

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**

December 23, 1998

**First Edition**

First Edition December 23, 1998

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 © 1998 Siemens AG 1998. 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	22,349.47	25.84\$

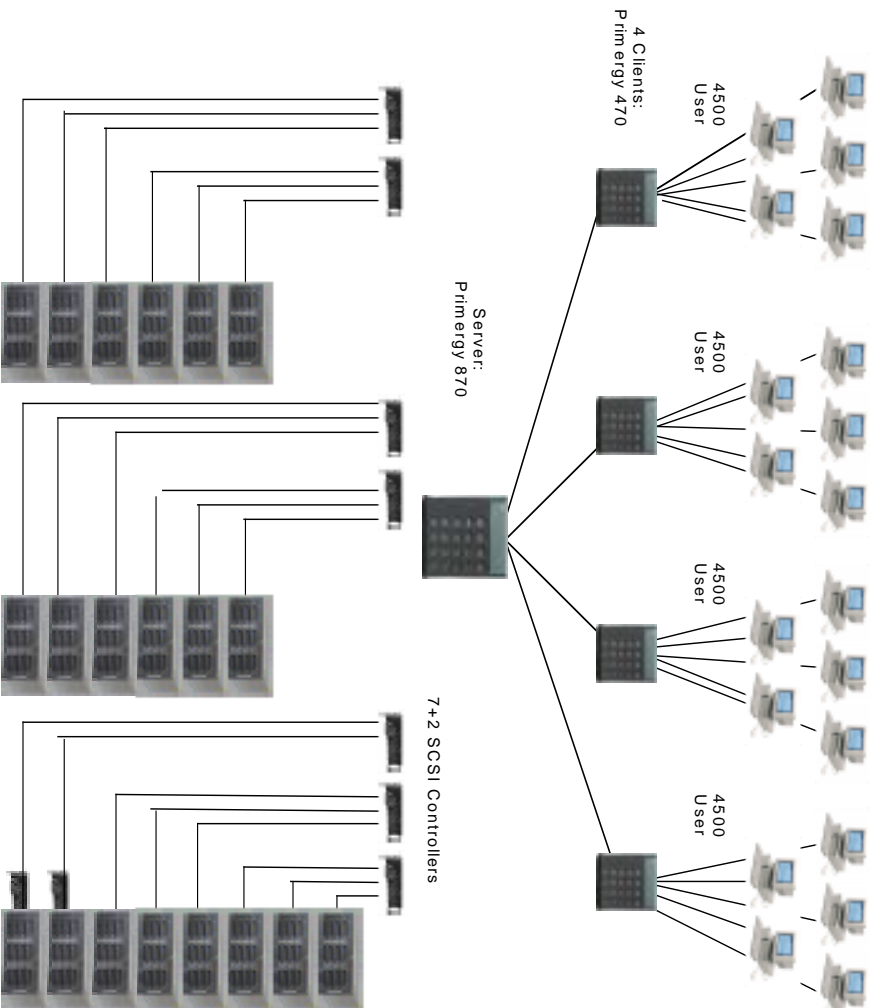
# SIEMENS

## Primergy 870-40 Client/Server

TPC-C REV 3.4 EXECUTIVE  
SUMMARY  
Page 1 of 2

Report Date: December 23, 1998

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
<b>\$ 577,459</b>	<b>22,349.47 tpmC</b>	<b>\$25.84/tpmC</b>	February 1999
Processors	Database Manager	Operating-System	Other Software
4 Intel Pentium® II Xeon 450 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
			<b>18,000</b>



18 Primergy S40 + 2 Primergy S40 with 24 disks for backup

System Components	Qty/Srv.	1 Primergy 870-40	Qty/Client	4 Primergy 470
Processors	4	Intel Pentium® II Xeon 450 MHz 2 MB SLC	2	Intel Pentium® II 450 MHz 512 KB SLC
Memory	4	GB	512	MB
Disk Controller	7+2	SCSI Controllers	1	SCSI Controller
Disk Drives	193	9 GB	1	4 GB
	8	18 GB		
Total GB of Storage	1	1737 GB	1	4 GB

