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\Speed

Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0

Name: Default
Type: REG_SZ
Data: 0

Value 1

Name: MiniHelp
Type: REG_SZ
Data:

Value 2

Name: ParamDesc
Type: REG_SZ
Data: Speed

Value 3

Name: Type
Type: REG_SZ
Data: enum

Key Name:

SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\Speed\Enum

Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0

Name: 0
Type: REG_SZ
Data: Auto Detect

Value 1

Name: 10
Type: REG_SZ
Data: 10 Mbps

Value 2

Name: 100
Type: REG_SZ
Data: 100 Mbps

Key Name:

SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\Threshold

Class Name: <NO CLASS>

Last Write Time: 2/19/99 - 12:25 PM

Value 0

Name: Base
Type: REG_SZ
Data: 10

Value 1

Name: Default
Type: REG_SZ
Data: 16

Value 2

Name: Max
Type: REG_SZ
Data: 200

Value 3

Name: Min
Type: REG_SZ
Data: 0

Value 4

Name: MiniHelp
Type: REG_SZ
Data:

Value 5

Name: ParamDesc
Type: REG_SZ
Data: Adaptive Transmit Threshold

Value 6

Name: Scale
Type: REG_SZ
Data: 1

Value 7

Name: Step
Type: REG_SZ
Data: 1

Value 8

Name: Type
Type: REG_SZ
Data: int

Key Name:

SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\UcodeSW

Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0

Name: Default
 Type: REG_SZ
 Data: 1

Value 1
 Name: MiniHelp
 Type: REG_SZ
 Data:

Value 2
 Name: ParamDesc
 Type: REG_SZ
 Data: Adaptive Technology

Value 3
 Name: Type
 Type: REG_SZ
 Data: enum

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B5\ProsetNdi\Params\UcodesW\Enum
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: 0
 Type: REG_SZ
 Data: Off

Value 1
 Name: 1
 Type: REG_SZ
 Data: On

Key Name: SYSTEM\CurrentControlSet\Services\E100B6
 Class Name: GenericClass
 Last Write Time: 12/7/97 - 5:49 PM

Value 0
 Name: ErrorControl
 Type: REG_DWORD
 Data: 0x1

Value 1
 Name: Start
 Type: REG_DWORD
 Data: 0x3

Value 2
 Name: Type
 Type: REG_DWORD
 Data: 0x4

Key Name: SYSTEM\CurrentControlSet\Services\E100B6\Linkage
 Class Name: GenericClass
 Last Write Time: 3/9/99 - 3:59 PM

Value 0
 Name: Bind
 Type: REG_MULTI_SZ
 Data: \Device\E100B6

Value 1
 Name: Export
 Type: REG_MULTI_SZ
 Data: \Device\E100B6

Value 2
 Name: Route
 Type: REG_MULTI_SZ
 Data: "E100B6"

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B6\Linkage\Disabled
 Class Name: GenericClass
 Last Write Time: 12/7/97 - 5:49 PM

Key Name: SYSTEM\CurrentControlSet\Services\E100B6\Parameters
 Class Name: GenericClass
 Last Write Time: 3/8/99 - 10:10 AM

Value 0
 Name: Adaptive_IFS
 Type: REG_DWORD
 Data: 0x1

Value 1
 Name: BoardHasBridge
 Type: REG_DWORD
 Data: 0

Value 2
 Name: BusNumber
 Type: REG_DWORD
 Data: 0x2

Value 3

Name:	BusType	Value 14	Name:	NumTbdPerTcb
Type:	REG_DWORD	Value 14	Type:	REG_DWORD
Data:	0x5	Value 14	Data:	0xc
Value 4		Value 15	Name:	NumTcb
Name:	BusTypeLocal	Value 15	Type:	REG_DWORD
Type:	REG_DWORD	Value 15	Data:	0x40
Data:	0x5	Value 16	Name:	PcNic
Value 5		Value 16	Type:	REG_DWORD
Name:	Coalesce	Value 16	Data:	0x1
Type:	REG_DWORD	Value 17	Name:	RxDmaCount
Data:	0	Value 17	Type:	REG_DWORD
Value 6		Value 17	Data:	0
Name:	CPUSaver	Value 18	Name:	RxFifo
Type:	REG_DWORD	Value 18	Type:	REG_DWORD
Data:	0x600	Value 18	Data:	0x8
Value 7		Value 19	Name:	SlotNumber
Name:	ForceDpx	Value 19	Type:	REG_DWORD
Type:	REG_DWORD	Value 19	Data:	0xa
Data:	0x1	Value 20	Name:	Speed
Value 8		Value 20	Type:	REG_DWORD
Name:	Location	Value 20	Data:	0xa
Type:	REG_SZ	Value 21	Name:	Threshold
Data:	501800	Value 21	Type:	REG_DWORD
Value 9		Value 21	Data:	0x10
Name:	MediaType	Value 22	Name:	TxDmaCount
Type:	REG_DWORD	Value 22	Type:	REG_DWORD
Data:	0x1	Value 22	Data:	0
Value 10		Value 23	Name:	TxFifo
Name:	MWIEnable	Value 23	Type:	REG_DWORD
Type:	REG_DWORD	Value 23	Data:	0x8
Data:	0	Value 24	Name:	Txmitwait
Value 11		Value 24		
Name:	NetworkAddress			
Type:	REG_SZ			
Data:	0			
Value 12				
Name:	NumCoalesce			
Type:	REG_DWORD			
Data:	0x20			
Value 13				
Name:	NumRfd			
Type:	REG_DWORD			
Data:	0x80			

Type: REG_DWORD
Data: 0x1

Value 25
Name: UcodeSW
Type: REG_DWORD
Data: 0x1

Value 26
Name: UnderrunRetry
Type: REG_DWORD
Data: 0x1

Value 27
Name: UseIo
Type: REG_DWORD
Data: 0x2

Value 28
Name: UseManualPciAssign
Type: REG_DWORD
Data: 0

Value 29
Name: VlanMode
Type: REG_DWORD
Data: 0

Key Name:
SYSTEM\CurrentControlSet\Services\E100B6\Parameters\Tcpip
Class Name: GenericClass
Last Write Time: 3/9/99 - 3:59 PM

Value 0
Name: DefaultGateway
Type: REG_MULTI_SZ
Data:

Value 1
Name: EnableDHCP
Type: REG_DWORD
Data: 0

Value 2
Name: IPAddress
Type: REG_MULTI_SZ
Data: 129.103.124.2

Value 3
Name: IPInterfaceContext
Type: REG_DWORD

Data: 0x2

Value 4
Name: IPInterfaceContextMax
Type: REG_DWORD
Data: 0x2

Value 5
Name: LLInterface
Type: REG_SZ
Data:

Value 6
Name: PPTPFiltering
Type: REG_DWORD
Data: 0

Value 7
Name: RawIPAllowedProtocols
Type: REG_MULTI_SZ
Data: 0

Value 8
Name: SubnetMask
Type: REG_MULTI_SZ
Data: 255.255.255.0

Value 9
Name: TCPAllowedPorts
Type: REG_MULTI_SZ
Data: 0

Value 10
Name: UDPAllowedPorts
Type: REG_MULTI_SZ
Data: 0

Value 11
Name: UseZeroBroadcast
Type: REG_DWORD
Data: 0

Key Name: SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:24 PM

Key Name:
SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params

Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:24 PM

Key Name:
SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\Adaptive_IFS
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 1

Value 2
Name: Max
Type: REG_SZ
Data: 255

Value 3
Name: Min
Type: REG_SZ
Data: 0

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc
Type: REG_SZ
Data: Adaptive Inter-Frame Spacing

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7
Name: Step
Type: REG_SZ
Data: 1

Value 8
Name: Type
Type: REG_SZ
Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\Coalesce
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Default
Type: REG_SZ
Data: 0

Value 1
Name: MiniHelp
Type: REG_SZ
Data:

Value 2
Name: ParamDesc
Type: REG_SZ
Data: PCI Bus Efficiency

Value 3
Name: Type
Type: REG_SZ
Data: enum

Key Name:
SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\Coalesce\Enum
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: 0
Type: REG_SZ
Data: Disabled

Value 1
Name: 1
Type: REG_SZ
Data: Enabled

Key Name:
SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\CPUSaver
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Default
Type: REG_SZ
Data: 1536

Value 1
Name: LeftLabel
Type: REG_SZ
Data: Adapter Bandwidth

Value 2
Name: MiniHelp
Type: REG_SZ
Data: Sets optimal point of CPU/Adapter performance for this system. See help.

Value 3
Name: ParamDesc
Type: REG_SZ
Data: Adaptive Performance Tuning

Value 4
Name: RightLabel
Type: REG_SZ
Data: CPU Utilization

Value 5
Name: Type
Type: REG_SZ
Data: slider

Key Name:
SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\CPUSaver\Values
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: 0
Type: REG_SZ
Data: 0

Value 1
Name: 1
Type: REG_SZ
Data: 256

Value 2
Name: 10
Type: REG_SZ
Data: 2560

Value 3
Name: 11
Type: REG_SZ
Data: 2816

Value 4
Name: 12
Type: REG_SZ
Data: 3072

Value 5

Name: 13
Type: REG_SZ
Data: 3328

Value 6
Name: 14
Type: REG_SZ
Data: 3584

Value 7
Name: 15
Type: REG_SZ
Data: 3840

Value 8
Name: 16
Type: REG_SZ
Data: 4096

Value 9
Name: 2
Type: REG_SZ
Data: 512

Value 10
Name: 3
Type: REG_SZ
Data: 768

Value 11
Name: 4
Type: REG_SZ
Data: 1024

Value 12
Name: 5
Type: REG_SZ
Data: 1280

Value 13
Name: 6
Type: REG_SZ
Data: 1536

Value 14
Name: 7
Type: REG_SZ
Data: 1792

Value 15
Name: 8
Type: REG_SZ
Data: 2048

Value 16
Name: 9
Type: REG_SZ
Data: 2304

Key Name:
SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\ForceDpx
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Default
Type: REG_SZ
Data: 0

Value 1
Name: MiniHelp
Type: REG_SZ
Data:

Value 2
Name: ParamDesc
Type: REG_SZ
Data: Duplex

Value 3
Name: Type
Type: REG_SZ
Data: enum

Key Name:
SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\ForceDpx\Enum
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: 0
Type: REG_SZ
Data: Auto Detect

Value 1
Name: 1
Type: REG_SZ
Data: Half-Duplex

Value 2
Name: 2
Type: REG_SZ
Data: Full-Duplex

Key Name:
SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\NetworkAddress
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Default
Type: REG_SZ
Data:

Value 1
Name: MiniHelp
Type: REG_SZ
Data:

Value 2
Name: ParamDesc
Type: REG_SZ
Data: Locally Administered Address

Value 3
Name: Type
Type: REG_SZ
Data: edit

Key Name:
SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\NumCoalesce
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 8

Value 2
Name: Max
Type: REG_SZ
Data: 32

Value 3
Name: Min
Type: REG_SZ
Data: 1

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc
Type: REG_SZ
Data: Coalesce Buffers

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7
Name: Step
Type: REG_SZ
Data: 1

Value 8
Name: Type
Type: REG_SZ
Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\NumRfd
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 48

Value 2
Name: Max
Type: REG_SZ
Data: 1024

Value 3
Name: Min
Type: REG_SZ
Data: 1

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc

Type: REG_SZ
Data: Receive Buffers

Value 6
Name: Scale
Type: REG_SZ
Data: 1

Value 7
Name: Step
Type: REG_SZ
Data: 1

Value 8
Name: Type
Type: REG_SZ
Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\NumTcb
Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM

Value 0
Name: Base
Type: REG_SZ
Data: 10

Value 1
Name: Default
Type: REG_SZ
Data: 32

Value 2
Name: Max
Type: REG_SZ
Data: 80

Value 3
Name: Min
Type: REG_SZ
Data: 1

Value 4
Name: MiniHelp
Type: REG_SZ
Data:

Value 5
Name: ParamDesc
Type: REG_SZ
Data: Transmit Control Blocks

Value 6
 Name: Scale
 Type: REG_SZ
 Data: 1

Value 7
 Name: Step
 Type: REG_SZ
 Data: 1

Value 8
 Name: Type
 Type: REG_SZ
 Data: int

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\Speed
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: Default
 Type: REG_SZ
 Data: 0

Value 1
 Name: MiniHelp
 Type: REG_SZ
 Data:

Value 2
 Name: ParamDesc
 Type: REG_SZ
 Data: Speed

Value 3
 Name: Type
 Type: REG_SZ
 Data: enum

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\Speed\Enum
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: 0
 Type: REG_SZ
 Data: Auto Detect

Value 1
 Name: 10
 Type: REG_SZ

Data: 10 Mbps

Value 2
 Name: 100
 Type: REG_SZ
 Data: 100 Mbps

Key Name:
 SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\Threshold
 Class Name: <NO CLASS>
 Last Write Time: 2/19/99 - 12:25 PM

Value 0
 Name: Base
 Type: REG_SZ
 Data: 10

Value 1
 Name: Default
 Type: REG_SZ
 Data: 16

Value 2
 Name: Max
 Type: REG_SZ
 Data: 200

Value 3
 Name: Min
 Type: REG_SZ
 Data: 0

Value 4
 Name: MiniHelp
 Type: REG_SZ
 Data:

Value 5
 Name: ParamDesc
 Type: REG_SZ
 Data: Adaptive Transmit Threshold

Value 6
 Name: Scale
 Type: REG_SZ
 Data: 1

Value 7
 Name: Step
 Type: REG_SZ
 Data: 1

Value 8

Name: Type
Type: REG_SZ
Data: int

Key Name:
SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\UcodeSW

Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name: Default
Type: REG_SZ
Data: 1

Value 1
Name: MiniHelp
Type: REG_SZ
Data:

Value 2
Name: ParamDesc
Type: REG_SZ
Data: Adaptive Technology

Value 3
Name: Type
Type: REG_SZ
Data: enum

Key Name:
SYSTEM\CurrentControlSet\Services\E100B6\ProsetNdi\Params\UcodeSW\Enum

Class Name: <NO CLASS>
Last Write Time: 2/19/99 - 12:25 PM
Value 0
Name: 0
Type: REG_SZ
Data: Off

Value 1
Name: 1
Type: REG_SZ
Data: On

Profile: HTML_19000x
File Path: F:\benchcrf\profiles\HTML_19000x.pro
Version: 1.0.1

Number of Engines: 20

Name: DRIVER_A
Description:
Directory: E:\benchcrf\logs\driver_a.log
Machine: blau
Parameter Set: All_Times
Index: 0
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER1134316890
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER_B
Description:
Directory: E:\benchcrf\logs\driver_b.log
Machine: blau
Parameter Set: All_Times
Index: 100000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER2134900640
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER_C
Description:
Directory: E:\benchcrf\logs\driver_c.log
Machine: blau
Parameter Set: All_Times
Index: 200000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER3134946468
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 2

Name: DRIVER_D
Description:
Directory: E:\benchcrf\logs\driver_d.log
Machine: blau
Parameter Set: All_Times

This section discloses the RTE parameters parameters used on the Primergy 470 client system.