# SIEMENS

## Primergy 870-40

TPC-C REV 3.4 EXECUTIVE  
SUMMARY  
Page 2 of 2

Client/Server

Report Date: December 23, 1998

Description	Part Number	Third Party	Brand	Pricing	Unit Price	Qty.	Extended Price	Svr Maint Price
<b>Base System</b>								
Pentium II Xeon Processor 450MHz/2MB	SNP-SY-K450A/101-P			1	\$11,284	1	\$11,284	
2. Memory Board	SNP-SY-F182A/E453-P			1	\$7,958	4	\$31,832	
Memory 512MB EDO-DIMM	SNP-SY-F182E1-P			1	\$798	1	\$798	
DAT Drive DDS-3 12GB, SE SCSI	SNP-SY-F1549E/K43-P			1	\$2,189	8	\$17,512	
Fast-Ether-Express-Proc/100+- Server (PCI)	SNP-SY-F1730E1-P			1	\$1,158	1	\$1,158	
FP 9GB/10K LVD-SCSI, Hd Plug	SNP-SY-F1899E/109-P			1	\$968	1	\$968	
Power Supply (add)	SNP-PS-F234E1-P			1	\$463	1	\$463	
Country Pack	S26391-K271-V310			1	\$25	1	\$25	
Country Pack	SNP-SY-F1899E/213-P			1	\$34	1	\$34	
Monitor MCM 1510	S26361-K494-V150			1	\$219	1	\$219	
<b>Server Hardware Subtotal</b>								
							\$64,364	
<b>RAID/Controller 3 Chan, 32Mb, LVD</b>								
RAID/Controller 3 Chan, 32Mb, LVD (10% spare)	S26361-D1124-V1			1	\$2,838	7	\$19,866	
FP 9GB/10K LVD-SCSI, Hd Plug	SNP-SY-F1899E/109-P			1	\$968	2	\$5,676	
FP 9GB/10K LVD-SCSI, Hd Plug (10% spare)	SNP-SY-F1899L/109-P			1	\$983	192	\$185,896	
Primergy 540 ES, 1 channel stack	SNP-SY-K4539V/201-P			1	\$2,362	18	\$42,516	
Primergy 502, 2-Host, 4-Chan, 64MB	S26361-K426-V592			1	\$7,394	2	\$14,788	
FP 18GB SCSI-3 WIDE-SCA, Hd Plg	S26361-F1322-E180			1	\$1,474	8	\$11,792	
Country Pack Primergy 502 (INT)	SNP-SY-F1478E/30-P			1	\$34	2	\$68	
SCSI SE/DE Converter - 3 channels	T26199-V2527-V205			1	\$1,032	1	\$1,032	
SCSI Cable HD-HD 5m	SNP-SY-F1947V/50-P			1	\$85	2	\$170	
							\$2,646	
							\$304,070	
<b>Maint. Servers/Storage</b>								
Primergy 470, FS Plg, 450	SNP-SY-K482V/714A			1	\$2,316	4	\$9,264	\$28,471
Plg 450MHz	SNP-SY-F172E/50-A			1	\$1,474	4	\$5,896	
Keyboard	S26391-K271-V310			1	\$25	4	\$100	
Country Pack International	SNP-SY-F1699B/153-A			1	\$34	4	\$136	
Memory 256 MB SDRAM 100MHz	SNP-SY-F1867E/504-P			1	\$1,221	4	\$4,884	
Memory 128 MB SDRAM 100MHz	SNP-SY-F1867E/503-A			1	\$589	8	\$4,712	
HD 4GB SCSI-3, Hd Plug	S26361-F1724-E40			1	\$505	4	\$2,020	
Fast-Ether-Express-Proc/100+- Server (PCI)	SNP-SY-F1888E/501-A			1	\$111	24	\$2,664	
Monitor MCM 1510	S26361-K494-V150			1	\$219	4	\$876	\$825
							\$303,552	\$7,579
<b>Client Hardware Subtotal</b>								
							\$3,999	
							\$28,999	
							\$32,998	\$10,475
<b>Server Software Subtotal</b>								
							\$3,999	
							\$28,999	
							\$3,236	
							\$12,000	\$9,000
							\$499	
							\$15,735	
<b>Client Software Subtotal</b>								
							\$28,75	2500
							\$28,75	\$71,875
							\$3,000	4
							\$499	1
							\$28,75	3
							\$73,390	
							\$521,109	
							\$56,350	
							\$22,349.47	
							\$25,84	

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

Five-Year Cost of Ownership \$577,459.00  
tpmC: 22,349.47  
\$/tpmC: \$25.84

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](mailto:pricing@tpc.org). Thank you.

Five-Year Cost of Ownership: \$577,459

tpmC Rating: 22,349.47

\$ / tpmC: 25.84

Note: The benchmark results and test methodology were audited by Francois Raab of Information Paradigm, Inc.

## Numerical Quantities Summary

<b>MQTh, computed Maximum Qualified Throughput</b>		<b>22,349.47 tpmC</b>
% throughput difference, reported & reproducibility runs		0.31 %
<b>Response Times (in seconds)</b>	<b>90th percentile</b>	<b>Average      Maximum</b>
- New-Order	0.91	0.61      77.34
- Payment	0.23	0.19      3.05
- Order-Status	0.41	0.29      3.95
- Delivery (interactive portion)	0.33	0.30      1.07
- Delivery (deferred portion)	0.94	0.57      3.64
- Stock-Level	2.49	2.07      3.79
- Menu	0.31	0.20      1.11
<b>Transaction Mix, in percent of total transactions</b>		
- New-Order		44.88 %
- Payment		43.04 %
- Order-Status		4.02 %
- Delivery		4.01 %
- Stock-Level		4.05 %
<b>Emulation Delay (in seconds)</b>		
		<b>Response Time      Menu</b>
- 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
<b>Keying/Think Times (in seconds)</b>		
	<b>Minimum</b>	<b>Average      Maximum</b>
- New-Order	18.00/0.00	18.01/12.04      18.06/120.51
- Payment	3.00/0.00	3.01/12.04      3.05/120.50
- Order-Status	2.00/0.00	2.01/10.04      2.06/100.50
- Delivery (interactive)	2.00/0.00	2.01/ 5.04      2.05/ 50.50
- Stock-Level	2.00/0.00	2.01/ 5.05      2.06/ 50.50
<b>Test Duration and Checkpointing</b>		
- Ramp-up time		71 minutes
- Measurement interval		30 minutes
- Number of checkpoints		1
- Checkpoint interval		30 minutes
- Transactions during measurement interval (all types)		1,494,099