Index: 300000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER4134975578
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 3

Name: DRIVER_E
Description:
Directory: E:\benchcrf\logs\driver_e.log
Machine: blau
Parameter Set: All_Times
Index: 400000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER5134997578
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 4

Name: DRIVER_F
Description:
Directory: G:\benchcrf\logs\driver_f.log
Machine: oliv
Parameter Set: All_Times
Index: 500000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER6135083421
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER_G
Description:
Directory: G:\benchcrf\logs\driver_g.log
Machine: oliv
Parameter Set: All_Times
Index: 600000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER7135111875
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER_H
Description:

Directory: G:\benchcrf\logs\driver_h.log
Machine: oliv
Parameter Set: All_Times
Index: 700000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER8135133406
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 2

Name: DRIVER_I
Description:
Directory: G:\benchcrf\logs\driver_i.log
Machine: oliv
Parameter Set: All_Times
Index: 800000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER9135155671
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 3

Name: DRIVER_J
Description:
Directory: G:\benchcrf\logs\driver_j.log
Machine: oliv
Parameter Set: All_Times
Index: 900000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER10135189187
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 2

Name: DRIVER_K
Description:
Directory: F:\benchcrf\logs\driver_k.log
Machine: tuerkis
Parameter Set: All_Times
Index: 1000000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER11135317187
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 0

Name: DRIVER_L
Description:
Directory: F:\benchcrf\logs\driver_l.log
Machine: tuerkis
Parameter Set: All_Times
Index: 1100000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER12135342140
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER_M
Description:
Directory: F:\benchcrf\logs\driver_m.log
Machine: tuerkis
Parameter Set: All_Times
Index: 1200000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER13135369953
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 2

Name: DRIVER_N
Description:
Directory: F:\benchcrf\logs\driver_n.log
Machine: tuerkis
Parameter Set: All_Times
Index: 1300000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER14135389546
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 3

Name: DRIVER_O
Description:
Directory: F:\benchcrf\logs\driver_o.log
Machine: tuerkis
Parameter Set: All_Times
Index: 1400000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER15135407015
Connect Rate: 150

Start Rate: 150
CLIENT_NURAND: 233
CPU: 1

Name: DRIVER_P
Description:
Directory: D:\benchcrf\logs\driver_p.log
Machine: rosa
Parameter Set: All_Times
Index: 1500000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER16135995546
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 5

Name: DRIVER_Q
Description:
Directory: D:\benchcrf\logs\driver_q.log
Machine: rosa
Parameter Set: All_Times
Index: 1600000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER17136631515
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 6

Name: DRIVER_R
Description:
Directory: D:\benchcrf\logs\driver_r.log
Machine: rosa
Parameter Set: All_Times
Index: 1700000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER18136671781
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 7

Name: DRIVER_S
Description:
Directory: D:\benchcrf\logs\driver_s.log
Machine: rosa
Parameter Set: All_Times
Index: 1800000000
Seed: 68291

Configured Users: 950
Pipe Name: DRIVER19136693703
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 2

Name: DRIVER_T
Description:
Directory: D:\benchcrf\logs\driver_t.log
Machine: rosa
Parameter Set: All_Times
Index: 1900000000
Seed: 68291
Configured Users: 950
Pipe Name: DRIVER20136719109
Connect Rate: 150
Start Rate: 150
CLIENT_NURAND: 233
CPU: 3

Number of User groups: 20

Driver Engine: DRIVER_A
IIS Server: braun0
SQL Server: spacelab
User: sa
Protocol: Html
w_id Range: 1 - 95
w_id Max Warehouse: 1900
Scale: Normal
User Count: 950
District id: 1
Scale Down: No

Driver Engine: DRIVER_J
IIS Server: gelb4
SQL Server: spacelab
User: sa
Protocol: Html
w_id Range: 856 - 950
w_id Max Warehouse: 1900
Scale: Normal
User Count: 950
District id: 1
Scale Down: No

Driver Engine: DRIVER_K
IIS Server: gruen0
SQL Server: spacelab
User: sa
Protocol: Html
w_id Range: 951 - 1045

w_id Max Warehouse: 1900
Scale: Normal
User Count: 950
District id: 1
Scale Down: No

Driver Engine: DRIVER_L
IIS Server: gruen1
SQL Server: spacelab
User: sa
Protocol: Html
w_id Range: 1046 - 1140
w_id Max Warehouse: 1900
Scale: Normal
User Count: 950
District id: 1
Scale Down: No

Driver Engine: DRIVER_M
IIS Server: gruen2
SQL Server: spacelab
User: sa
Protocol: Html
w_id Range: 1141 - 1235
w_id Max Warehouse: 1900
Scale: Normal
User Count: 950
District id: 1
Scale Down: No

Driver Engine: DRIVER_N
IIS Server: gruen3
SQL Server: spacelab
User: sa
Protocol: Html
w_id Range: 1236 - 1330
w_id Max Warehouse: 1900
Scale: Normal
User Count: 950
District id: 1
Scale Down: No

Driver Engine: DRIVER_O
IIS Server: gruen4
SQL Server: spacelab
User: sa
Protocol: Html
w_id Range: 1331 - 1425
w_id Max Warehouse: 1900
Scale: Normal
User Count: 950
District id: 1
Scale Down: No

Driver Engine: DRIVER_P
IIS Server: weiss0
SQL Server: spacelab
User: sa
Protocol: Html
w_id Range: 1426 - 1520
w_id Max Warehouse: 1900
Scale: Normal
User Count: 950
District id: 1
Scale Down: No

Driver Engine: DRIVER_Q
IIS Server: weiss1
SQL Server: spacelab
User: sa
Protocol: Html
w_id Range: 1521 - 1615
w_id Max Warehouse: 1900
Scale: Normal
User Count: 950
District id: 1
Scale Down: No

Driver Engine: DRIVER_R
IIS Server: weiss2
SQL Server: spacelab
User: sa
Protocol: Html
w_id Range: 1616 - 1710
w_id Max Warehouse: 1900
Scale: Normal
User Count: 950
District id: 1
Scale Down: No

Driver Engine: DRIVER_S
IIS Server: weiss3
SQL Server: spacelab
User: sa
Protocol: Html
w_id Range: 1711 - 1805
w_id Max Warehouse: 1900
Scale: Normal
User Count: 950
District id: 1
Scale Down: No

Driver Engine: DRIVER_B
IIS Server: braun1
SQL Server: spacelab
User: sa

Protocol: Html
w_id Range: 96 - 190
w_id Max Warehouse: 1900
Scale: Normal
User Count: 950
District id: 1
Scale Down: No

Driver Engine: DRIVER_T
IIS Server: weiss4
SQL Server: spacelab
User: sa
Protocol: Html
w_id Range: 1806 - 1900
w_id Max Warehouse: 1900
Scale: Normal
User Count: 950
District id: 1
Scale Down: No

Driver Engine: DRIVER_C
IIS Server: braun2
SQL Server: spacelab
User: sa
Protocol: Html
w_id Range: 191 - 285
w_id Max Warehouse: 1900
Scale: Normal
User Count: 950
District id: 1
Scale Down: No

Driver Engine: DRIVER_D
IIS Server: braun3
SQL Server: spacelab
User: sa
Protocol: Html
w_id Range: 286 - 380
w_id Max Warehouse: 1900
Scale: Normal
User Count: 950
District id: 1
Scale Down: No

Driver Engine: DRIVER_E
IIS Server: braun4
SQL Server: spacelab
User: sa
Protocol: Html
w_id Range: 381 - 475
w_id Max Warehouse: 1900
Scale: Normal
User Count: 950

District id: 1
 Scale Down: No

 Driver Engine: DRIVER_F
 IIS Server: gelb0
 SQL Server: spacelab
 User: sa
 Protocol: Html
 w_id Range: 476 - 570
 w_id Max Warehouse: 1900
 Scale: Normal
 User Count: 950
 District id: 1
 Scale Down: No

Driver Engine: DRIVER_G
 IIS Server: gelb1
 SQL Server: spacelab
 User: sa
 Protocol: Html
 w_id Range: 571 - 665
 w_id Max Warehouse: 1900
 Scale: Normal
 User Count: 950
 District id: 1
 Scale Down: No

Driver Engine: DRIVER_H
 IIS Server: gelb2
 SQL Server: spacelab
 User: sa
 Protocol: Html
 w_id Range: 666 - 760
 w_id Max Warehouse: 1900
 Scale: Normal
 User Count: 950
 District id: 1
 Scale Down: No

Driver Engine: DRIVER_I
 IIS Server: gelb3
 SQL Server: spacelab
 User: sa
 Protocol: Html
 w_id Range: 761 - 855
 w_id Max Warehouse: 1900
 Scale: Normal
 User Count: 950
 District id: 1
 Scale Down: No

Number of Parameter Sets: 2

All_Times						
HTML Param. Set (All Times: Default)						
Menu		Txn	Think	Key	RT	RT
Delay		Weight	Time	Time	Delay	Fence
5.00	0.10	New Order	44.83	12.05	18.01	0.10
5.00	0.10	Payment	43.05	12.05	3.01	0.10
5.00	0.10	Delivery	4.04	5.05	2.01	0.10
20.00	0.10	Stock Level	4.04	5.05	2.01	0.10
5.00	0.10	Order Status	4.04	10.05	2.01	0.10
~Default						
Default Parameter Set						
Menu		Txn	Think	Key	RT	RT
Delay		Weight	Time	Time	Delay	Fence
5.00	0.10	New Order	10.00	12.05	18.01	0.10
5.00	0.10	Payment	10.00	12.05	3.01	0.10
5.00	0.10	Delivery	1.00	5.05	2.01	0.10
20.00	0.10	Stock Level	1.00	5.05	2.01	0.10
5.00	0.10	Order Status	1.00	10.05	2.01	0.10

This section discloses the Transaction monitor tunable parameters parameters used on the Primergy 470 client system.

* for each of the 4 clients

*RESOURCES
 IPCKEY 133133

 MAXACCESSERS 800
 MAXSERVERS 120
 MAXSERVICES 800
 MODEL SHM

```
MASTER WEISS
LDBAL Y
SCANUNIT 15
BLOCKTIME 120
BBLQUERY 120

*MACHINES
DEFAULT:

"WEISS" LMID= WEISS
TUXDIR="C:\tuxedo"
APPDIR="C:\InetPub\wwwroot"
TUXCONFIG="C:\InetPub\wwwroot\tuxconfig"
ULOGPFX="C:\InetPub\wwwroot\ULOG"
TYPE="WinNT"
UID= 0
GID= 0
```

```
*GROUPS
GROUPNO
LMID=WEISS GRPNO=1 OPENINFO=NONE

GROUPPAY
LMID=WEISS GRPNO=2 OPENINFO=NONE

GROUPOS
LMID=WEISS GRPNO=3 OPENINFO=NONE

GROUPSL
LMID=WEISS GRPNO=4 OPENINFO=NONE

GROUPDEL
LMID=WEISS GRPNO=5 OPENINFO=NONE
```

```
*SERVERS
DEFAULT:

neworder SRVGRP=GROUPNO
SRVID=100
MIN=21 MAX=25
CLOPT="-A -- -Sspacelab"
RQADDR=newq REPLYQ=Y
```

```
payment SRVGRP=GROUPPAY
SRVID=200
MIN=8 MAX=20
CLOPT="-A -- -Sspacelab"
RQADDR=payq REPLYQ=Y
```

```
orderstatus SRVGRP=GROUPOS
SRVID=300
MIN=2 MAX=5
CLOPT="-A -- -Sspacelab"
```

```
RQADDR=ordq REPLYQ=Y

stocklevel SRVGRP=GROUPSL
SRVID=400
MIN=18 MAX=25
CLOPT="-A -- -Sspacelab"
RQADDR=stkq REPLYQ=Y

delivery SRVGRP=GROUPDEL
SRVID=500
MIN=4 MAX=8
CLOPT="-A -- -Sspacelab -F"
RQADDR=delq REPLYQ=N
```

```
*SERVICES
```

This section discloses the Microsoft SQL Server 7.0 Enterprise Edition parameters used on the Primergy 870-40 server system.

```
Microsoft SQL Server Startup Parameters:
sqlservr -c -x -T3502 -g40
```

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 start and end of each checkpoint
- g40 memory in MB reserved for memory requests outside the buffer pool

Microsoft SQL Server Stack Size:

The default stack size for Microsoft SQL Server 7.0 was changed using the EDITBIN utility:
editbin /STACK:131072

Microsoft SQL Server Configuration Parameters:

- ```
1> 2> 3> 4> 5> 6> 7> 8> 9> 10> 11>
-- File: VERSION.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.00
-- Copyright Microsoft, 1996
-- Purpose: Returns SQL Server version string
```

```
print " "
select convert(char(30), getdate(), 9)
print " "
```

```

Mar 8 1999 11:49:52:570AM
```

(1 row affected)

```
1> 2> 3>
select @@version
```

```


Microsoft SQL Server 7.00 - 7.00.623 (Intel X86)
Nov 13 1998 02:37:14
Cop
yright (c) 1988-1998 Microsoft Corporation
Enterprise Edition on Windows NT 4.0 (Build 1381: Service Pack 4)
```

(1 row affected)

- ```
1> 2>
1> 2> 3> 4> 5> 6> 7> 8> 9> 10>
-- File:      CONFIG.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.00
--           Copyright Microsoft, 1996
-- Purpose:   Collects SQL Server configuration parameters
```

```
print " "
select convert(char(30), getdate(), 9)
print " "
```

Mar 8 1999 11:49:53:367AM

(1 row affected)

1> 2> 3> DBCC execution completed. If DBCC printed error messages, contact your system administrator.
Configuration option changed. Run the RECONFIGURE statement to install.

sp_configure "show advanced",1
1> 2> reconfigure with override
1> 2> sp_configure

name	minimum	maximum
config_value run_value		
-----	-----	-----
affinity mask	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
0	0	
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
2147483647	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)	30	30	0	2147483647
remote proc trans	0	0	0	1
remote query timeout (s)	0	0	0	2147483647
resource timeout (s)	10	10	5	2147483647
scan for startup procs	0	0	0	1
set working set size	0	0	0	1
show advanced options	1	1	0	1
spin counter	10000	10000	1	2147483647
time slice (ms)	100	100	50	1000
two digit year cutoff	2049	2049	1753	9999
Unicode comparison style	0	0	0	2147483647
Unicode locale id	33280	33280	0	2147483647
user connections	220	220	0	32767
user options	0	0	0	4095

1>

This section discloses hardware information of the Primergy 870-40 server system.

***** NT diagnostic *****

Microsoft Diagnostics Report For \\SPACELAB

OS Version Report

Microsoft (R) Windows NT (TM) Server
Version 4.0 (Build 1381: Service Pack 4) x86 Multiprocessor Free
Registered Owner: TPC-C, Siemens
Product Number: 36397-OEM-0029424-01381

System Report

System: AT/AT COMPATIBLE
Hardware Abstraction Layer: MPS 1.4 - APIC platform
BIOS Date: 02/12/99
BIOS Version: PhoenixBIOS Version 4.06 Rev. 2

Processor list:
0: x86 Family 6 Model 7 Stepping 2 GenuineIntel ~500 Mhz
1: x86 Family 6 Model 7 Stepping 2 GenuineIntel ~500 Mhz
2: x86 Family 6 Model 7 Stepping 2 GenuineIntel ~500 Mhz
3: x86 Family 6 Model 7 Stepping 2 GenuineIntel ~500 Mhz

Video Display Report

BIOS Date: 05/21/97
BIOS Version: CL-GD5446 PCI VGA BIOS Version 1.33
Rel. 1.00

Adapter:
Setting: 800 x 600 x 256
75 Hz
Type: cirrus compatible display adapter
String: Cirrus Logic Compatible
Memory: 2 MB
Chip Type: Cirrus Logic 5446
DAC Type: Integrated RAMDAC

Driver:
Vendor: Microsoft Corporation
File(s): cirrus.sys, vga.dll, cirrus.dll, vga256.dll,
vga64K.dll
Version: 4.00, 4.0.0

Drives Report

C:\ (Local - NTFS) WINNT40EE Total: 2,040,223 KB, Free: 1,772,306 KB
Serial Number: 369B - 5B89
Bytes per cluster: 512
Sectors per cluster: 1
Filename length: 255
D:\ (Local - NTFS) TPCC Total: 6,827,592 KB, Free: 1,265,076 KB
Serial Number: 6D36 - BE32
Bytes per cluster: 512
Sectors per cluster: 8
Filename length: 255
X:\ (Local - NTFS) BACKUP1 Total: 106,695,660 KB, Free: 34,138,188 KB
Serial Number: 9CBA - 487A
Bytes per cluster: 512
Sectors per cluster: 8
Filename length: 255
Y:\ (Local - NTFS) BACKUP2 Total: 106,695,660 KB, Free: 32,128,224 KB
Serial Number: 10AC - 718B
Bytes per cluster: 512
Sectors per cluster: 8
Filename length: 255

Memory Report

Handles: 2,467
Threads: 115
Processes: 17

Physical Memory (K)
Total: 3,734,948
Available: 3,507,692
File Cache: 13,660

Kernel Memory (K)
Total: 14,840
Paged: 8,692
Nonpaged: 6,148

Commit Charge (K)

Total: 93,564
 Limit: 7,781,608
 Peak: 3,097,008

Pagefile Space (K)
 Total: 4,193,280
 Total in use: 5,036
 Peak: 7,436

D:\pagefile.sys
 Total: 4,193,280
 Total in use: 5,036
 Peak: 7,436

Services Report

```

-----
----
FAST Remote Services Agent           Stopped   (Disabled)
  C:\Program Files\Adaptec\AAC\System\AfaAgent.Exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Own Process, Interactive
  Service Dependencies:
    RpcSs

Alerter                               Stopped   (Manual)
  C:\WINNTPCC\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    LanmanWorkstation

Computer Browser                     Stopped   (Manual)
  C:\WINNTPCC\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    LanmanWorkstation
    LanmanServer
    LmHosts

ClipBook Server                      Stopped   (Manual)
  C:\WINNTPCC\system32\clipsrv.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Own Process
  Service Dependencies:
    NetDDE

DHCP Client (TDI)                   Stopped   (Disabled)
  C:\WINNTPCC\System32\services.exe
  Service Account Name: LocalSystem
  
```

```

  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    Tcpip
    Afd
    NetBT
EventLog (Event log)                 Running
(Automatic)
  C:\WINNTPCC\system32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
GAM Server Services                  Stopped   (Manual)
  C:\WINNTPCC\SYSTEM32\GAMSERV\gamscm.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Own Process
Server                                Running
(Automatic)
  C:\WINNTPCC\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Group Dependencies:
    TDI
Workstation (NetworkProvider)        Running
(Automatic)
  C:\WINNTPCC\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Group Dependencies:
    TDI
License Logging Service               Stopped   (Manual)
  C:\WINNTPCC\System32\llssrv.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Own Process
TCP/IP NetBIOS Helper                Stopped   (Manual)
  C:\WINNTPCC\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Group Dependencies:
    NetworkProvider
Messenger                             Stopped   (Manual)
  C:\WINNTPCC\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    LanmanWorkstation
  
```

NetBios			Directory Replicator	Stopped	(Manual)
MSDTC (MS Transactions)	Stopped	(Manual)	C:\WINNTPCC\System32\lmrepl.exe		
C:\WINNTPCC\System32\msdtc.exe			Service Account Name: LocalSystem		
Service Account Name: LocalSystem			Error Severity: Normal		
Error Severity: Normal			Service Flags: Own Process		
Service Flags: Own Process			Service Dependencies:		
Service Dependencies:			LanmanWorkstation		
RPCSS			LanmanServer		
NTLMSSP			Remote Procedure Call (RPC) Locator	Stopped	(Manual)
MSSQLServer	Stopped	(Manual)	C:\WINNTPCC\System32\LOCATOR.EXE		
D:\MSSQL7\binn\sqlservr.exe			Service Account Name: LocalSystem		
Service Account Name: LocalSystem			Error Severity: Normal		
Error Severity: Normal			Service Flags: Own Process		
Service Flags: Own Process			Service Dependencies:		
Network DDE (NetDDEGroup)	Stopped	(Disabled)	LanmanWorkstation		
C:\WINNTPCC\system32\netdde.exe			Rdr		
Service Account Name: LocalSystem			Remote Procedure Call (RPC) Service	Running	
Error Severity: Normal			(Automatic)		
Service Flags: Shared Process			C:\WINNTPCC\system32\RpcSs.exe		
Service Dependencies:			Service Account Name: LocalSystem		
NetDDESDM			Error Severity: Normal		
Network DDE DSDM	Stopped	(Disabled)	Service Flags: Own Process		
C:\WINNTPCC\system32\netdde.exe			Schedule	Stopped	(Manual)
Service Account Name: LocalSystem			C:\WINNTPCC\System32\AtSvc.Exe		
Error Severity: Normal			Service Account Name: LocalSystem		
Service Flags: Shared Process			Error Severity: Normal		
Net Logon (RemoteValidation)	Stopped	(Manual)	Service Flags: Own Process		
C:\WINNTPCC\System32\lsass.exe			Spooler (SpoolerGroup)	Stopped	(Disabled)
Service Account Name: LocalSystem			C:\WINNTPCC\system32\spoolss.exe		
Error Severity: Normal			Service Account Name: LocalSystem		
Service Flags: Shared Process			Error Severity: Normal		
Service Dependencies:			Service Flags: Own Process, Interactive		
LanmanWorkstation			SQLServerAgent	Stopped	(Manual)
LmHosts			D:\MSSQL7\binn\sqlagent.exe		
NT LM Security Support Provider	Running	(Manual)	Service Account Name: LocalSystem		
C:\WINNTPCC\System32\SERVICES.EXE			Error Severity: Normal		
Service Account Name: LocalSystem			Service Flags: Own Process		
Error Severity: Normal			Service Dependencies:		
Service Flags: Shared Process			MSSQLServer		
Plug and Play (PlugPlay)	Stopped	(Manual)	Telephony Service	Stopped	(Manual)
C:\WINNTPCC\system32\services.exe			C:\WINNTPCC\system32\tapisrv.exe		
Service Account Name: LocalSystem			Service Account Name: LocalSystem		
Error Severity: Normal			Error Severity: Normal		
Service Flags: Shared Process			Service Flags: Own Process		
Protected Storage	Running		UPS	Stopped	(Manual)
(Automatic)			C:\WINNTPCC\System32\ups.exe		
c:\winntppcc\system32\pstores.exe			Service Account Name: LocalSystem		
Service Account Name: LocalSystem			Error Severity: Normal		
Error Severity: Normal			Service Flags: Own Process		
Service Flags: Own Process, Interactive					
Service Dependencies:					
RpcSs					

Drivers Report

```

-----
----
Abiosdsk (Primary disk)           Stopped (Disabled)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
afacomm (SCSI miniport)          Stopped (Disabled)
  C:\WINNTPCC\System32\DRIVERS\afacomm.sys
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
afadisk (Primary Disk)           Stopped (Disabled)
  C:\WINNTPCC\System32\DRIVERS\afadisk.sys
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
AfaFs (File System)              Stopped (Disabled)
  Error Severity: Normal
  Service Flags: File System Driver, Shared Process
AfaRec (File System)             Stopped (Disabled)
  Error Severity: Normal
  Service Flags: File System Driver, Shared Process
afasa (SCSI miniport)           Stopped (Disabled)
  C:\WINNTPCC\System32\DRIVERS\afasa.sys
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
afascsi (port)                  Stopped (Disabled)
  C:\WINNTPCC\System32\DRIVERS\afascsi.sys
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
  Group Dependencies:
    SCSI miniport
AFD Networking Support Environment (TDI) Running
(Automatic)
  C:\WINNTPCC\System32\drivers\afd.sys
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Ahal154x (SCSI miniport)        Stopped (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Ahal174x (SCSI miniport)        Stopped (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
aic78xx (SCSI miniport)         Stopped (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Always (SCSI miniport)          Stopped (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
amiOnt (SCSI miniport)          Stopped (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
amsint (SCSI miniport)          Stopped (Disabled)
  Error Severity: Normal

```

```

  Service Flags: Kernel Driver, Shared Process
Arrow (SCSI miniport)           Stopped (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
atapi (SCSI miniport)           Stopped (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Atdisk (Primary disk)           Stopped (Disabled)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
ati (Video)                     Stopped (Disabled)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
Beep (Base)                     Running (System)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
BusLogic (SCSI miniport)        Stopped (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Busmouse (Pointer Port)         Stopped (Disabled)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
Cdaudio (Filter)                Stopped (System)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
Cdfs (File system)              Running (Disabled)
  Error Severity: Normal
  Service Flags: File System Driver, Shared Process
  Group Dependencies:
    SCSI CDROM Class
Cdrom (SCSI CDROM Class)        Running (System)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
  Group Dependencies:
    SCSI miniport
Changer (Filter)                Stopped (System)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
cirrus (Video)                  Running (System)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Cpqarray (SCSI miniport)        Stopped (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
cpqfws2e (SCSI miniport)        Stopped (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
dac960nt (SCSI miniport)        Running (Boot)
  C:\WINNTPCC\System32\drivers\dac960nt.sys
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
dce376nt (SCSI miniport)        Stopped (Disabled)

```

Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Delldsa (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Dell_DGX (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Disk (SCSI Class) Running (Boot)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Group Dependencies:
SCSI miniport
Diskperf (Filter) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
DptScsi (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
dte329x (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Intel(R) PRO NDIS Driver (NDIS) Running (Automatic)
C:\WINNTPCC\System32\drivers\el00bnt.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
et4000 (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
FAfaSa (SCSI miniport) Stopped (Disabled)
System32\DRIVERS\FAfaSa.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
FastAfa Stopped (Disabled)
C:\WINNTPCC\System32\DRIVERS\FastAfa.sys
Error Severity: Normal
Service Flags: File System Driver, Shared Process
Fastfat (Boot file system) Running (Disabled)
Error Severity: Normal
Service Flags: File System Driver, Shared Process
Fd16_700 (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Fd7000ex (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Fd8xx (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
flashpnt (SCSI miniport) Stopped (Disabled)
Error Severity: Normal

Service Flags: Kernel Driver, Shared Process
Floppy (Primary disk) Running (System)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Ftdisk (Filter) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
gamdrv (SCSI Class) Stopped (Manual)
C:\WINNTPCC\System32\drivers\gamdrv.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
i8042 Keyboard and PS/2 Mouse Port Driver (Keyboard Port) Running (System)
System32\DRIVERS\i8042prt.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Inport (Pointer Port) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Jazzg300 (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Jazzg364 (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Jzvxl484 (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
Keyboard Class Driver (Keyboard Class) Running (System)
System32\DRIVERS\kbdclass.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
KSecDD (Base) Running (System)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
macdisk (Filter) Running (Boot)
C:\WINNTPCC\System32\drivers\macdisk.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
mga (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
mga_mil (Video) Stopped (Disabled)
Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process
mitsumi (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
mkecr5xx (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Modem (Extended base) Stopped (Disabled)

Error Severity: Ignore Service Flags: Kernel Driver, Shared Process Mouse Class Driver (Pointer Class) Running (System) System32\DRIVERS\mouclass.sys Error Severity: Normal Service Flags: Kernel Driver, Shared Process Msfs (File system) Running (System) Error Severity: Normal Service Flags: File System Driver, Shared Process Mup (Network) Running (Manual) C:\WINNTPCC\System32\drivers\mup.sys Error Severity: Normal Service Flags: File System Driver, Shared Process Ncr53c9x (SCSI miniport) Stopped (Disabled) Error Severity: Normal Service Flags: Kernel Driver, Shared Process ncr77c22 (Video) Stopped (Disabled) Error Severity: Ignore Service Flags: Kernel Driver, Shared Process Ncr700 (SCSI miniport) Stopped (Disabled) Error Severity: Normal Service Flags: Kernel Driver, Shared Process Ncr710 (SCSI miniport) Stopped (Disabled) Error Severity: Normal Service Flags: Kernel Driver, Shared Process Microsoft NDIS System Driver (NDIS) Running (System) Error Severity: Normal Service Flags: Kernel Driver, Shared Process NetBIOS Interface (NetBIOSGroup) Stopped (Manual) C:\WINNTPCC\System32\drivers\netbios.sys Error Severity: Normal Service Flags: File System Driver, Shared Process Group Dependencies: TDI WINS Client (TCP/IP) (PNP_TDI) Running (Automatic) C:\WINNTPCC\System32\drivers\netbt.sys Error Severity: Normal Service Flags: Kernel Driver, Shared Process Service Dependencies: TcPIP NetDetect Stopped (Manual) C:\WINNTPCC\system32\drivers\netdtect.sys Error Severity: Normal Service Flags: Kernel Driver, Shared Process Npfs (File system) Running (System) Error Severity: Normal Service Flags: File System Driver, Shared Process Ntfs (File system) Running (Disabled) Error Severity: Normal Service Flags: File System Driver, Shared Process Null (Base) Running (System)	Error Severity: Normal Service Flags: Kernel Driver, Shared Process Oliscsi (SCSI miniport) Stopped (Disabled) Error Severity: Normal Service Flags: Kernel Driver, Shared Process Parallel (Extended base) Stopped (Manual) Error Severity: Ignore Service Flags: Kernel Driver, Shared Process Service Dependencies: Parport Group Dependencies: Parallel arbitrator Parport (Parallel arbitrator) Stopped (Manual) Error Severity: Ignore Service Flags: Kernel Driver, Shared Process ParVdm (Extended base) Stopped (Automatic) Error Severity: Ignore Service Flags: Kernel Driver, Shared Process Service Dependencies: Parport Group Dependencies: Parallel arbitrator PCIDump (PCI Configuration) Stopped (System) Error Severity: Ignore Service Flags: Kernel Driver, Shared Process Pcmcia (System Bus Extender) Stopped (Disabled) Error Severity: Normal Service Flags: Kernel Driver, Shared Process PnP ISA Enabler Driver (Base) Stopped (System) Error Severity: Ignore Service Flags: Kernel Driver, Shared Process psdisp (Video) Stopped (Disabled) Error Severity: Ignore Service Flags: Kernel Driver, Shared Process Ql10wnt (SCSI miniport) Stopped (Disabled) Error Severity: Normal Service Flags: Kernel Driver, Shared Process qv (Video) Stopped (Disabled) Error Severity: Ignore Service Flags: Kernel Driver, Shared Process Rdr (Network) Running (Manual) C:\WINNTPCC\System32\drivers\rdr.sys Error Severity: Normal Service Flags: File System Driver, Shared Process s3 (Video) Stopped (Disabled) Error Severity: Ignore Service Flags: Kernel Driver, Shared Process ScsiPrnt (Extended base) Stopped (Manual) Error Severity: Ignore Service Flags: Kernel Driver, Shared Process Group Dependencies:
--	--

```

SCSI miniport
Scsiscan (SCSI Class)           Running   (System)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
  Group Dependencies:
    SCSI miniport
Serial (Extended base)         Stopped   (Manual)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
Sermouse (Pointer Port)       Stopped   (Disabled)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
Sfloppy (Primary disk)        Stopped   (System)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
  Group Dependencies:
    SCSI miniport
Simbad (Filter)                Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
slcd32 (SCSI miniport)        Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Sparrow (SCSI miniport)       Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Spock (SCSI miniport)         Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Srv (Network)                  Running   (Manual)
  C:\WINNTPCC\System32\drivers\srv.sys
  Error Severity: Normal
  Service Flags: File System Driver, Shared Process
symc810 (SCSI miniport)       Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
symc8XX (SCSI miniport)       Running   (Boot)
  C:\WINNTPCC\system32\drivers\sycm8XX.sys
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
T128 (SCSI miniport)          Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
T13B (SCSI miniport)          Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
TCP/IP Service (PNP_TDI)      Running
(Automatic)
  C:\WINNTPCC\System32\drivers\tcpip.sys
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
tga (Video)                    Stopped   (Disabled)

```

```

  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
tmv1 (SCSI miniport)          Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Ultra124 (SCSI miniport)      Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Ultra14f (SCSI miniport)      Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
Ultra24f (SCSI miniport)      Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
update (Base)                 Stopped   (System)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
v7vram (Video)                Stopped   (Disabled)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
VgaSave (Video Save)          Stopped   (System)
  C:\WINNTPCC\System32\drivers\vga.sys
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
VgaStart (Video Init)         Stopped   (System)
  C:\WINNTPCC\System32\drivers\vga.sys
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
Wd33c93 (SCSI miniport)       Stopped   (Disabled)
  Error Severity: Normal
  Service Flags: Kernel Driver, Shared Process
wd90c24a (Video)              Stopped   (Disabled)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
wdvga (Video)                 Stopped   (Disabled)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
weitekp9 (Video)              Stopped   (Disabled)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process
Xga (Video)                   Stopped   (Disabled)
  Error Severity: Ignore
  Service Flags: Kernel Driver, Shared Process

```

IRQ and Port Report

```

-----
----
Devices                               Vector Level Affinity
-----
----

```

MPS 1.4 - APIC platform	8	8	0x0000000f	MPS 1.4 - APIC platform	193	193	0x0000000f
MPS 1.4 - APIC platform	0	0	0x0000000f	MPS 1.4 - APIC platform	225	225	0x0000000f
MPS 1.4 - APIC platform	1	1	0x0000000f	MPS 1.4 - APIC platform	253	253	0x0000000f
MPS 1.4 - APIC platform	2	2	0x0000000f	MPS 1.4 - APIC platform	254	254	0x0000000f
MPS 1.4 - APIC platform	3	3	0x0000000f	MPS 1.4 - APIC platform	255	255	0x0000000f
MPS 1.4 - APIC platform	4	4	0x0000000f	i8042prt	1	1	0xffffffff
MPS 1.4 - APIC platform	5	5	0x0000000f	i8042prt	12	12	0xffffffff
MPS 1.4 - APIC platform	6	6	0x0000000f	E100B	4	4	0x00000000
MPS 1.4 - APIC platform	7	7	0x0000000f	Floppy	6	6	0x00000000
MPS 1.4 - APIC platform	8	8	0x0000000f	dac960nt	32	32	0x00000000
MPS 1.4 - APIC platform	9	9	0x0000000f	dac960nt	32	32	0x00000000
MPS 1.4 - APIC platform	10	10	0x0000000f	dac960nt	32	32	0x00000000
MPS 1.4 - APIC platform	11	11	0x0000000f	dac960nt	32	32	0x00000000
MPS 1.4 - APIC platform	12	12	0x0000000f	dac960nt	32	32	0x00000000
MPS 1.4 - APIC platform	13	13	0x0000000f	dac960nt	32	32	0x00000000
MPS 1.4 - APIC platform	14	14	0x0000000f	dac960nt	32	32	0x00000000
MPS 1.4 - APIC platform	15	15	0x0000000f	dac960nt	32	32	0x00000000
MPS 1.4 - APIC platform	16	16	0x0000000f	dac960nt	32	32	0x00000000
MPS 1.4 - APIC platform	17	17	0x0000000f	symc8XX	8	8	0x00000000
MPS 1.4 - APIC platform	18	18	0x0000000f	symc8XX	12	12	0x00000000
MPS 1.4 - APIC platform	19	19	0x0000000f	-----			
MPS 1.4 - APIC platform	20	20	0x0000000f	----			
MPS 1.4 - APIC platform	21	21	0x0000000f	Devices		Physical Address	Length
MPS 1.4 - APIC platform	22	22	0x0000000f	-----			
MPS 1.4 - APIC platform	23	23	0x0000000f	----			
MPS 1.4 - APIC platform	24	24	0x0000000f	MPS 1.4 - APIC platform	0x00000000	0x0000000010	
MPS 1.4 - APIC platform	25	25	0x0000000f	MPS 1.4 - APIC platform	0x00000020	0x0000000002	
MPS 1.4 - APIC platform	26	26	0x0000000f	MPS 1.4 - APIC platform	0x00000040	0x0000000004	
MPS 1.4 - APIC platform	27	27	0x0000000f	MPS 1.4 - APIC platform	0x00000048	0x0000000004	
MPS 1.4 - APIC platform	28	28	0x0000000f	MPS 1.4 - APIC platform	0x00000061	0x0000000001	
MPS 1.4 - APIC platform	29	29	0x0000000f	MPS 1.4 - APIC platform	0x00000070	0x0000000002	
MPS 1.4 - APIC platform	30	30	0x0000000f	MPS 1.4 - APIC platform	0x00000080	0x0000000010	
MPS 1.4 - APIC platform	31	31	0x0000000f	MPS 1.4 - APIC platform	0x00000092	0x0000000001	
MPS 1.4 - APIC platform	32	32	0x0000000f	MPS 1.4 - APIC platform	0x000000a0	0x0000000002	
MPS 1.4 - APIC platform	33	33	0x0000000f	MPS 1.4 - APIC platform	0x000000c0	0x0000000010	
MPS 1.4 - APIC platform	34	34	0x0000000f	MPS 1.4 - APIC platform	0x000000f0	0x0000000010	
MPS 1.4 - APIC platform	35	35	0x0000000f	i8042prt	0x00000060	0x0000000001	
MPS 1.4 - APIC platform	36	36	0x0000000f	i8042prt	0x00000064	0x0000000001	
MPS 1.4 - APIC platform	37	37	0x0000000f	E100B	0x00003800	0x000000001e	
MPS 1.4 - APIC platform	38	38	0x0000000f	Floppy	0x000003f0	0x0000000006	
MPS 1.4 - APIC platform	39	39	0x0000000f	Floppy	0x000003f7	0x0000000001	
MPS 1.4 - APIC platform	40	40	0x0000000f	dac960nt	0x00004000	0x0000000080	
MPS 1.4 - APIC platform	41	41	0x0000000f	dac960nt	0x00005000	0x0000000080	
MPS 1.4 - APIC platform	42	42	0x0000000f	dac960nt	0x00006000	0x0000000080	
MPS 1.4 - APIC platform	43	43	0x0000000f	dac960nt	0x00007000	0x0000000080	
MPS 1.4 - APIC platform	44	44	0x0000000f	dac960nt	0x00008000	0x0000000080	
MPS 1.4 - APIC platform	45	45	0x0000000f	dac960nt	0x00009000	0x0000000080	
MPS 1.4 - APIC platform	46	46	0x0000000f	dac960nt	0x0000a000	0x0000000080	
MPS 1.4 - APIC platform	47	47	0x0000000f	dac960nt	0x0000b000	0x0000000080	
MPS 1.4 - APIC platform	61	61	0x0000000f	symc8XX	0x00003000	0x0000000100	
MPS 1.4 - APIC platform	65	65	0x0000000f	symc8XX	0x00003400	0x0000000100	
MPS 1.4 - APIC platform	80	80	0x0000000f	cirrus	0x000003b0	0x000000000c	
				cirrus	0x000003c0	0x0000000020	