# 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
<b>1. GENERAL ITEMS.....</b>	<b>13</b>
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
<b>2. CLAUSE 1 RELATED ITEMS - LOGICAL DATABASE DESIGN.....</b>	<b>17</b>
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
<b>3. CLAUSE 2 RELATED ITEMS - TRANSACTION AND TERMINAL PROFILES.....</b>	<b>19</b>
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>Queueing Mechanism</i> .....	20
<b>4. CLAUSE 3 RELATED ITEMS - TRANSACTION AND SYSTEM PROPERTIES.....</b>	<b>21</b>
4.1 <i>Atomicity</i> .....	21
4.2 <i>Consistency</i> .....	22
4.3 <i>Isolation</i> .....	22
4.4 <i>Durability</i> .....	23
<b>5. CLAUSE 4 RELATED ITEMS - SCALING AND DATABASE POPULATION .....</b>	<b>25</b>
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
<b>6. CLAUSE 5 RELATED ITEMS - PERFORMANCE METRICS AND RESPONSE TIME .....</b>	<b>29</b>

6.1 Measured tpmC.....	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
<b>7. CLAUSE 6 RELATED ITEMS - SUT, DRIVER, AND COMMUNICATION DEFINITION.....</b>	<b>37</b>
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
<b>8. CLAUSE 7 RELATED ITEMS - PRICING.....</b>	<b>39</b>
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
<b>9. CLAUSE 8 RELATED ITEMS - AUDIT.....</b>	<b>41</b>
<b>APPENDIX A - APPLICATION SOURCE CODE.....</b>	<b>43</b>
<b>APPENDIX B - DATABASE DETAILS.....</b>	<b>161</b>
<b>APPENDIX C - TUNABLE PARAMETERS AND OPTIONS.....</b>	<b>177</b>
<b>APPENDIX D - PRICING DETAILS.....</b>	<b>309</b>
180 Day Space Calculation.....	309
Price/Performance Spreadsheet.....	310
<b>APPENDIX E - PRICE QUOTATIONS.....</b>	<b>311</b>
<b>APPENDIX F - ATTESTATION LETTER.....</b>	<b>316</b>

# 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® II Xeon 450 MHz processors manufactured by Intel.

The processor power may be upgraded by three further Intel Pentium® II Xeon 450 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. Bathé  
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® II Xeon 450 MHz with 2 MB Second Level Cache
- 4 GB memory
- 7+2 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
- 512 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**

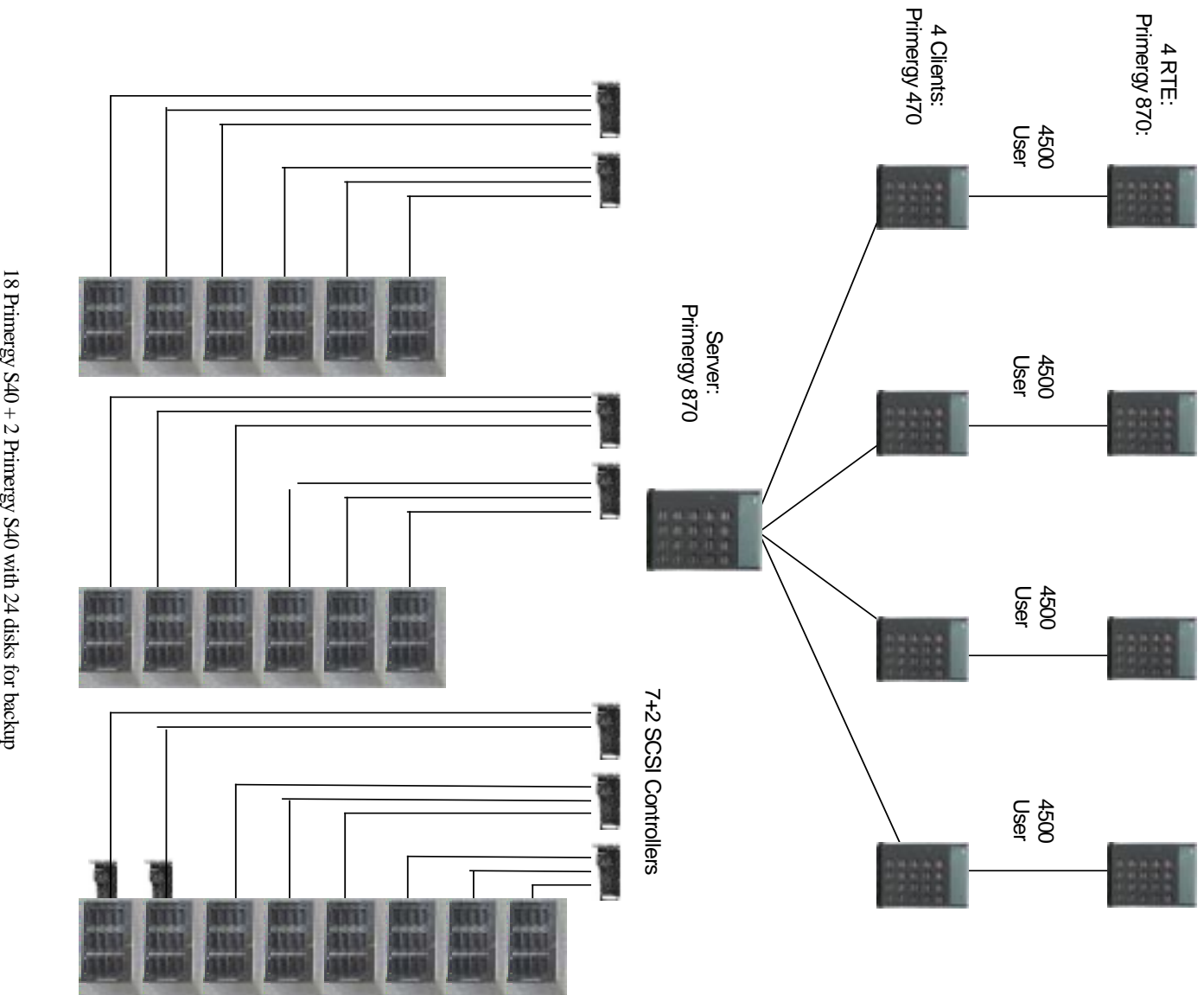
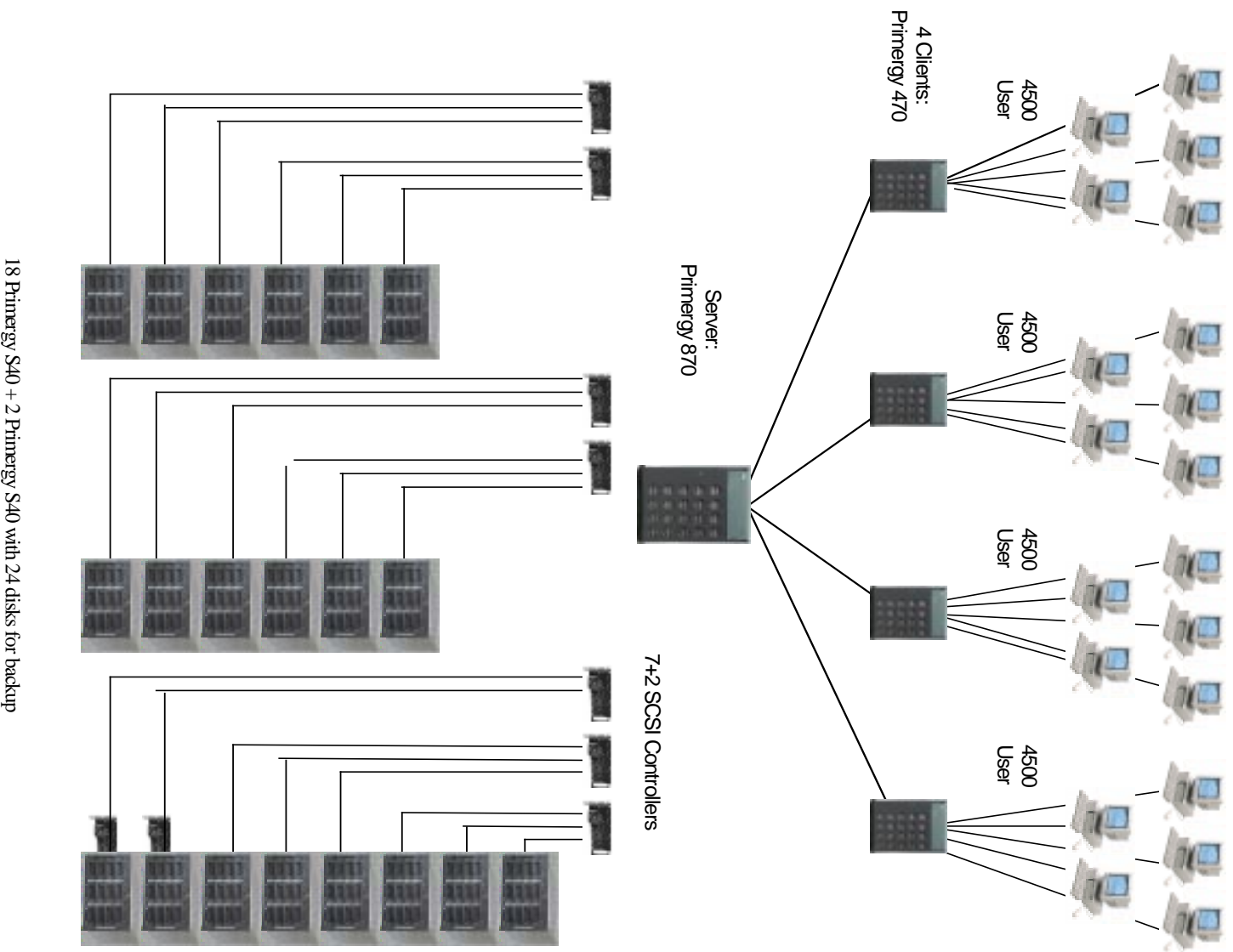


FIGURE 2: PRICED SYSTEM CONFIGURATION PRIMERGY 870-40





## 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:	45000	log
3	DAC1164P #2	32 x 9 GB	RAID0	H: O:	20000 10000	cs1 misc1
4	DAC1164P #3	32 x 9 GB	RAID0	I: P:	20000 10000	cs2 misc2
5	DAC1164P #4	32 x 9 GB	RAID0	J: Q:	20000 10000	cs3 misc3
6	DAC1164P #5	32 x 9 GB	RAID0	K: R:	20000 10000	cs4 misc4
7	DAC1164P #6	32 x 9 GB	RAID0	M: S:	20000 10000	cs5 misc5
8	DAC1164P #7	32 x 9 GB	RAID0	N: T:	20000 10000	cs6 misc6
9	DAC1164P #8	12 x 9 GB	RAID0	BACKUP 2 Y:	104195	NTFS
10	DAC1164P #8	12 x 9 GB	RAID0	BACKUP 1 X:	104195	NTFS