DMA and Memory Report

```

-----
Devices                Channel   Port
-----
Floppy                  2       0
-----

```

```

-----
Devices                Physical Address  Length
-----
MPS 1.4 - APIC platform 0xfec00000 0x00000400
MPS 1.4 - APIC platform 0xfec00000 0x00000400
E100B                   0xe80ff000 0x0000001e
dac960nt                 0xe4310000 0x00000080
dac960nt                 0xea000000 0x02000000
dac960nt                 0xec110000 0x00000080
dac960nt                 0xee000000 0x02000000
dac960nt                 0xec210000 0x00000080
dac960nt                 0xf0000000 0x02000000
dac960nt                 0xec310000 0x00000080
dac960nt                 0xf2000000 0x02000000
dac960nt                 0xf4110000 0x00000080
dac960nt                 0xf6000000 0x02000000
dac960nt                 0xf4210000 0x00000080
dac960nt                 0xf8000000 0x02000000
dac960nt                 0xf4310000 0x00000080
dac960nt                 0xfa000000 0x02000000
dac960nt                 0xf4410000 0x00000080
dac960nt                 0xfc000000 0x02000000
symc8XX                  0xe4202000 0x00000100
symc8XX                  0xe4200000 0x00001000
symc8XX                  0xe4202400 0x00000100
symc8XX                  0xe4201000 0x00001000
cirrus                   0x000a0000 0x00020000
cirrus                   0xe6000000 0x02000000
-----

```

Environment Report

System Environment Variables

```

ComSpec=C:\WINNTPCC\system32\cmd.exe
NUMBER_OF_PROCESSORS=4

```

```

OS=Windows NT
Os2LibPath=C:\WINNTPCC\system32\os2\dll;
Path=C:\WINNTPCC\system32;C:\WINNTPCC;C:\Program
Files\Adaptec\AAC\System;D:\MSSQL7\BINN
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 7 Stepping 2,
GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=0702
windir=C:\WINNTPCC

```

Environment Variables for Current User

```

TEMP=C:\TEMP
TMP=C:\TEMP

```

Network Report

```

-----
Your Access Level: Admin & Local
Workgroup or Domain: CWIEN
Network Version: 4.0
LanRoot: CWIEN
Logged On Users: 1
Current User (1): Administrator
  Logon Domain: SPACELAB
  Logon Server: SPACELAB

```

Transport: NetBT_E100B1, 08-00-06-0D-68-2D, VC's: 0, Wan: Wan

```

Character Wait: 3,600
Collection Time: 250
Maximum Collection Count: 16
Keep Connection: 600
Maximum Commands: 5
Session Time Out: 45
Character Buffer Size: 512
Maximum Threads: 17
Lock Quota: 6,144
Lock Increment: 10
Maximum Locks: 500
Pipe Increment: 10
Maximum Pipes: 500
Cache Time Out: 40
Dormant File Limit: 45
Read Ahead Throughput: 4,294,967,295
Mailslot Buffers: 3
Server Announce Buffers: 20
Illegal Datagrams: 5

```

Datagram Reset Frequency: 60
 Log Election Packets: False
 Use Opportunistic Locking: True
 Use Unlock Behind: True
 Use Close Behind: True
 Buffer Pipes: True
 Use Lock, Read, Unlock: True
 Use NT Caching: True
 Use Raw Read: True
 Use Raw Write: True
 Use Write Raw Data: True
 Use Encryption: True
 Buffer Deny Write Files: True
 Buffer Read Only Files: True
 Force Core Creation: True
 512 Byte Max Transfer: False
 Bytes Received: 4,087
 SMB's Received: 44
 Paged Read Bytes Requested: 0
 Non Paged Read Bytes Requested: 0
 Cache Read Bytes Requested: 0
 Network Read Bytes Requested: 0
 Bytes Transmitted: 5,072
 SMB's Transmitted: 44
 Paged Read Bytes Requested: 0
 Non Paged Read Bytes Requested: 0
 Cache Read Bytes Requested: 0
 Network Read Bytes Requested: 0
 Initially Failed Operations: 0
 Failed Completion Operations: 0
 Read Operations: 0
 Random Read Operations: 0
 Read SMB's: 0
 Large Read SMB's: 0
 Small Read SMB's: 0
 Write Operations: 0
 Random Write Operations: 0
 Write SMB's: 0
 Large Write SMB's: 0
 Small Write SMB's: 0
 Raw Reads Denied: 0
 Raw Writes Denied: 0
 Network Errors: 0
 Sessions: 3
 Failed Sessions: 0
 Reconnects: 0
 Core Connects: 0
 LM 2.0 Connects: 0
 LM 2.x Connects: 0
 Windows NT Connects: 3
 Server Disconnects: 0
 Hung Sessions: 0

Use Count: 0
 Failed Use Count: 0
 Current Commands: 0
 Server File Opens: 0
 Server Device Opens: 0
 Server Jobs Queued: 0
 Server Session Opens: 0
 Server Sessions Timed Out: 0
 Server Sessions Errored Out: 0
 Server Password Errors: 0
 Server Permission Errors: 0
 Server System Errors: 0
 Server Bytes Sent: 0
 Server Bytes Received: 0
 Server Average Response Time: 0
 Server Request Buffers Needed: 0
 Server Big Buffers Needed: 0

```

*****
****
*           MYLEX Disk Array Controller - Configuration Utility
*
*           Version 4.78-15
*
*****
****

CONFIGURATION INFORMATION OF :
=====

3 Channel - 15 Target DAC1164P #1 Firmware version
5.07-0-1

Auto Rebuild Management      : Disabled
Storage Works Fault Management : Disabled
Rebuild/Add Capacity Rate    : 50
Stripe Size                  : 64K
Cache Segment Size           : 8K

SCSI Transfer Parameters
-----

Data Transfer Rate for channel 0: 20 MHz
Data Bus Width for channel 0   : 16 Bit
Command Tags for channel 0     : Enabled

Data Transfer Rate for channel 1: 20 MHz
  
```

Data Bus Width for channel 1 : 16 Bit
Command Tags for channel 1 : Enabled

Data Transfer Rate for channel 2: 40 MHz
Data Bus Width for channel 2 : 16 Bit
Command Tags for channel 2 : Enabled

Startup Parameters

Spin Up Option : Automatic
Number of devices per spin up : 2
Length of delay : 6 seconds
Sequence delay : 6 seconds

PHYSICAL PACK INFORMATION :
=====

Number of Packs = 4

Pack 0 : [0:0] [1:0]
Pack 1 : [0:1] [1:1]
Pack 2 : [0:2] [1:2]
Pack 3 : [0:3] [1:3]

SYSTEM DRIVE INFORMATION :
=====

Number of System Drives = 1

Sys Drv#	Phy.	Size	Raid Level	Eff. Size	Write Policy	State
0	138928 MB	1		69464 MB	Write Thru	Online

Device Information

Chnl/Targ	Vendor	Model	Version	Size	State
0-0	SEAGATE	ST118202LC	7503	17366 MB	Online
0-1	SEAGATE	ST118202LC	7503	17366 MB	Online
0-2	SEAGATE	ST118202LC	7503	17366 MB	Online
0-3	SEAGATE	ST118202LC	7503	17366 MB	Online
1-0	SEAGATE	ST118202LC	7503	17366 MB	Online
1-1	SEAGATE	ST118202LC	7503	17366 MB	Online
1-2	SEAGATE	ST118202LC	7503	17366 MB	Online
1-3	SEAGATE	ST118202LC	7503	17366 MB	Online

* MYLEX Disk Array Controller - Configuration Utility
*

* Version 4.78-15
*

CONFIGURATION INFORMATION OF :
=====

3 Channel - 15 Target DAC1164P #2 Firmware version
5.07-0-1

Auto Rebuild Management : Disabled
Storage Works Fault Management : Disabled
Rebuild/Add Capacity Rate : 50
Stripe Size : 64K
Cache Segment Size : 8K

SCSI Transfer Parameters

Data Transfer Rate for channel 0: 40 MHz
Data Bus Width for channel 0 : 16 Bit
Command Tags for channel 0 : Enabled

Data Transfer Rate for channel 1: 40 MHz
Data Bus Width for channel 1 : 16 Bit
Command Tags for channel 1 : Enabled

Data Transfer Rate for channel 2: 40 MHz
Data Bus Width for channel 2 : 16 Bit
Command Tags for channel 2 : Enabled

Startup Parameters

Spin Up Option : Automatic
Number of devices per spin up : 2
Length of delay : 6 seconds
Sequence delay : 6 seconds

PHYSICAL PACK INFORMATION :
=====

Number of Packs = 4

Pack 0 : [0:0] [1:0] [2:0] [0:1] [1:1] [2:1] [0:2]
[1:2]
Pack 1 : [2:2] [0:3] [1:3] [2:3] [0:4] [1:4] [2:4]
[0:5]
Pack 2 : [1:5] [2:5] [0:10] [1:10] [2:10] [0:11]
[1:11] [2:11]

Pack 3 : [0:12] [1:12] [2:12] [0:13] [1:13] [2:13]
[0:14] [1:14]

SYSTEM DRIVE INFORMATION :
=====

Number of System Drives = 1

Sys Drv#	Phy. Size	Raid Level	Eff. Size	Write Policy	State
0	277856 MB	0	277856 MB	Write Thru	

Online

Device Information

Chnl/Targ	Vendor	Model	Version	Size	State
0-0	SEAGATE	ST39102LC	7503	8683 MB	Online
0-1	SEAGATE	ST39102LC	7503	8683 MB	Online
0-2	SEAGATE	ST39102LC	7503	8683 MB	Online
0-3	SEAGATE	ST39102LC	7503	8683 MB	Online
0-4	SEAGATE	ST39102LC	7503	8683 MB	Online
0-5	SEAGATE	ST39102LC	7503	8683 MB	Online
0-10	SEAGATE	ST39102LC	7503	8683 MB	Online
0-11	SEAGATE	ST39102LC	7503	8683 MB	Online
0-12	SEAGATE	ST39102LC	7503	8683 MB	Online
0-13	SEAGATE	ST39102LC	7503	8683 MB	Online
0-14	SEAGATE	ST39102LC	7503	8683 MB	Online
1-0	SEAGATE	ST39102LC	7503	8683 MB	Online
1-1	SEAGATE	ST39102LC	7503	8683 MB	Online
1-2	SEAGATE	ST39102LC	7503	8683 MB	Online
1-3	SEAGATE	ST39102LC	7503	8683 MB	Online
1-4	SEAGATE	ST39102LC	7503	8683 MB	Online
1-5	SEAGATE	ST39102LC	7503	8683 MB	Online
1-10	SEAGATE	ST39102LC	7503	8683 MB	Online
1-11	SEAGATE	ST39102LC	7503	8683 MB	Online
1-12	SEAGATE	ST39102LC	7503	8683 MB	Online
1-13	SEAGATE	ST39102LC	7503	8683 MB	Online
1-14	SEAGATE	ST39102LC	7503	8683 MB	Online
2-0	SEAGATE	ST39102LC	7503	8683 MB	Online
2-1	SEAGATE	ST39102LC	7503	8683 MB	Online
2-2	SEAGATE	ST39102LC	7503	8683 MB	Online
2-3	SEAGATE	ST39102LC	7503	8683 MB	Online
2-4	SEAGATE	ST39102LC	7503	8683 MB	Online
2-5	SEAGATE	ST39102LC	7503	8683 MB	Online
2-10	SEAGATE	ST39102LC	7503	8683 MB	Online
2-11	SEAGATE	ST39102LC	7503	8683 MB	Online
2-12	SEAGATE	ST39102LC	7503	8683 MB	Online
2-13	SEAGATE	ST39102LC	7503	8683 MB	Online

* MYLEX Disk Array Controller - Configuration Utility

*

* Version 4.78-15

*

CONFIGURATION INFORMATION OF :
=====

3 Channel - 15 Target DAC1164P #3 Firmware version
5.07-0-1

Auto Rebuild Management : Disabled
Storage Works Fault Management : Disabled
Rebuild/Add Capacity Rate : 50
Stripe Size : 64K
Cache Segment Size : 8K

SCSI Transfer Parameters

Data Transfer Rate for channel 0: 40 MHz
Data Bus Width for channel 0 : 16 Bit
Command Tags for channel 0 : Enabled

Data Transfer Rate for channel 1: 40 MHz
Data Bus Width for channel 1 : 16 Bit
Command Tags for channel 1 : Enabled

Data Transfer Rate for channel 2: 40 MHz
Data Bus Width for channel 2 : 16 Bit
Command Tags for channel 2 : Enabled

Startup Parameters

Spin Up Option : Automatic
Number of devices per spin up : 2
Length of delay : 6 seconds
Sequence delay : 6 seconds

PHYSICAL PACK INFORMATION :
=====

Number of Packs = 4

Pack 0 : [0:0] [1:0] [2:0] [0:1] [1:1] [2:1] [0:2]

[1:2]

```

Pack 1 : [2:2] [0:3] [1:3] [2:3] [0:4] [1:4] [2:4]          2-13  SEAGATE  ST39102LC          7503    8683 MB  Online
[0:5]
Pack 2 : [1:5] [2:5] [0:10] [1:10] [2:10] [0:11]
[1:11] [2:11]
Pack 3 : [0:12] [1:12] [2:12] [0:13] [1:13] [2:13]
[0:14] [1:14]

```

SYSTEM DRIVE INFORMATION :
=====

Number of System Drives = 1

Sys Drv#	Phy. Size	Raid Level	Eff. Size	Write Policy	State
0	277856 MB	0	277856 MB	Write Thru	Online