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	1.00%
	Average items per order	10.00
Payment	Home transactions	85.01%
	Remote transactions	14.99%
	Non-primary key access	60.04%
Order-Status	Non-primary key access	59.73
Delivery	Skipped transactions	0
Transaction Mix	New-Order	44.88 %
	Payment	43.04 %
	Order-Status	4.02 %
	Delivery	4.01 %
	Stock-Level	4.05 %

### 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 be disclosed along with a description of how the ACID requirements were met. This includes disclosing which case was followed for the execution of Isolation Test 7. [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 Primergy 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.
- A 35 minutes test was run (5 min. ramp up and 30 min. measurement)
- After 5 minutes in measurement a checkpoint was performed
- After checkpoint has completed, we pulled off one of the log disks. As we use hardware-mirrored diskpairs with the SCSI-controller, execution continued
- After additional 2 min we powered of the server to emulate the loss of memory. After server system reboot, SQL-Server starts with recovering the database tpcc. 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 24 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.
- A 35 minutes test was run (5 min. ramp up and 30 min. steady state)
- No checkpoint was performed.
- After 5 min in 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 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.]*

The database for the Primergy 870-40 system was scaled for 1920 warehouses. 120 warehouses were deleted prior to performing the measurement runs.

Table	Number of Records
Warehouse	1920
District	19,200
Customer	57,600,000
History	57,600,000
Order	57,600,000
New-Order	17,280,000
Order-Line	575,998,743
Stock	192,000,000
Item	100,000
Deleted Warehouses	120

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*  
**Distribution of Tables and Log** *for the tested and priced systems. [Clause 8.1.5.2]*

**FIGURE 1: LOGICAL ORGANIZATION OF THE DATABASE**

	<b>device</b>	<b>raw size</b>	<b>use</b>
D:	MSSQL70_tproc_root	10 MB	root
L:	MSSQL70_tproc_log	45,000 MB	Log
O:	MSSQL70_misc1	9,200 MB	Warehouse, District, Item, New Order, History, Order, Orderline
P:	MSSQL70_misc2	9,200 MB	Warehouse, District, Item, New Order, History, Order, Orderline
Q:	MSSQL70_misc3	9,200 MB	Warehouse, District, Item, New Order, History, Order, Orderline
R:	MSSQL70_misc4	9,200 MB	Warehouse, District, Item, New Order, History, Order, Orderline
S:	MSSQL70_misc5	9,200 MB	Warehouse, District, Item, New Order, History, Order, Orderline
T:	MSSQL70_misc6	9,200 MB	Warehouse, District, Item, New Order, History, Order, Orderline
H:	MSSQL70_cs1	19,100 MB	Stock, Customer
I:	MSSQL70_cs2	19,100 MB	Stock, Customer
J:	MSSQL70_cs3	19,100 MB	Stock, Customer
K:	MSSQL70_cs4	19,100 MB	Stock, Customer
M:	MSSQL70_cs5	19,100 MB	Stock, Customer
N:	MSSQL70_cs6	19,100 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 subtain 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 22,349.47 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	0.61	77.34	0.91
Payment	0.19	3.05	0.23
Order-Status	0.29	3.95	0.41
Interactive Delivery	0.30	1.07	0.33
Deferred Delivery	0.57	3.64	0.94
Stock-Level	2.07	3.79	2.49
Menu	0.20	1.11	0.31

### 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	18.06	18.00
Payment	3.01	3.05	3.00
Order-Status	2.01	2.06	2.00
Delivery	2.01	2.05	2.00
Stock-Level	2.01	2.06	2.00

Think Times			
Type	Average	Maximum	Minimum
New-Order	12.04	120.51	0.00
Payment	12.04	120.50	0.00
Order-Status	10.04	100.50	0.00
Delivery	5.04	50.50	0.00
Stock-Level	5.05	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]

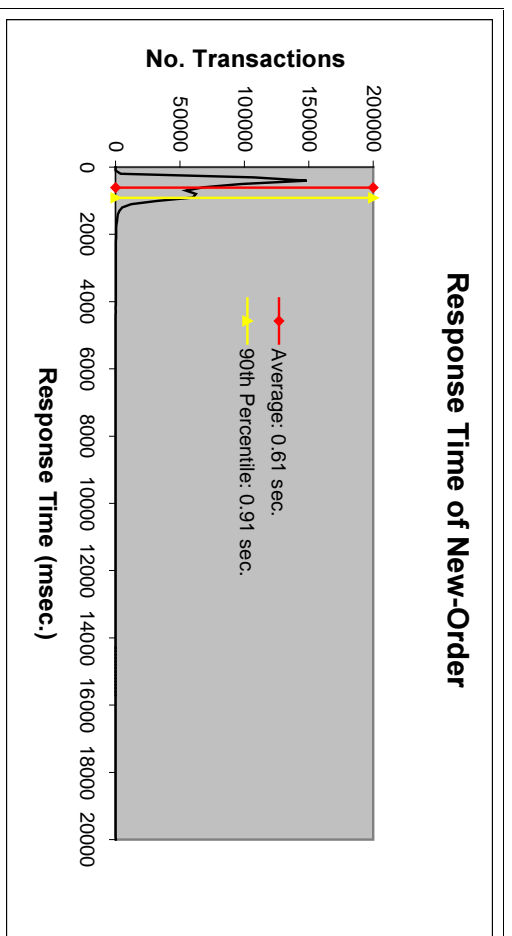
The performance curve for response times versus throughput (see Clause 5.6.2) must be reported for the New-Order transaction. [Clause 8.1.6.5]