Device Information

Chnl/Targ	Vendor	Model	Version	Size	State
0-0	SEAGATE	ST39102LC	7503	8683 MB	Online
0-1	SEAGATE	ST39102LC	7503	8683 MB	Online
0-2	SEAGATE	ST39102LC	7503	8683 MB	Online
0-3	SEAGATE	ST39102LC	7503	8683 MB	Online
0-4	SEAGATE	ST39102LC	7503	8683 MB	Online
0-5	SEAGATE	ST39102LC	7503	8683 MB	Online
0-10	SEAGATE	ST39102LC	7503	8683 MB	Online
0-11	SEAGATE	ST39102LC	7503	8683 MB	Online
0-12	SEAGATE	ST39102LC	7503	8683 MB	Online
0-13	SEAGATE	ST39102LC	7503	8683 MB	Online
0-14	SEAGATE	ST39102LC	7503	8683 MB	Online
1-0	SEAGATE	ST39102LC	7503	8683 MB	Online
1-1	SEAGATE	ST39102LC	7503	8683 MB	Online
1-2	SEAGATE	ST39102LC	7503	8683 MB	Online
1-3	SEAGATE	ST39102LC	7503	8683 MB	Online
1-4	SEAGATE	ST39102LC	7503	8683 MB	Online
1-5	SEAGATE	ST39102LC	7503	8683 MB	Online
1-10	SEAGATE	ST39102LC	7503	8683 MB	Online
1-11	SEAGATE	ST39102LC	7503	8683 MB	Online
1-12	SEAGATE	ST39102LC	7503	8683 MB	Online
1-13	SEAGATE	ST39102LC	7503	8683 MB	Online
1-14	SEAGATE	ST39102LC	7503	8683 MB	Online
2-0	SEAGATE	ST39102LC	7503	8683 MB	Online
2-1	SEAGATE	ST39102LC	7503	8683 MB	Online
2-2	SEAGATE	ST39102LC	7503	8683 MB	Online
2-3	SEAGATE	ST39102LC	7503	8683 MB	Online
2-4	SEAGATE	ST39102LC	7503	8683 MB	Online
2-5	SEAGATE	ST39102LC	7503	8683 MB	Online
2-10	SEAGATE	ST39102LC	7503	8683 MB	Online
2-11	SEAGATE	ST39102LC	7503	8683 MB	Online
2-12	SEAGATE	ST39102LC	7503	8683 MB	Online

```

*****
****
*           MYLEX Disk Array Controller - Configuration Utility
*
*           Version 4.78-15
*
*****
****

```

CONFIGURATION INFORMATION OF :
=====

```

3 Channel - 15 Target  DAC1164P  #4  Firmware version
5.07-0-1

Auto Rebuild Management      : Disabled
Storage Works Fault Management : Disabled
Rebuild/Add Capacity Rate    : 50
Stripe Size                  : 64K
Cache Segment Size           : 8K

SCSI Transfer Parameters
-----

Data Transfer Rate for channel 0: 40 MHz
Data Bus Width for channel 0   : 16 Bit
Command Tags for channel 0     : Enabled

Data Transfer Rate for channel 1: 40 MHz
Data Bus Width for channel 1   : 16 Bit
Command Tags for channel 1     : Enabled

Data Transfer Rate for channel 2: 40 MHz
Data Bus Width for channel 2   : 16 Bit
Command Tags for channel 2     : Enabled

Startup Parameters
-----

Spin Up Option                : Automatic
Number of devices per spin up : 2
Length of delay               : 6 seconds
Sequence delay                : 6 seconds

```

PHYSICAL PACK INFORMATION :
=====

Number of Packs = 4

```

Pack 0 : [0:0] [1:0] [2:0] [0:1] [1:1] [2:1] [0:2]
[1:2]
Pack 1 : [2:2] [0:3] [1:3] [2:3] [0:4] [1:4] [2:4]
[0:5]
Pack 2 : [1:5] [2:5] [0:10] [1:10] [2:10] [0:11]
[1:11] [2:11]
Pack 3 : [0:12] [1:12] [2:12] [0:13] [1:13] [2:13]
[0:14] [1:14]

```

```

2-10 SEAGATE ST39102LC 7503 8683 MB Online
2-11 SEAGATE ST39102LC 7503 8683 MB Online
2-12 SEAGATE ST39102LC 7503 8683 MB Online
2-13 SEAGATE ST39102LC 7503 8683 MB Online

```

SYSTEM DRIVE INFORMATION :
=====

Number of System Drives = 1

Sys Drv#	Phy. Size	Raid Level	Eff. Size	Write Policy	State
0	277856 MB	0	277856 MB	Write Thru	Online

Device Information

Chnl/Targ	Vendor	Model	Version	Size	State
0-0	SEAGATE	ST39102LC	7503	8683 MB	Online
0-1	SEAGATE	ST39102LC	7503	8683 MB	Online
0-2	SEAGATE	ST39102LC	7503	8683 MB	Online
0-3	SEAGATE	ST39102LC	7503	8683 MB	Online
0-4	SEAGATE	ST39102LC	7503	8683 MB	Online
0-5	SEAGATE	ST39102LC	7503	8683 MB	Online
0-10	SEAGATE	ST39102LC	7503	8683 MB	Online
0-11	SEAGATE	ST39102LC	7503	8683 MB	Online
0-12	SEAGATE	ST39102LC	7503	8683 MB	Online
0-13	SEAGATE	ST39102LC	7503	8683 MB	Online
0-14	SEAGATE	ST39102LC	7503	8683 MB	Online
1-0	SEAGATE	ST39102LC	7503	8683 MB	Online
1-1	SEAGATE	ST39102LC	7503	8683 MB	Online
1-2	SEAGATE	ST39102LC	7503	8683 MB	Online
1-3	SEAGATE	ST39102LC	7503	8683 MB	Online
1-4	SEAGATE	ST39102LC	7503	8683 MB	Online
1-5	SEAGATE	ST39102LC	7503	8683 MB	Online
1-10	SEAGATE	ST39102LC	7503	8683 MB	Online
1-11	SEAGATE	ST39102LC	7503	8683 MB	Online
1-12	SEAGATE	ST39102LC	7503	8683 MB	Online
1-13	SEAGATE	ST39102LC	7503	8683 MB	Online
1-14	SEAGATE	ST39102LC	7503	8683 MB	Online
2-0	SEAGATE	ST39102LC	7503	8683 MB	Online
2-1	SEAGATE	ST39102LC	7503	8683 MB	Online
2-2	SEAGATE	ST39102LC	7503	8683 MB	Online
2-3	SEAGATE	ST39102LC	7503	8683 MB	Online
2-4	SEAGATE	ST39102LC	7503	8683 MB	Online
2-5	SEAGATE	ST39102LC	7503	8683 MB	Online

* MYLEX Disk Array Controller - Configuration Utility

*

Version 4.78-15

*

CONFIGURATION INFORMATION OF :
=====

3 Channel - 15 Target DAC1164P #5 Firmware version
5.07-0-1

Auto Rebuild Management : Disabled
Storage Works Fault Management : Disabled
Rebuild/Add Capacity Rate : 50
Stripe Size : 64K
Cache Segment Size : 8K

SCSI Transfer Parameters

Data Transfer Rate for channel 0: 40 MHz
Data Bus Width for channel 0 : 16 Bit
Command Tags for channel 0 : Enabled

Data Transfer Rate for channel 1: 40 MHz
Data Bus Width for channel 1 : 16 Bit
Command Tags for channel 1 : Enabled

Data Transfer Rate for channel 2: 40 MHz
Data Bus Width for channel 2 : 16 Bit
Command Tags for channel 2 : Enabled

Startup Parameters

Spin Up Option : Automatic
Number of devices per spin up : 2
Length of delay : 6 seconds
Sequence delay : 6 seconds

PHYSICAL PACK INFORMATION :

```

=====
Number of Packs = 4

Pack 0 : [0:0] [1:0] [2:0] [0:1] [1:1] [2:1] [0:2]
[1:2]
Pack 1 : [2:2] [0:3] [1:3] [2:3] [0:4] [1:4] [2:4]
[0:5]
Pack 2 : [1:5] [2:5] [0:10] [1:10] [2:10] [0:11]
[1:11] [2:11]
Pack 3 : [0:12] [1:12] [2:12] [0:13] [1:13] [2:13]
[0:14] [1:14]

```

SYSTEM DRIVE INFORMATION :
=====

Number of System Drives = 1

Sys Drv#	Phy. Size	Raid Level	Eff. Size	Write Policy	State
0	277856 MB	0	277856 MB	Write Thru	Online

Device Information

Chnl/Targ	Vendor	Model	Version	Size	State
0-0	SEAGATE	ST39102LC	7503	8683 MB	Online
0-1	SEAGATE	ST39102LC	7503	8683 MB	Online
0-2	SEAGATE	ST39102LC	7503	8683 MB	Online
0-3	SEAGATE	ST39102LC	7503	8683 MB	Online
0-4	SEAGATE	ST39102LC	7503	8683 MB	Online
0-5	SEAGATE	ST39102LC	7503	8683 MB	Online
0-10	SEAGATE	ST39102LC	7503	8683 MB	Online
0-11	SEAGATE	ST39102LC	7503	8683 MB	Online
0-12	SEAGATE	ST39102LC	7503	8683 MB	Online
0-13	SEAGATE	ST39102LC	7503	8683 MB	Online
0-14	SEAGATE	ST39102LC	7503	8683 MB	Online
1-0	SEAGATE	ST39102LC	7503	8683 MB	Online
1-1	SEAGATE	ST39102LC	7503	8683 MB	Online
1-2	SEAGATE	ST39102LC	7503	8683 MB	Online
1-3	SEAGATE	ST39102LC	7503	8683 MB	Online
1-4	SEAGATE	ST39102LC	7503	8683 MB	Online
1-5	SEAGATE	ST39102LC	7503	8683 MB	Online
1-10	SEAGATE	ST39102LC	7503	8683 MB	Online
1-11	SEAGATE	ST39102LC	7503	8683 MB	Online
1-12	SEAGATE	ST39102LC	7503	8683 MB	Online
1-13	SEAGATE	ST39102LC	7503	8683 MB	Online
1-14	SEAGATE	ST39102LC	7503	8683 MB	Online
2-0	SEAGATE	ST39102LC	7503	8683 MB	Online
2-1	SEAGATE	ST39102LC	7503	8683 MB	Online
2-2	SEAGATE	ST39102LC	7503	8683 MB	Online

2-3	SEAGATE	ST39102LC	7503	8683 MB	Online
2-4	SEAGATE	ST39102LC	7503	8683 MB	Online
2-5	SEAGATE	ST39102LC	7503	8683 MB	Online
2-10	SEAGATE	ST39102LC	7503	8683 MB	Online
2-11	SEAGATE	ST39102LC	7503	8683 MB	Online
2-12	SEAGATE	ST39102LC	7503	8683 MB	Online
2-13	SEAGATE	ST39102LC	7503	8683 MB	Online

```

*****
****
*           MYLEX Disk Array Controller - Configuration Utility
*
*           Version 4.78-15
*
*****

```

CONFIGURATION INFORMATION OF :
=====

3 Channel - 15 Target DAC1164P #6 Firmware version
5.07-0-1

```

Auto Rebuild Management      : Disabled
Storage Works Fault Management : Disabled
Rebuild/Add Capacity Rate    : 50
Stripe Size                  : 64K
Cache Segment Size           : 8K

```

SCSI Transfer Parameters

```

Data Transfer Rate for channel 0: 40 MHz
Data Bus Width for channel 0      : 16 Bit
Command Tags for channel 0       : Enabled

Data Transfer Rate for channel 1: 40 MHz
Data Bus Width for channel 1      : 16 Bit
Command Tags for channel 1       : Enabled

Data Transfer Rate for channel 2: 40 MHz
Data Bus Width for channel 2      : 16 Bit
Command Tags for channel 2       : Enabled

```

Startup Parameters

```

Spin Up Option                : Automatic
Number of devices per spin up : 2
Length of delay                : 6 seconds

```

Sequence delay : 6 seconds

PHYSICAL PACK INFORMATION :
=====

Number of Packs = 4

Pack 0 : [0:0] [1:0] [2:0] [0:1] [1:1] [2:1] [0:2]
[1:2]
Pack 1 : [2:2] [0:3] [1:3] [2:3] [0:4] [1:4] [2:4]
[0:5]
Pack 2 : [1:5] [2:5] [0:10] [1:10] [2:10] [0:11]
[1:11] [2:11]
Pack 3 : [0:12] [1:12] [2:12] [0:13] [1:13] [2:13]
[0:14] [1:14]

SYSTEM DRIVE INFORMATION :
=====

Number of System Drives = 1

Sys Drv# Phy. Size Raid Level Eff. Size Write Policy State
=====

Online

Device Information

Chnl/Targ	Vendor	Model	Version	Size	State
0-0	SEAGATE	ST39102LC	7503	8683 MB	Online
0-1	SEAGATE	ST39102LC	7503	8683 MB	Online
0-2	SEAGATE	ST39102LC	7503	8683 MB	Online
0-3	SEAGATE	ST39102LC	7503	8683 MB	Online
0-4	SEAGATE	ST39102LC	7503	8683 MB	Online
0-5	SEAGATE	ST39102LC	7503	8683 MB	Online
0-10	SEAGATE	ST39102LC	7503	8683 MB	Online
0-11	SEAGATE	ST39102LC	7503	8683 MB	Online
0-12	SEAGATE	ST39102LC	7503	8683 MB	Online
0-13	SEAGATE	ST39102LC	7503	8683 MB	Online
0-14	SEAGATE	ST39102LC	7503	8683 MB	Online
1-0	SEAGATE	ST39102LC	7503	8683 MB	Online
1-1	SEAGATE	ST39102LC	7503	8683 MB	Online
1-2	SEAGATE	ST39102LC	7503	8683 MB	Online
1-3	SEAGATE	ST39102LC	7503	8683 MB	Online
1-4	SEAGATE	ST39102LC	7503	8683 MB	Online
1-5	SEAGATE	ST39102LC	7503	8683 MB	Online
1-10	SEAGATE	ST39102LC	7503	8683 MB	Online
1-11	SEAGATE	ST39102LC	7503	8683 MB	Online
1-12	SEAGATE	ST39102LC	7503	8683 MB	Online
1-13	SEAGATE	ST39102LC	7503	8683 MB	Online
1-14	SEAGATE	ST39102LC	7503	8683 MB	Online

2-0	SEAGATE	ST39102LC	7503	8683 MB	Online
2-1	SEAGATE	ST39102LC	7503	8683 MB	Online
2-2	SEAGATE	ST39102LC	7503	8683 MB	Online
2-3	SEAGATE	ST39102LC	7503	8683 MB	Online
2-4	SEAGATE	ST39102LC	7503	8683 MB	Online
2-5	SEAGATE	ST39102LC	7503	8683 MB	Online
2-10	SEAGATE	ST39102LC	7503	8683 MB	Online
2-11	SEAGATE	ST39102LC	7503	8683 MB	Online
2-12	SEAGATE	ST39102LC	7503	8683 MB	Online
2-13	SEAGATE	ST39102LC	7503	8683 MB	Online

* MYLEX Disk Array Controller - Configuration Utility
*
* Version 4.78-15
*

CONFIGURATION INFORMATION OF :
=====

3 Channel - 15 Target DAC1164P #7 Firmware version
5.07-0-1

Auto Rebuild Management : Disabled
Storage Works Fault Management : Disabled
Rebuild/Add Capacity Rate : 50
Stripe Size : 64K
Cache Segment Size : 8K

SCSI Transfer Parameters

Data Transfer Rate for channel 0: 40 MHz
Data Bus Width for channel 0 : 16 Bit
Command Tags for channel 0 : Enabled

Data Transfer Rate for channel 1: 40 MHz
Data Bus Width for channel 1 : 16 Bit
Command Tags for channel 1 : Enabled

Data Transfer Rate for channel 2: 40 MHz
Data Bus Width for channel 2 : 16 Bit
Command Tags for channel 2 : Enabled

Startup Parameters

Spin Up Option : Automatic
 Number of devices per spin up : 2
 Length of delay : 6 seconds
 Sequence delay : 6 seconds

PHYSICAL PACK INFORMATION :
 =====

Number of Packs = 4

Pack 0 : [0:0] [1:0] [2:0] [0:1] [1:1] [2:1] [0:2]
 [1:2]
 Pack 1 : [2:2] [0:3] [1:3] [2:3] [0:4] [1:4] [2:4]
 [0:5]
 Pack 2 : [1:5] [2:5] [0:10] [1:10] [2:10] [0:11]
 [1:11] [2:11]
 Pack 3 : [0:12] [1:12] [2:12] [0:13] [1:13] [2:13]
 [0:14] [1:14]

SYSTEM DRIVE INFORMATION :
 =====

Number of System Drives = 1

Sys Drv#	Phy. Size	Raid Level	Eff. Size	Write Policy	State
0	277856 MB	0	277856 MB	Write Thru	Online

Device Information

Chnl/Targ	Vendor	Model	Version	Size	State
0-0	SEAGATE	ST39102LC	7503	8683 MB	Online
0-1	SEAGATE	ST39102LC	7503	8683 MB	Online
0-2	SEAGATE	ST39102LC	7503	8683 MB	Online
0-3	SEAGATE	ST39102LC	7503	8683 MB	Online
0-4	SEAGATE	ST39102LC	7503	8683 MB	Online
0-5	SEAGATE	ST39102LC	7503	8683 MB	Online
0-10	SEAGATE	ST39102LC	7503	8683 MB	Online
0-11	SEAGATE	ST39102LC	7503	8683 MB	Online
0-12	SEAGATE	ST39102LC	7503	8683 MB	Online
0-13	SEAGATE	ST39102LC	7503	8683 MB	Online
0-14	SEAGATE	ST39102LC	7503	8683 MB	Online
1-0	SEAGATE	ST39102LC	7503	8683 MB	Online
1-1	SEAGATE	ST39102LC	7503	8683 MB	Online
1-2	SEAGATE	ST39102LC	7503	8683 MB	Online
1-3	SEAGATE	ST39102LC	7503	8683 MB	Online
1-4	SEAGATE	ST39102LC	7503	8683 MB	Online
1-5	SEAGATE	ST39102LC	7503	8683 MB	Online
1-10	SEAGATE	ST39102LC	7503	8683 MB	Online
1-11	SEAGATE	ST39102LC	7503	8683 MB	Online

1-12	SEAGATE	ST39102LC	7503	8683 MB	Online
1-13	SEAGATE	ST39102LC	7503	8683 MB	Online
1-14	SEAGATE	ST39102LC	7503	8683 MB	Online
2-0	SEAGATE	ST39102LC	7503	8683 MB	Online
2-1	SEAGATE	ST39102LC	7503	8683 MB	Online
2-2	SEAGATE	ST39102LC	7503	8683 MB	Online
2-3	SEAGATE	ST39102LC	7503	8683 MB	Online
2-4	SEAGATE	ST39102LC	7503	8683 MB	Online
2-5	SEAGATE	ST39102LC	7503	8683 MB	Online
2-10	SEAGATE	ST39102LC	7503	8683 MB	Online
2-11	SEAGATE	ST39102LC	7503	8683 MB	Online
2-12	SEAGATE	ST39102LC	7503	8683 MB	Online
2-13	SEAGATE	ST39102LC	7503	8683 MB	Online

 * MYLEX Disk Array Controller - Configuration Utility
 *
 * Version 4.78-15
 *

CONFIGURATION INFORMATION OF :
 =====

3 Channel - 15 Target DAC1164P #8 Firmware version
 5.07-0-1

Auto Rebuild Management : Disabled
 Storage Works Fault Management : Disabled
 Rebuild/Add Capacity Rate : 50
 Stripe Size : 64K
 Cache Segment Size : 8K

SCSI Transfer Parameters

Data Transfer Rate for channel 0: 20 MHz
 Data Bus Width for channel 0 : 16 Bit
 Command Tags for channel 0 : Enabled

Data Transfer Rate for channel 1: 20 MHz
 Data Bus Width for channel 1 : 16 Bit
 Command Tags for channel 1 : Enabled

Data Transfer Rate for channel 2: 20 MHz
 Data Bus Width for channel 2 : 16 Bit
 Command Tags for channel 2 : Enabled

Startup Parameters

Spin Up Option : Automatic
Number of devices per spin up : 2
Length of delay : 6 seconds
Sequence delay : 6 seconds

PHYSICAL PACK INFORMATION :
=====

Number of Packs = 6

Pack 0 : [0:0] [0:1] [0:2] [0:3]
Pack 1 : [0:4] [0:5] [0:10] [0:11]
Pack 2 : [0:12] [0:13] [0:14] [0:15]
Pack 3 : [1:0] [1:1] [1:2] [1:3]
Pack 4 : [1:4] [1:5] [1:10] [1:11]
Pack 5 : [1:12] [1:13] [1:14] [1:15]

SYSTEM DRIVE INFORMATION :
=====

Number of System Drives = 2

Sys Drv#	Phy. Size	Raid Level	Eff. Size	Write Policy	State
0	104196 MB	0	104196 MB	Write Thru	Online
1	104196 MB	0	104196 MB	Write Thru	Online

Device Information

Chnl/Targ	Vendor	Model	Version	Size	State
0-0	SEAGATE	ST19171W	B501	8683 MB	Online
0-1	SEAGATE	ST19171W	B501	8683 MB	Online
0-2	SEAGATE	ST19171W	B501	8683 MB	Online
0-3	SEAGATE	ST19171W	B501	8683 MB	Online
0-4	SEAGATE	ST19171W	B501	8683 MB	Online
0-5	SEAGATE	ST19171W	B501	8683 MB	Online
0-10	SEAGATE	ST19171W	B501	8683 MB	Online
0-11	SEAGATE	ST19171W	B501	8683 MB	Online
0-12	SEAGATE	ST19101WC	7502	8683 MB	Online
0-13	SEAGATE	ST19101WC	7502	8683 MB	Online
0-14	SEAGATE	ST19101WC	7502	8683 MB	Online
0-15	SEAGATE	ST19101WC	7502	8683 MB	Online
1-0	SEAGATE	ST19171W	B501	8683 MB	Online
1-1	SEAGATE	ST19171W	B501	8683 MB	Online
1-2	SEAGATE	ST19171W	B501	8683 MB	Online
1-3	SEAGATE	ST19171W	B501	8683 MB	Online

1-4	SEAGATE	ST19171W	B501	8683 MB	Online
1-5	SEAGATE	ST19171W	B501	8683 MB	Online
1-10	SEAGATE	ST19171W	B501	8683 MB	Online
1-11	SEAGATE	ST19171W	B501	8683 MB	Online
1-12	SEAGATE	ST19101WC	7502	8683 MB	Online
1-13	SEAGATE	ST19101WC	7502	8683 MB	Online
1-14	SEAGATE	ST19101WC	7502	8683 MB	Online
1-15	SEAGATE	ST19101WC	7502	8683 MB	Online

Disk Array Controller firmware and driver for databases with high workload are available at <http://www.mylex.com>.

This section discloses hardware information of the Primergy 460 client system.

***** NT diagnostic *****

Microsoft Diagnostics Report For \\WEISS

OS Version Report

Microsoft (R) Windows NT (TM) Server
Version 4.0 (Build 1381: Service Pack 4) x86 Multiprocessor Free
Registered Owner: tpcc, Siemens
Product Number: 50382-040-1111111-54972

System Report

System: AT/AT COMPATIBLE
Hardware Abstraction Layer: MPS 1.4 - APIC platform
BIOS Date: 09/04/98
BIOS Version: PhoenixBIOS Version 4.06 Rev. 1

Processor list:

0: x86 Family 6 Model 5 Stepping 2 GenuineIntel ~446 Mhz
1: x86 Family 6 Model 5 Stepping 2 GenuineIntel ~446 Mhz

Video Display Report

```

-----
----
BIOS Date: 05/21/97
BIOS Version: CL-GD5446 PCI VGA BIOS Version 1.33
              Rel. 1.00

```

```

Adapter:
  Setting: 800 x 600 x 65536
           75 Hz
  Type: cirrus compatible display adapter
  String: Cirrus Logic Compatible
  Memory: 2 MB
  Chip Type: Cirrus Logic 5446
  DAC Type: Integrated RAMDAC
Driver:
  Vendor: Microsoft Corporation
  File(s): cirrus.sys, vga.dll, cirrus.dll, vga256.dll,
vga64K.dll
  Version: 4.00, 4.0.0

```

Drives Report

```

-----
----
C:\ (Local - NTFS) Total: 2,049,862 KB, Free: 674,540 KB
  Serial Number: 6CD7 - ED86
  Bytes per cluster: 512
  Sectors per cluster: 1
  Filename length: 255
D:\ (Local - NTFS) DISK_D Total: 2,203,708 KB, Free: 1,584,300 KB
  Serial Number: 2C85 - 6841
  Bytes per cluster: 512
  Sectors per cluster: 8
  Filename length: 255

```

Memory Report

```

-----
----
Handles: 16,976
Threads: 127
Processes: 17

```

Physical Memory (K)

```

Total: 392,624
Available: 313,932
File Cache: 21,228

```

Kernel Memory (K)

```

Total: 31,200
Paged: 15,816
Nonpaged: 15,384

```

```

Commit Charge (K)
Total: 59,460
Limit: 1,417,552
Peak: 60,580

```

```

Pagefile Space (K)
Total: 1,048,576
Total in use: 2,220
Peak: 2,220

```

```

C:\pagefile.sys
Total: 1,048,576
Total in use: 2,220
Peak: 2,220

```

Services Report

```

-----
----
Alerter                               Stopped (Manual)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    LanmanWorkstation
Computer Browser                       Stopped (Manual)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    LanmanWorkstation
    LanmanServer
    LmHosts
ClipBook Server                         Stopped (Manual)
  C:\WINNT\system32\clipsrv.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Own Process
  Service Dependencies:
    NetDDE
DHCP Client (TDI)                       Stopped (Disabled)
  C:\WINNT\System32\services.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Shared Process
  Service Dependencies:
    Tcpip
    Afd
    NetBT

```


EventLog (Event log) (Automatic) C:\WINNT\system32\services.exe Service Account Name: LocalSystem Error Severity: Normal Service Flags: Shared Process	Running		Service Account Name: LocalSystem Error Severity: Normal Service Flags: Shared Process		
Server (Automatic) C:\WINNT\System32\services.exe Service Account Name: LocalSystem Error Severity: Normal Service Flags: Shared Process Group Dependencies: TDI	Running		Net Logon (RemoteValidation) C:\WINNT\System32\lsass.exe Service Account Name: LocalSystem Error Severity: Normal Service Flags: Shared Process Service Dependencies: LanmanWorkstation LmHosts	Stopped	(Manual)
Workstation (NetworkProvider) (Automatic) C:\WINNT\System32\services.exe Service Account Name: LocalSystem Error Severity: Normal Service Flags: Shared Process Group Dependencies: TDI	Running		NT LM Security Support Provider C:\WINNT\System32\SERVICES.EXE Service Account Name: LocalSystem Error Severity: Normal Service Flags: Shared Process	Running	(Manual)
License Logging Service C:\WINNT\System32\llssrv.exe Service Account Name: LocalSystem Error Severity: Normal Service Flags: Own Process	Stopped	(Manual)	Plug and Play (PlugPlay) C:\WINNT\system32\services.exe Service Account Name: LocalSystem Error Severity: Normal Service Flags: Shared Process	Stopped	(Manual)
TCP/IP NetBIOS Helper C:\WINNT\System32\services.exe Service Account Name: LocalSystem Error Severity: Normal Service Flags: Shared Process Group Dependencies: NetworkProvider	Stopped	(Manual)	Protected Storage (Automatic) C:\WINNT\System32\pstores.exe Service Account Name: LocalSystem Error Severity: Normal Service Flags: Own Process, Interactive Service Dependencies: RpcSs	Running	
Messenger (Automatic) C:\WINNT\System32\services.exe Service Account Name: LocalSystem Error Severity: Normal Service Flags: Shared Process Service Dependencies: LanmanWorkstation NetBios	Running		Directory Replicator C:\WINNT\System32\lmrepl.exe Service Account Name: LocalSystem Error Severity: Normal Service Flags: Own Process Service Dependencies: LanmanWorkstation LanmanServer	Stopped	(Manual)
Network DDE (NetDDEGroup) C:\WINNT\system32\netdde.exe Service Account Name: LocalSystem Error Severity: Normal Service Flags: Shared Process Service Dependencies: NetDDESDM	Stopped	(Manual)	Remote Command Service (Automatic) c:\benchcrf\rsys.exe Service Account Name: LocalSystem Error Severity: Normal Service Flags: Own Process, Interactive	Running	
Network DDE DSDM C:\WINNT\system32\netdde.exe	Stopped	(Manual)	Remote Procedure Call (RPC) Locator C:\WINNT\System32\LOCATOR.EXE Service Account Name: LocalSystem Error Severity: Normal Service Flags: Own Process Service Dependencies: LanmanWorkstation Rdr	Stopped	(Manual)

Remote Procedure Call (RPC) Service (Automatic)	Running		----- ----		
C:\WINNT\system32\RpcSs.exe			Abiosdsk (Primary disk)	Stopped	(Disabled)
Service Account Name: LocalSystem			Error Severity: Ignore		
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process		
Service Flags: Own Process			AFD Networking Support Environment (TDI)	Running	
Schedule	Stopped	(Manual)	(Automatic)		
C:\WINNT\System32\AtSvc.Exe			C:\WINNT\System32\drivers\afd.sys		
Service Account Name: LocalSystem			Error Severity: Normal		
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process		
Service Flags: Own Process			Aha154x (SCSI miniport)	Stopped	(Disabled)
Spooler (SpoolerGroup)	Stopped	(Manual)	Error Severity: Normal		
C:\WINNT\system32\spoolss.exe			Service Flags: Kernel Driver, Shared Process		
Service Account Name: LocalSystem			Aha174x (SCSI miniport)	Stopped	(Disabled)
Error Severity: Normal			Error Severity: Normal		
Service Flags: Own Process, Interactive			Service Flags: Kernel Driver, Shared Process		
Telephony Service	Stopped	(Manual)	aic78xx (SCSI miniport)	Stopped	(Disabled)
C:\WINNT\system32\tapisrv.exe			Error Severity: Normal		
Service Account Name: LocalSystem			Service Flags: Kernel Driver, Shared Process		
Error Severity: Normal			Always (SCSI miniport)	Stopped	(Disabled)
Service Flags: Own Process			Error Severity: Normal		
TUXEDO IPC Helper	Running		Service Flags: Kernel Driver, Shared Process		
(Automatic)			ami0nt (SCSI miniport)	Stopped	(Disabled)
C:\TUXEDO\bin\tuxipc.exe			Error Severity: Normal		
Service Account Name: LocalSystem			Service Flags: Kernel Driver, Shared Process		
Error Severity: Normal			amsint (SCSI miniport)	Stopped	(Disabled)
Service Flags: Own Process			Error Severity: Normal		
TListen (Port: 3050)	Running		Service Flags: Kernel Driver, Shared Process		
(Automatic)			Arrow (SCSI miniport)	Stopped	(Disabled)
C:\TUXEDO\bin\slisten.exe			Error Severity: Normal		
Service Account Name: LocalSystem			Service Flags: Kernel Driver, Shared Process		
Error Severity: Normal			atapi (SCSI miniport)	Stopped	(Disabled)
Service Flags: Own Process			Error Severity: Normal		
UPS	Stopped	(Manual)	Service Flags: Kernel Driver, Shared Process		
C:\WINNT\System32\ups.exe			Atdisk (Primary disk)	Stopped	(Disabled)
Service Account Name: LocalSystem			Error Severity: Ignore		
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process		
Service Flags: Own Process			ati (Video)	Stopped	(Disabled)
World Wide Web Publishing Service	Running		Error Severity: Ignore		
(Automatic)			Service Flags: Kernel Driver, Shared Process		
C:\WINNT\System32\inetsrv\inetinfo.exe			Beep (Base)	Running	(System)
Service Account Name: LocalSystem			Error Severity: Normal		
Error Severity: Ignore			Service Flags: Kernel Driver, Shared Process		
Service Flags: Shared Process			BusLogic (SCSI miniport)	Stopped	(Disabled)
Service Dependencies:			Error Severity: Normal		
RPCSS			Service Flags: Kernel Driver, Shared Process		
NTLMSSP			Busmouse (Pointer Port)	Stopped	(Disabled)
			Error Severity: Ignore		
			Service Flags: Kernel Driver, Shared Process		
Drivers Report			Cdaudio (Filter)	Stopped	(System)
			Error Severity: Ignore		
			Service Flags: Kernel Driver, Shared Process		

Cdfs (File system)	Running	(Disabled)	Service Flags: Kernel Driver, Shared Process
Error Severity: Normal			et4000 (Video) Stopped (Disabled)
Service Flags: File System Driver, Shared Process			Error Severity: Ignore
Group Dependencies:			Service Flags: Kernel Driver, Shared Process
SCSI CDROM Class			Fastfat (Boot file system) Stopped (Disabled)
Cdrom (SCSI CDROM Class)	Running	(System)	Error Severity: Normal
Error Severity: Ignore			Service Flags: File System Driver, Shared Process
Service Flags: Kernel Driver, Shared Process			Fdl6_700 (SCSI miniport) Stopped (Disabled)
Group Dependencies:			Error Severity: Normal
SCSI miniport			Service Flags: Kernel Driver, Shared Process
Changer (Filter)	Stopped	(System)	Fd7000ex (SCSI miniport) Stopped (Disabled)
Error Severity: Ignore			Error Severity: Normal
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process
cirrus (Video)	Running	(System)	Fd8xx (SCSI miniport) Stopped (Disabled)
Error Severity: Normal			Error Severity: Normal
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process
Cpqarray (SCSI miniport)	Stopped	(Disabled)	flashpnt (SCSI miniport) Stopped (Disabled)
Error Severity: Normal			Error Severity: Normal
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process
cpqfw2e (SCSI miniport)	Stopped	(Disabled)	Floppy (Primary disk) Running (System)
Error Severity: Normal			Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process
dac960nt (SCSI miniport)	Stopped	(Disabled)	Ftdisk (Filter) Stopped (Disabled)
Error Severity: Normal			Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process
dce376nt (SCSI miniport)	Stopped	(Disabled)	i8042 Keyboard and PS/2 Mouse Port Driver (Keyboard Port) Running (System)
Error Severity: Normal			System32\DRIVERS\i8042prt.sys
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal
Delldsa (SCSI miniport)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process
Error Severity: Normal			Inport (Pointer Port) Stopped (Disabled)
Service Flags: Kernel Driver, Shared Process			Error Severity: Ignore
Dell_DGX (Video)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process
Error Severity: Ignore			Jazzg300 (Video) Stopped (Disabled)
Service Flags: Kernel Driver, Shared Process			Error Severity: Ignore
Disk (SCSI Class)	Running	(Boot)	Service Flags: Kernel Driver, Shared Process
Error Severity: Ignore			Jazzg364 (Video) Stopped (Disabled)
Service Flags: Kernel Driver, Shared Process			Error Severity: Ignore
Group Dependencies:			Service Flags: Kernel Driver, Shared Process
SCSI miniport			Jzvxl484 (Video) Stopped (Disabled)
Diskperf (Filter)	Stopped	(Disabled)	Error Severity: Ignore
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process
Service Flags: Kernel Driver, Shared Process			Keyboard Class Driver (Keyboard Class) Running (System)
DptScsi (SCSI miniport)	Stopped	(Disabled)	System32\DRIVERS\kbdclass.sys
Error Severity: Normal			Error Severity: Normal
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process
dtc329x (SCSI miniport)	Stopped	(Disabled)	KSecDD (Base) Running (System)
Error Severity: Normal			Error Severity: Normal
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process
Intel(R) PRO NDIS Driver (NDIS)	Running		mga (Video) Stopped (Disabled)
(Automatic)			Error Severity: Ignore
C:\WINNT\System32\drivers\e100bnt.sys			Service Flags: Kernel Driver, Shared Process
Error Severity: Normal			