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

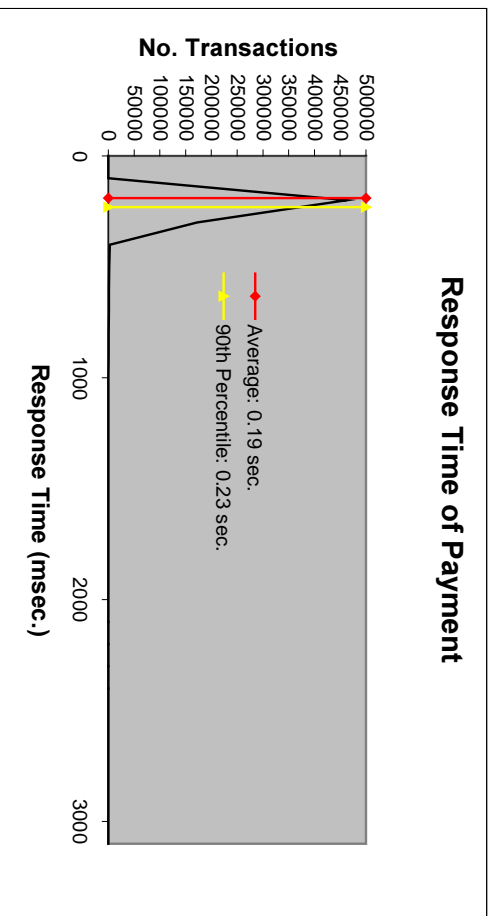
Keying Time frequency distribution curves (see Clause 5.6.4) must be reported for each transaction type. [Clause 8.1.6.7]

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]

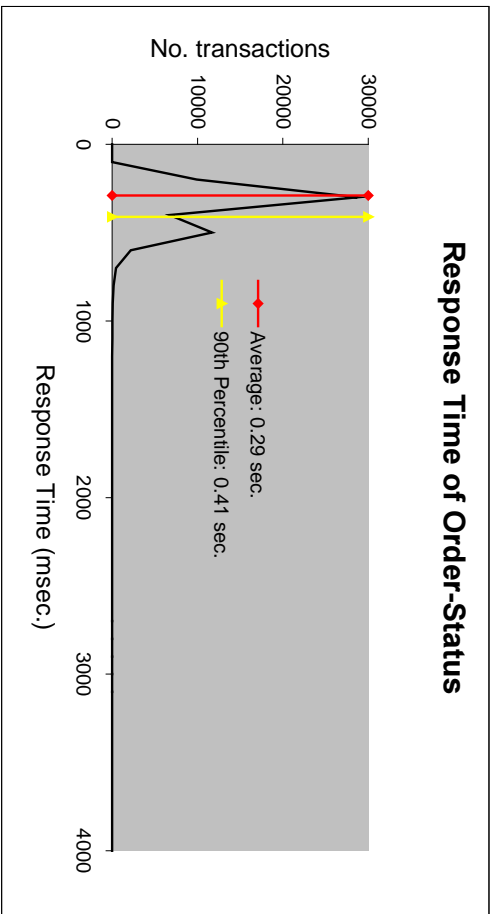
**FIGURE 1: NEW-ORDER RESPONSE TIME DISTRIBUTION**



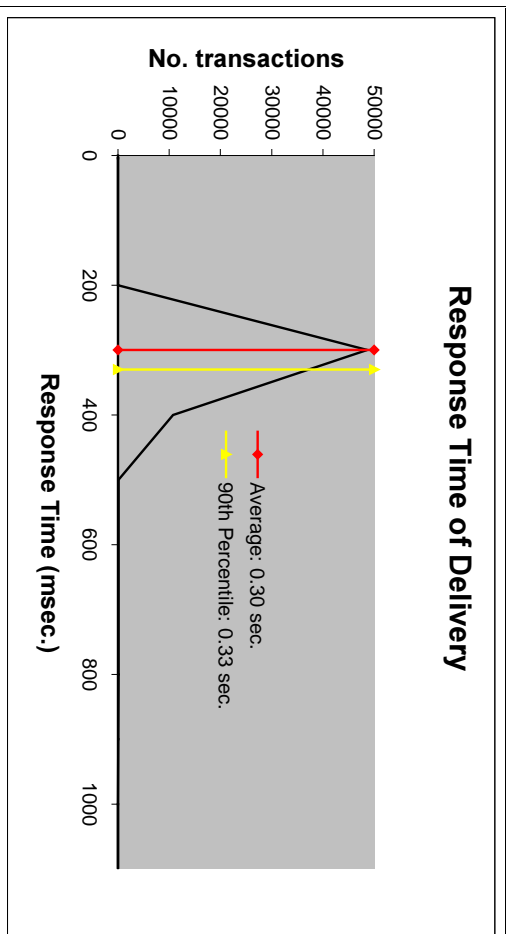
**FIGURE 2: PAYMENT RESPONSE TIME DISTRIBUTION**



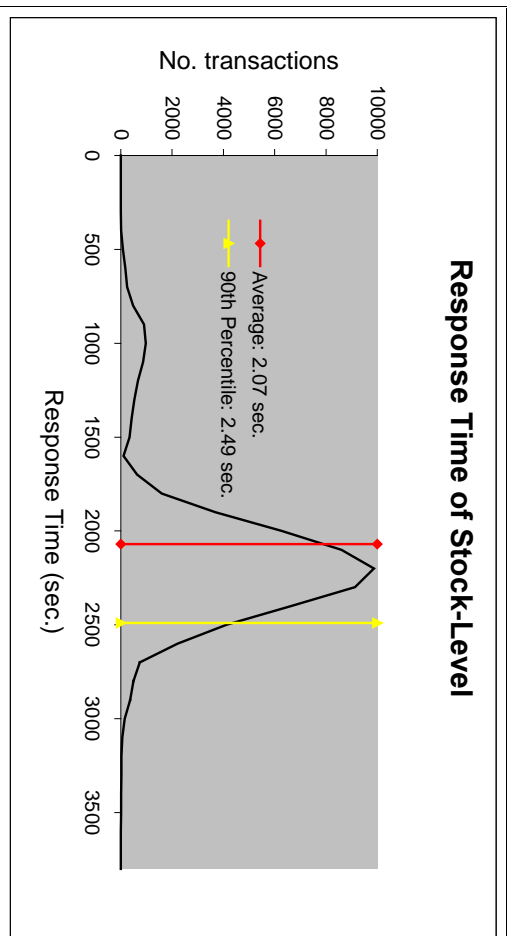
**FIGURE 3: ORDER-STATUS RESPONSE TIME DISTRIBUTION**



**FIGURE 4: DELIVERY RESPONSE TIME DISTRIBUTION**



**FIGURE 5: STOCK-LEVEL RESPONSE TIME DISTRIBUTION**



**FIGURE 1: RESPONSE TIME VERSUS THROUGHPUT**

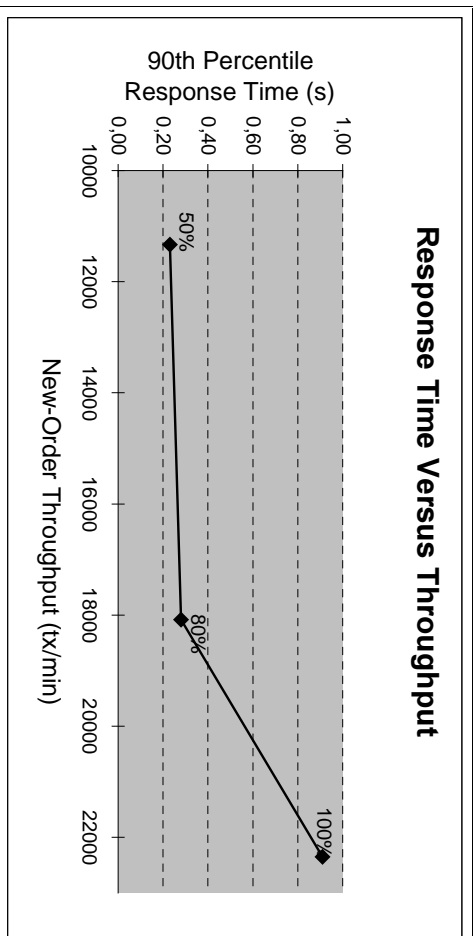




FIGURE 1: NEW-ORDER THINK TIME DISTRIBUTION

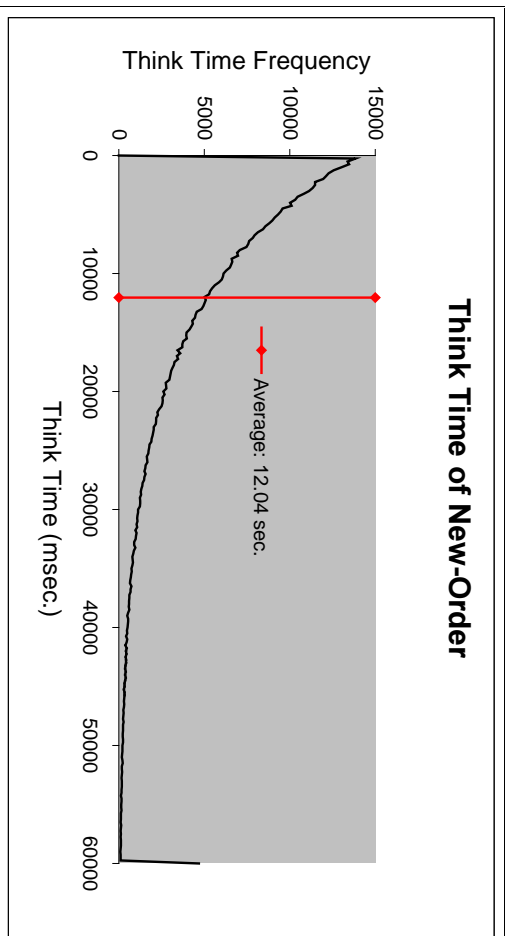
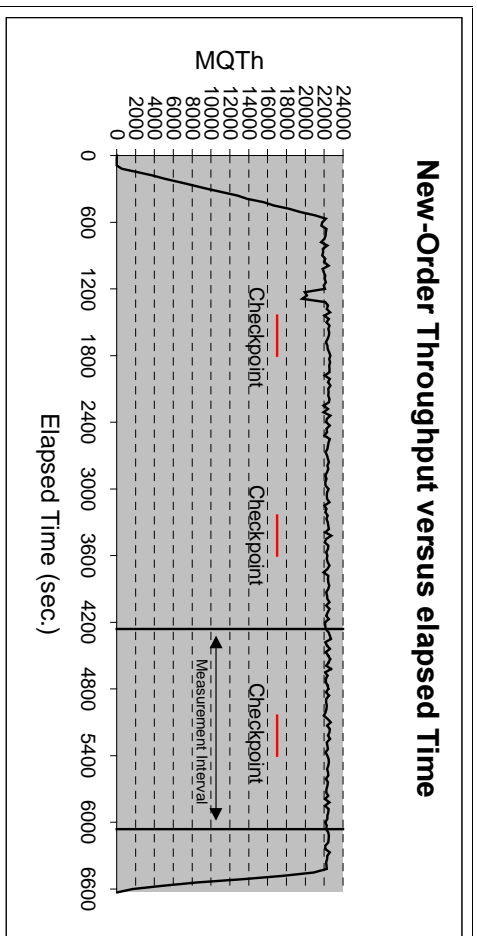