mga_mil (Video)	Stopped	(Disabled)	C:\WINNT\system32\drivers\netdect.sys	
Error Severity: Ignore			Error Severity: Normal	
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process	
mitsumi (SCSI miniport)	Stopped	(Disabled)	Npfs (File system)	Running (System)
Error Severity: Normal			Error Severity: Normal	
Service Flags: Kernel Driver, Shared Process			Service Flags: File System Driver, Shared Process	
mkecr5xx (SCSI miniport)	Stopped	(Disabled)	Ntfs (File system)	Running (Disabled)
Error Severity: Normal			Error Severity: Normal	
Service Flags: Kernel Driver, Shared Process			Service Flags: File System Driver, Shared Process	
Modem (Extended base)	Stopped	(Manual)	Null (Base)	Running (System)
Error Severity: Ignore			Error Severity: Normal	
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process	
Mouse Class Driver (Pointer Class)	Running	(System)	Oliscsi (SCSI miniport)	Stopped (Disabled)
System32\DRIVERS\mouclass.sys			Error Severity: Normal	
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process	
Service Flags: Kernel Driver, Shared Process			Parallel (Extended base)	Stopped
Msfs (File system)	Running	(System)	(Automatic)	
Error Severity: Normal			Error Severity: Ignore	
Service Flags: File System Driver, Shared Process			Service Flags: Kernel Driver, Shared Process	
Mup (Network)	Running	(Manual)	Service Dependencies:	
C:\WINNT\System32\drivers\mup.sys			Parport	
Error Severity: Normal			Group Dependencies:	
Service Flags: File System Driver, Shared Process			Parallel arbitrator	
Ncr53c9x (SCSI miniport)	Stopped	(Disabled)	Parport (Parallel arbitrator)	Stopped
Error Severity: Normal			(Automatic)	
Service Flags: Kernel Driver, Shared Process			Error Severity: Ignore	
ncr77c22 (Video)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process	
Error Severity: Ignore			ParVdm (Extended base)	Stopped
Service Flags: Kernel Driver, Shared Process			(Automatic)	
Ncrc700 (SCSI miniport)	Stopped	(Disabled)	Error Severity: Ignore	
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process	
Service Flags: Kernel Driver, Shared Process			Service Dependencies:	
Ncrc710 (SCSI miniport)	Stopped	(Disabled)	Parport	
Error Severity: Normal			Group Dependencies:	
Service Flags: Kernel Driver, Shared Process			Parallel arbitrator	
Microsoft NDIS System Driver (NDIS)	Running	(System)	PCIDump (PCI Configuration)	Stopped (System)
Error Severity: Normal			Error Severity: Ignore	
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process	
NetBIOS Interface (NetBIOSGroup)	Running	(Manual)	Pcmcia (System Bus Extender)	Stopped (Disabled)
C:\WINNT\System32\drivers\netbios.sys			Error Severity: Normal	
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process	
Service Flags: File System Driver, Shared Process			PnP ISA Enabler Driver (Base)	Stopped (System)
Group Dependencies:			Error Severity: Ignore	
TDI			Service Flags: Kernel Driver, Shared Process	
WINS Client (TCP/IP) (PNP_TDI)	Running		psidisp (Video)	Stopped (Disabled)
(Automatic)			Error Severity: Ignore	
C:\WINNT\System32\drivers\netbt.sys			Service Flags: Kernel Driver, Shared Process	
Error Severity: Normal			Ql10wnt (SCSI miniport)	Stopped (Disabled)
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal	
Service Dependencies:			Service Flags: Kernel Driver, Shared Process	
Tcpip			qv (Video)	Stopped (Disabled)
NetDetect	Stopped	(Manual)	Error Severity: Ignore	

Service Flags: Kernel Driver, Shared Process
 Rdr (Network) Running (Manual)
 C:\WINNT\System32\drivers\rdr.sys
 Error Severity: Normal
 Service Flags: File System Driver, Shared Process
 s3 (Video) Stopped (Disabled)
 Error Severity: Ignore
 Service Flags: Kernel Driver, Shared Process
 Scsiprnt (Extended base) Stopped (Automatic)
 Error Severity: Ignore
 Service Flags: Kernel Driver, Shared Process
 Group Dependencies:
 SCSI miniport
 Scsiscan (SCSI Class) Running (System)
 Error Severity: Ignore
 Service Flags: Kernel Driver, Shared Process
 Group Dependencies:
 SCSI miniport
 Serial (Extended base) Stopped (Automatic)
 Error Severity: Ignore
 Service Flags: Kernel Driver, Shared Process
 Sermouse (Pointer Port) Stopped (Disabled)
 Error Severity: Ignore
 Service Flags: Kernel Driver, Shared Process
 Sfloppy (Primary disk) Stopped (System)
 Error Severity: Ignore
 Service Flags: Kernel Driver, Shared Process
 Group Dependencies:
 SCSI miniport
 Simbad (Filter) Stopped (Disabled)
 Error Severity: Normal
 Service Flags: Kernel Driver, Shared Process
 slcd32 (SCSI miniport) Stopped (Disabled)
 Error Severity: Normal
 Service Flags: Kernel Driver, Shared Process
 Sparrow (SCSI miniport) Stopped (Disabled)
 Error Severity: Normal
 Service Flags: Kernel Driver, Shared Process
 Spock (SCSI miniport) Stopped (Disabled)
 Error Severity: Normal
 Service Flags: Kernel Driver, Shared Process
 Srv (Network) Running (Manual)
 C:\WINNT\System32\drivers\srvc.sys
 Error Severity: Normal
 Service Flags: File System Driver, Shared Process
 symc810 (SCSI miniport) Stopped (Disabled)
 Error Severity: Normal
 Service Flags: Kernel Driver, Shared Process
 symc8XX (SCSI miniport) Running (Boot)
 C:\WINNT\system32\drivers\symc8XX.sys

Error Severity: Normal
 Service Flags: Kernel Driver, Shared Process
 T128 (SCSI miniport) Stopped (Disabled)
 Error Severity: Normal
 Service Flags: Kernel Driver, Shared Process
 T13B (SCSI miniport) Stopped (Disabled)
 Error Severity: Normal
 Service Flags: Kernel Driver, Shared Process
 TCP/IP Service (PNP_TDI) Running (Automatic)
 C:\WINNT\System32\drivers\tcpip.sys
 Error Severity: Normal
 Service Flags: Kernel Driver, Shared Process
 tga (Video) Stopped (Disabled)
 Error Severity: Ignore
 Service Flags: Kernel Driver, Shared Process
 tmv1 (SCSI miniport) Stopped (Disabled)
 Error Severity: Normal
 Service Flags: Kernel Driver, Shared Process
 Ultra124 (SCSI miniport) Stopped (Disabled)
 Error Severity: Normal
 Service Flags: Kernel Driver, Shared Process
 Ultra14f (SCSI miniport) Stopped (Disabled)
 Error Severity: Normal
 Service Flags: Kernel Driver, Shared Process
 Ultra24f (SCSI miniport) Stopped (Disabled)
 Error Severity: Normal
 Service Flags: Kernel Driver, Shared Process
 update (Base) Stopped (System)
 Error Severity: Ignore
 Service Flags: Kernel Driver, Shared Process
 v7vram (Video) Stopped (Disabled)
 Error Severity: Ignore
 Service Flags: Kernel Driver, Shared Process
 VgaSave (Video Save) Stopped (System)
 C:\WINNT\System32\drivers\vga.sys
 Error Severity: Ignore
 Service Flags: Kernel Driver, Shared Process
 VgaStart (Video Init) Stopped (System)
 C:\WINNT\System32\drivers\vga.sys
 Error Severity: Ignore
 Service Flags: Kernel Driver, Shared Process
 Wd33c93 (SCSI miniport) Stopped (Disabled)
 Error Severity: Normal
 Service Flags: Kernel Driver, Shared Process
 wd90c24a (Video) Stopped (Disabled)
 Error Severity: Ignore
 Service Flags: Kernel Driver, Shared Process
 wdvga (Video) Stopped (Disabled)
 Error Severity: Ignore
 Service Flags: Kernel Driver, Shared Process
 weitek9 (Video) Stopped (Disabled)

Error Severity: Ignore
 Service Flags: Kernel Driver, Shared Process
 Xga (Video) Stopped (Disabled)
 Error Severity: Ignore
 Service Flags: Kernel Driver, Shared Process

IRQ and Port Report

 Devices Vector Level Affinity

 MPS 1.4 - APIC platform 8 8 0x00000003
 MPS 1.4 - APIC platform 0 0 0x00000003
 MPS 1.4 - APIC platform 1 1 0x00000003
 MPS 1.4 - APIC platform 2 2 0x00000003
 MPS 1.4 - APIC platform 3 3 0x00000003
 MPS 1.4 - APIC platform 4 4 0x00000003
 MPS 1.4 - APIC platform 5 5 0x00000003
 MPS 1.4 - APIC platform 6 6 0x00000003
 MPS 1.4 - APIC platform 7 7 0x00000003
 MPS 1.4 - APIC platform 8 8 0x00000003
 MPS 1.4 - APIC platform 9 9 0x00000003
 MPS 1.4 - APIC platform 10 10 0x00000003
 MPS 1.4 - APIC platform 11 11 0x00000003
 MPS 1.4 - APIC platform 12 12 0x00000003
 MPS 1.4 - APIC platform 13 13 0x00000003
 MPS 1.4 - APIC platform 14 14 0x00000003
 MPS 1.4 - APIC platform 15 15 0x00000003
 MPS 1.4 - APIC platform 16 16 0x00000003
 MPS 1.4 - APIC platform 17 17 0x00000003
 MPS 1.4 - APIC platform 18 18 0x00000003
 MPS 1.4 - APIC platform 19 19 0x00000003
 MPS 1.4 - APIC platform 20 20 0x00000003
 MPS 1.4 - APIC platform 21 21 0x00000003
 MPS 1.4 - APIC platform 22 22 0x00000003
 MPS 1.4 - APIC platform 23 23 0x00000003
 MPS 1.4 - APIC platform 24 24 0x00000003
 MPS 1.4 - APIC platform 25 25 0x00000003
 MPS 1.4 - APIC platform 26 26 0x00000003
 MPS 1.4 - APIC platform 27 27 0x00000003
 MPS 1.4 - APIC platform 28 28 0x00000003
 MPS 1.4 - APIC platform 29 29 0x00000003
 MPS 1.4 - APIC platform 30 30 0x00000003
 MPS 1.4 - APIC platform 31 31 0x00000003
 MPS 1.4 - APIC platform 32 32 0x00000003
 MPS 1.4 - APIC platform 33 33 0x00000003
 MPS 1.4 - APIC platform 34 34 0x00000003
 MPS 1.4 - APIC platform 35 35 0x00000003
 MPS 1.4 - APIC platform 36 36 0x00000003

MPS 1.4 - APIC platform 37 37 0x00000003
 MPS 1.4 - APIC platform 38 38 0x00000003
 MPS 1.4 - APIC platform 39 39 0x00000003
 MPS 1.4 - APIC platform 40 40 0x00000003
 MPS 1.4 - APIC platform 41 41 0x00000003
 MPS 1.4 - APIC platform 42 42 0x00000003
 MPS 1.4 - APIC platform 43 43 0x00000003
 MPS 1.4 - APIC platform 44 44 0x00000003
 MPS 1.4 - APIC platform 45 45 0x00000003
 MPS 1.4 - APIC platform 46 46 0x00000003
 MPS 1.4 - APIC platform 47 47 0x00000003
 MPS 1.4 - APIC platform 61 61 0x00000003
 MPS 1.4 - APIC platform 65 65 0x00000003
 MPS 1.4 - APIC platform 80 80 0x00000003
 MPS 1.4 - APIC platform 193 193 0x00000003
 MPS 1.4 - APIC platform 225 225 0x00000003
 MPS 1.4 - APIC platform 253 253 0x00000003
 MPS 1.4 - APIC platform 254 254 0x00000003
 MPS 1.4 - APIC platform 255 255 0x00000003
 i8042prt 1 1 0xffffffff
 i8042prt 12 12 0xffffffff
 E100B 3 3 0x00000000
 E100B 10 10 0x00000000
 E100B 15 15 0x00000000
 E100B 5 5 0x00000000
 E100B 14 14 0x00000000
 E100B 4 4 0x00000000
 Floppy 6 6 0x00000000
 symc8XX 7 7 0x00000000

 Devices Physical Address Length

 MPS 1.4 - APIC platform 0x00000000 0x000000010
 MPS 1.4 - APIC platform 0x00000020 0x000000002
 MPS 1.4 - APIC platform 0x00000040 0x000000004
 MPS 1.4 - APIC platform 0x00000048 0x000000004
 MPS 1.4 - APIC platform 0x00000061 0x000000001
 MPS 1.4 - APIC platform 0x00000070 0x000000002
 MPS 1.4 - APIC platform 0x00000080 0x000000010
 MPS 1.4 - APIC platform 0x00000092 0x000000001
 MPS 1.4 - APIC platform 0x000000a0 0x000000002
 MPS 1.4 - APIC platform 0x000000c0 0x000000010
 MPS 1.4 - APIC platform 0x000000f0 0x000000010
 i8042prt 0x00000060 0x000000001
 i8042prt 0x00000064 0x000000001
 E100B 0x0000e400 0x00000001e
 E100B 0x0000e800 0x00000001e
 E100B 0x0000d400 0x00000001e
 E100B 0x0000d000 0x00000001e
 E100B 0x0000c800 0x00000001e

```

E100B          0x0000e000  0x000000001e
Floppy         0x000003f0  0x0000000006
Floppy         0x000003f7  0x0000000001
symc8XX       0x0000d800  0x0000000100
cirrus        0x000003b0  0x000000000c
cirrus        0x000003c0  0x0000000020

```

```

PROCESSOR_LEVEL=6
PROCESSOR_REVISION=0502
TMCONTEXTS=1
TUXCONFIG=C:\InetPub\wwwroot\tuxconfig
TUXDIR=C:\TUXEDO
windir=C:\WINNT

```

DMA and Memory Report

```

-----
Devices          Channel   Port
-----

```

```

Floppy           2       0
-----

```

```

-----
Devices          Physical Address  Length
-----

```

```

MPS 1.4 - APIC platform  0xfec00000  0x00000400
MPS 1.4 - APIC platform  0xfec00000  0x00000400
E100B                   0xfecfd000  0x0000001e
E100B                   0xfecfe000  0x0000001e
E100B                   0xf7ffe000  0x0000001e
E100B                   0xf7ffd000  0x0000001e
E100B                   0xf7ffc000  0x0000001e
E100B                   0xfecfc000  0x0000001e
symc8XX                 0xfe8ff800  0x00000100
symc8XX                 0xfe8fd000  0x00000100
cirrus                   0x000a0000  0x00020000
cirrus                   0xf4000000  0x02000000

```

Environment Report

```

-----

```

System Environment Variables

```

APPDIR=C:\InetPub\wwwroot
ComSpec=C:\WINNT\system32\cmd.exe
NUMBER_OF_PROCESSORS=2
OS=Windows_NT
Os2LibPath=C:\WINNT\system32\os2\dll;
Path=C:\WINNT\system32;C:\WINNT;C:\MSSQL7\BINN;C:\TUXEDO\bin
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 5 Stepping 2,
GenuineIntel

```

Environment Variables for Current User

```

TEMP=C:\TEMP
TMP=C:\TEMP

```

Network Report

```

-----

```

```

Your Access Level: Admin & Local
Workgroup or Domain: CWIEN
Network Version: 4.0
LanRoot: CWIEN
Logged On Users: 2
Current User (1): Administrator
    Logon Domain: WEISS
    Logon Server: WEISS
Current User (2): Administrator
    Logon Domain: WEISS
    Logon Server: WEISS

```

```

Transport: NetBT_E100B1, 00-A0-C9-30-33-91, VC's: 0, Wan: Wan
Transport: NetBT_E100B3, 00-A0-C9-AB-00-98, VC's: 0, Wan: Wan
Transport: NetBT_E100B4, 00-A0-C9-42-64-4B, VC's: 0, Wan: Wan
Transport: NetBT_E100B5, 00-A0-C9-42-64-D7, VC's: 0, Wan: Wan
Transport: NetBT_E100B6, 00-A0-C9-AB-15-B8, VC's: 0, Wan: Wan
Transport: NetBT_E100B2, 00-A0-C9-AB-01-1B, VC's: 0, Wan: Wan

```

```

Character Wait: 3,600
Collection Time: 250
Maximum Collection Count: 16
Keep Connection: 600
Maximum Commands: 5
Session Time Out: 45
Character Buffer Size: 512
Maximum Threads: 17
Lock Quota: 6,144
Lock Increment: 10
Maximum Locks: 500
Pipe Increment: 10
Maximum Pipes: 500
Cache Time Out: 40
Dormant File Limit: 45

```

Read Ahead Throughput: 4,294,967,295
Mailslot Buffers: 3
Server Announce Buffers: 20
Illegal Datagrams: 5
Datagram Reset Frequency: 60
Log Election Packets: False
Use Opportunistic Locking: True
Use Unlock Behind: True
Use Close Behind: True
Buffer Pipes: True
Use Lock, Read, Unlock: True
Use NT Caching: True
Use Raw Read: True
Use Raw Write: True
Use Write Raw Data: True
Use Encryption: True
Buffer Deny Write Files: True
Buffer Read Only Files: True
Force Core Creation: True
512 Byte Max Transfer: False
Bytes Received: 253
SMB's Received: 3
Paged Read Bytes Requested: 0
Non Paged Read Bytes Requested: 0
Cache Read Bytes Requested: 0
Network Read Bytes Requested: 0
Bytes Transmitted: 477
SMB's Transmitted: 3
Paged Read Bytes Requested: 0
Non Paged Read Bytes Requested: 0
Cache Read Bytes Requested: 0
Network Read Bytes Requested: 0
Initially Failed Operations: 0
Failed Completion Operations: 0
Read Operations: 0
Random Read Operations: 0
Read SMB's: 0
Large Read SMB's: 0
Small Read SMB's: 0
Write Operations: 0
Random Write Operations: 0
Write SMB's: 0
Large Write SMB's: 0
Small Write SMB's: 0
Raw Reads Denied: 0
Raw Writes Denied: 0
Network Errors: 0
Sessions: 1
Failed Sessions: 0
Reconnects: 0
Core Connects: 0
LM 2.0 Connects: 0

LM 2.x Connects: 0
Windows NT Connects: 1
Server Disconnects: 0
Hung Sessions: 0
Use Count: 0
Failed Use Count: 0
Current Commands: 0
Server File Opens: 0
Server Device Opens: 0
Server Jobs Queued: 0
Server Session Opens: 0
Server Sessions Timed Out: 0
Server Sessions Errored Out: 0
Server Password Errors: 0
Server Permission Errors: 0
Server System Errors: 0
Server Bytes Sent: 253
Server Bytes Received: 477
Server Average Response Time: 0
Server Request Buffers Needed: 0
Server Big Buffers Needed: 0

Appendix D - Pricing Details

This appendix contains the calculations used to determine the number of disk drives and the number of LAN segments necessary in the priced configuration and the spreadsheet used to determine the price/performance figure.

180 Day Space Calculation

*The following worksheet was used to calculate the 180 day space of the system.
Note: Numbers are in 2K pages unless otherwise specified*

Note : Numbers are in Kbytes unless otherwise specified						
Warehouses	1900	tpmC	23670	tpmC/W	12.41	
Table	Rows	Data	Index	5% Space	8H Space	Total Space
Warehouse	1,900	208	48	13		269
District	19,000	2,112	56	108		2,276
Item	100,000	9,528	80	221		9,829
New-order	17,100,000	270,360	744		38,000	309,104
History	57,000,000	3,166,728	0		526,073	3,692,801
Orders	57,000,000	1,747,128	965,008		450,554	3,162,690
Customer	57,000,000	41,454,552	2,661,912	1,014,679		45,131,143
Order-line	570,001,928	35,625,128	88,728		5,932,969	41,646,825
Stock	190,000,000	60,800,000	136,272	1,401,534		62,337,806
Totals		143,075,744	3,852,848	2,416,555	6,947,596	156,292,743
Segment	LogDev Cnt.	Seg. Size	Needed	Overhead	Not Needed	
misc	6	58,368,000	49,312,032	493,120		8,562,847
customer/stock	6	110,284,800	108,543,638	1,085,436		655,725
Totals		168,652,800	157,855,671	1,578,557		9,218,572
Dynamic space	39,444,431	Sum of Data for Order, Order-Line and History (excluding free extents)				
Static space	111,479,272	Data + Index + 5% Space + Overhead - Dynamic space				
Free space	8,510,524	Total Seg. Size - Dynamic Space - Static Space - Not Needed				
Daily growth	7,829,097	(Dynamic space/W* 62.5)* tpmC				
Daily spread	-3,233,122	Free space - 1.5 * Daily growth (zero if negative)				
180 day (KB)	1,520,716,702	Static space + 180 (daily growth + daily spread)				
180 day (GB)	1,450,27	Excludes OS, Paging and RDBMS Logs				
Log size (MB)	45,000	Total size of log file				
% Log used	23,2595	% of log file used during entire run				
Total N-O Txn	2064406	Total count of N-O transactions during entire run				
Log per N-O tx	5,1918	K per New-Order transaction				
8 Hour Log (GB)	56,02					
Disk Capacity	MB	GB	bits needed	disks priced		
9GB 10000 rpm	8,676	8,47				
180 day (GB)		1,450,27	171,17	192		
Disk Capacity	MB	GB	bits needed	disks priced		
18 GB 10000 rpm	17,366	16,96				
8 Hour Log (RAID 1)		56,02	3,30	444		

Price/Performance Spreadsheet *The following detailed worksheet was used to calculate the price/performance of the system.*

Description	Part Number	Third Party Brand	Unit Pricing	Qty.	Extended Price	5Yr Maint. Price
Base System						
2. Pentium III Xeon Processor 500MHz/2MB						
Memory 512MB EDO-DIMM						
DAT Drive DDS-3 12GB, SE-SCSI						
9GB/10K LVD-SCSI, Hot Plug						
Power Supply (add)						
Keyboard						
Country Pack						
Monitor MCM 1510						
				1	\$65,673	
Server Hardware Subtotal						
RAID-Controller;3 Chan.,32Mb, LVD						
RAID-Controller;3 Chan.,32Mb, LVD (10% spares						
18GB/10K LVD, Hot Plug (incl. 10% spare)						
APC Smart UPS 1400 (incl. 10% spare)						
9GB/10K LVD-SCSI, Hot Plug						
9GB/10K LVD-SCSI, Hot Plug (10% spares)						
Primergy S40 ES 1 channel stack						
SCSI Cable UHD-HD LVD 5m						
				1	\$156	\$2,808
				18		\$2,808
Storage Subtotal						
Maint. Server + Storage						
SNP-SY-F1960E32-P				1	\$2,133	\$14,931
SNP-SY-F1960E32-P				1	\$2,133	\$4,266
SNP-SY-F1899L118-P				1	\$1,660	\$16,600
588203 APC						
				5	\$587	\$1,761
SNP-SY-F1899E109-P				1	\$933	\$179,136
SNP-SY-F1899L109-P				1	\$949	\$18,980
SNP-SY-K4538V201-P				1	\$2,356	\$42,408
SNP-SY-F1947L50-P				1	\$156	\$2,808
Storage Subtotal						
Maint. Server + Storage						
SNP-SY-K482V714-A				4	\$2,289	\$9,156
SNP-SY-F1727E450-A				4	\$1,200	\$4,800
S26381-K271-V310				4	\$25	\$100
SNP-SY-F1699B153-A				4	\$36	\$144,00
SNP-SY-F1867E503-A				4	\$578	\$2,312
SNP-SY-F1867E504-A				4	\$1,289	\$5,156
S26361-F1724-E40				4	\$489	\$1,966
SNP-SY-F1868E501-A				24	\$111	\$2,664
S26361-K494-V150				4	\$191	\$764
Client Hardware Subtotal						
Client Hardware Subtotal						
				1		\$27,052
Microsoft Windows NT-Server, Enterprise Edition 4.0, incl 25 CALs						
MS SQL-Server 7.0 Enterp.Edition unlim. License				2	\$3,999	\$3,999
				2	\$28,999	\$28,999
Server Software Subtotal						
Server Software Subtotal						
				1		\$10,475
Microsoft Windows NT-Server 4.0, incl 5 CALs						
Tuxedo 6.4 Core Functionality Services for NT				2	\$809	\$3,236
Microsoft Visual C++ Professional 5.0				4	\$3,000	\$12,000
				1	\$499	\$499
Client Software Subtotal						
Client Software Subtotal						
				2	\$809	\$3,236
				4	\$3,000	\$12,000
				1	\$499	\$499
Complex 8-port 10Baset Hub (8+1) ports (incl.10% Kingston Switch 8 port 10/100 incl. 10% spare						
				5	\$29,00	\$75,922
				3	\$605	\$1,815
User Connectivity Subtotal						
User Connectivity Subtotal						
				5	\$29,00	\$75,922
				3	\$605	\$1,815
Total						
					\$500,085	\$51,642
					\$77,737	
					\$29,000	\$75,922
					\$605	\$1,815
					\$500,085	\$51,642
					\$551,727	\$551,727
					tpmC	23670,33
					\$/tpmC	\$23,41

1=Siemens, 2=Microsoft, 4=BEA Systems, 5=Software House

Five-Year Cost of Ownership

tpmC \$551,727
\$/tpmC 23670,33
\$23,41

Appendix E - Price Quotations



EYEMANAGEMENT SOLUTIONS

March 11, 1999

Miguel Isenberg
 Database Benchmarking
 ICP OS PS DS B1
 Siemens AG
 Paderborn, Germany
 Fax: +49-5231-815148

Dear Mr. Isenberg,

Per your request I am enclosing the pricing information regarding TUXEDO 9.2 that you requested. This pricing applies to Tuxedo 8.1, 8.2, 8.3 and 8.4. Please note that Tuxedo 6.4 is our most recent version of Tuxedo, but that all 6.x releases are generally available. Core functionality services pricing is appropriate for your activities. As per the table below, server systems are eligible in one of 5 tiers based on CPUs type and capacity. Initial major processor based systems with 2 CPUs capacity are classified as Tier 1 systems, those with 4 CPU capacity are Tier 2. This quote is valid for 90 days from the date of issue of this letter.

Tuxedo Core Functionality Services (CFS) Program Product Pricing and Description

TUX-CFS provides a user level or rackmount support for distributed computing, and is best used by organizations with substantial resources and knowledge for advanced distributed computing implementation.

TUX-CFS price are cover only and are based on the overall performance characteristics of the server and uses the same five tier cost/size classification as TUXEDO 8.x. Prices range from \$3,000 for Tier 1 to \$29,000 for Tier 5. Under this pricing option (C/F/S) systems ranging TUX-CFS at the user site must have a TUXEDO license installed and pay the appropriate per server license fee.

BEA TUX-CFS Unlimited User Licenses Price Per Server

Unlimited User Licenses Price per server	Number of Users	Dollar Amount	Maintenance per x 5/yr	Maintenance per x 5/yr
Tier 1 - PC Servers with 1 or 2 CPUs, entry level RISC Unix processor workstations and servers	Unlimited	\$3,000.00	\$450.00	\$450.00
Tier 2 - PC Servers with 3 or 4 CPUs, Midrange RISC Unix processor servers and workstations	Unlimited	\$17,000.00	\$1,800.00	\$1,800.00
Tier 3 - Midrange	Unlimited	\$30,000.00	\$4,000.00	\$4,000.00

10/3/97

SEA SYSTEMS, INC.

Multiprocessors, up to 8 CPUs per system capacity					
Tier 4 -- Large (more than 8, less than 32 CPUs) and Mainframe Systems	Unlimited	\$100,000.00	\$15,000.00	\$22,000.00	
Tier 5 -- Massively Parallel Systems, > 32 processors	Unlimited	\$250,000.00	\$37,500.00	\$65,000.00	

Intel based server tier classifications:

Platform	Operating System	Tier 1	Tier 1	Tier 2	Tier 3	Tier 3
Intel Pentium/ Pentium Pro PCs	Interactive ESIX SVR 4.0 SCO UNIX 3.2.2 ams 3.2.4 SCO ODT 2.x, 3.x Solaris x86 2.x Lynx/Novo, Windows NT 3.514.0	All 386/486 PCs are Class 1	ALL Pentium and Pentium Pro PCs with 1 or 2 CPUs capacity are Tier 1	ALL Pentium and Pentium Pro PCs with 3 or 4 CPUs capacity are Tier 2		ALL Pentium and Pentium Pro PCs with 5, 6, 7, or 8 CPUs are Tier 3

Very Truly Yours,



Lewis D. Brentano,
Director, Market Planning

13. P. 79/121

Software House International Pricing Proposal	Quotation #MID-900515-47422 02/16/99
---	---

Siemens
Tom Kraus
Quot. Good for Ninety Days
Phone: 408-754-9224 Fax: 408-754-8287

SIH Account Exec: Matthew D. Smith
Telephone : (408) 522-1100
Fax : (408) 526-1222

References:

Product	Part	Qty	Unit	Year Price	Total
3 Pwr 1187 Hsk	285984	2500		\$29.00	\$72,500.00
Standard 1400	588203	3		\$587.00	\$1,761.00
Keypoint 800c 101103 500	737064	1		\$600.00	\$600.00
Total					\$74,861.00

Additional Comments

TIME 09:54 PM '99

81:31 FULL-PT-99M

MICROSOFT CORPORATION
One Microsoft Way
Redmond, WA 98073-0999

1-800-541-5551
Fax: 206-880-7008
<http://www.microsoft.com>

Microsoft

March 12, 1999

Mr. Erwin Kauf Zalka
Siemens Microsoft Informationssysteme AG
Heinrich-Rader Ring 1
D-53106 Paterborn
Germany

Via FAX # 011-49-5251-815149

Dear Mr. Zalka,

Here is the information you requested regarding US pricing of certain Microsoft products:

Microsoft SQL Server 7.0 Enterprise Edition, unlimited user license	\$33,999
Microsoft Windows NT Server, Enterprise Edition 4.0 incl 14 CALs	\$33,999
Windows NT Server 4.0 incl 5 CALs	\$469
Visual C++ Professional 5.0	\$189
5-yr maintenance for above software @ \$209/5yr	\$1,047.5

The prices quoted above are valid for the next 90 days. Please let me know if I can be of any further assistance.

Sincerely,



Sid Arora
Product Manager, Microsoft SQL Server
Applications Marketing

Microsoft Corporation. See myu.opera.com/microsoft.

Microsoft Corporation
One Microsoft Way
Redmond, WA 98073-4333

Tel: 425 882 8280
Fax: 425 935 7225
http://www.microsoft.com

Microsoft

March 17, 1999

Mr. Franz-Josef Dinkel
Siemens Nixdorf Informationssysteme AG
Helm-Neubert-Ring 1
D 33106 Paderborn
Germany

Via FAX # 011-49-5251-815149

Dear Mr. Dinkel,

Microsoft has received your request for permission to disclose results of TPC-C benchmarks conducted by SNI with the following system and Microsoft SQL Server 7.0, Enterprise Edition:

SNI Primergy 370-M4, 4-processor, Pentium III Xeon-based, 500 MHz, 2GB SDRAM
Test results: 21500 tpmC @ 52.5tpmC approximately

Microsoft hereby grants SNI permission to disclose these results and acknowledges that SNI has formally requested permission to do so in accordance with the license agreement for Microsoft SQL Server software.

Best Regards,


Sid Arora
Product Manager, Microsoft SQL Server
Applications Marketing

Microsoft Corporation. All rights reserved.

Appendix F - Attestation Letter



Information Paradigm



Certified Auditor

Sponsor: Franz-Josef Bathe

Siemens AG

Heinz-Nixdorf-Ring 1

D-33106 Paderborn, Germany

March 16, 1999

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

Platform: Siemens Primergy 870-40 c/s

Operating system: Microsoft Windows NT 4.0 Enterprise Edition

Database Manager: Microsoft SQL Server 7.0 Enterprise Edition

Transaction Manager: Tuxedo Version 6.4

The results were:

CPU's Speed	Memory	Disks	NewOrder 90% Response Time	tpmC
Server: Siemens Primergy 870-40				
4 x Pentium III Xeon (500 Mhz)	4.0 GB (2 MB L2 cache per processor)	193 x 9 GB 8 x 18 GB	.53 Seconds	23,570.33
Four Clients: Siemens Primergy 470 (Specification for each)				
2 x Pentium II (450 Mhz)	384 MB	1 x 4 GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC requirements for Revision 3.4 of the benchmark. The following verification items were given special attention:

- The transactions were correctly implemented
 - The database records were the proper size
- 1373 North Franklin Street • Colorado Springs, CO 80903-2527 • **Office:** 719/473-7555 • **Fax:** 719/473-7554

- The database was properly scaled and populated
- The ACID properties were met
- Input data was generated according to the specified percentages
- The transaction cycle times included the required keying and think times
- The reported response times were correctly measured.
- At least 90% of all delivery transactions met the 80 Second completion time limit
- All 90% response times were under the specified maximums
- The measurement interval was representative of steady state conditions
- The reported measurement interval was 30 minutes.
- One checkpoint was taken during the measurement interval
- Measurement repeatability was verified
- The 180 day storage requirement was correctly computed
- The system pricing was verified for major components and maintenance

Additional Audit Notes:

None

Respectfully Yours,



François Raab
President

Primergy 870-40

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