FIGURE 1: THROUGHPUT VERSUS ELAPSED TIME



### 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 fpmC 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 0.31% of the reported tpmC.

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

	<b>tpmC</b>
reported test	22.349,47
reproducibility test	22.280,67

## 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]*

	<b>Percentage</b>
New-Order	44.88 %
Payment	43.04 %
Order-Status	4.02 %
Delivery	4.01 %
Stock-Level	4.05 %

## **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 were two checkpoints before and one during the measurement interval. The third checkpoint occurred 720 seconds after the start of the measurement interval. The checkpoint interval was 1800 seconds. The duration of the checkpoint during measurement was 6:09 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 Benchmark 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 18,000 workstations. Each client has 4500 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 February 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 22,349.47 tpmC with Microsoft SQL Server 7.0 Enterprise Edition with a 5-year system price of \$577,459. The respective price/performance for the Primergy 870-40 is \$25.84/tpmC. The priced Primergy 870-40 will be available as of February 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 Primergy 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.

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. Bathe  
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
                        //insuficient memory to process 90th percentile report
#define ERR_CANNOT_OPEN_RESULTS_FILE 1005
                        //Cannot open delivery results file delilog.

typedef struct _RPTLINE
{
    SYSTEMTIME    start;
                        //delilog report line start time
    SYSTEMTIME    end;
                        //delilog report line end time

```

```

    int           response;
                        //delilog report line time
    int           delivery took in milliseconds
    int           w_id;
                        //delilog report line warehouse
    int           id for delivery
    int           o_carrier_id;
                        //delilog report line carier id for
    int           delivery
    int           items[10];
                        //delilog report line delivery
} RPTLINE, *PRPTLINE;

//error message structure used in ErrorMessage API
typedef struct _SERRORMSG
{
    int           iError;
                        //error id of
    char          szMsg[80];
                        //message to sent to
} 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      iTotResponse;
    int      iLines;
    double  fAverage;
    char    szDelivery[128];

```

```

ResetLogFile();

iTotResponse = 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++;
        iTotResponse += 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)iTotResponse /
(double)iLines)/(double)1000;
        printf("Total Deliveries:      %10.0f\n",
(float)iLines);
        printf("Total Response Times: %10.3f\n",
((float)iTotResponse/(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   iTTotal;
    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;
        }
    }

    iTTotalBuckets = iMaxSeconds + 1;

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

    iBucketSize = iTTotalBuckets * sizeof(short);

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

    ZeroMemory(psBuckets, iBucketSize);

    iTTotal = 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]++;
        iTTotal++;
    }
}

i90thPercent = iTTotal * .9;

for(i=0, iTTotal = 0.0; iTTotal < i90thPercent; iTTotal +=
(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 to long cannot be a valid delivery
item
    if ( i >= 128 )
        return TRUE;

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

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

```

```

*/
static BOOL ParseReportLine(char *szLine, PRPTLINE pRptLine)
{
    int i;

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

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

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

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

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

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

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

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

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

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

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

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

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

```

```

szLine++;

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

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

return FALSE;
}

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

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

    pTime->wYear = atoi(szDate);

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

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

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

    return FALSE;
}

```

```

}

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

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

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

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

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

    return FALSE;
}

/* FUNCTION: void ErrorMessage(int iError)
*

```

```

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

```

```

static void ErrorMessage(int iError)
{
    int i;

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

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

```

```

/* FUNCTION: BOOL GetParameters(int argc, char *argv[])
*
* PURPOSE:   This function parses the command line passed in to the
delivery executable, initializing
*           and filling in global variable parameters.

```

```

*
* ARGUMENTS: int          argc number of command line arguments
passed to delivery
*           char *argv[] array of command line
argument pointers
*
* RETURNS:   BOOL FALSE parameter read successfull
             TRUE user has requested
parameter information screen be displayed.
*
* COMMENTS:  None
*/

```

```

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

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

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

```

```

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

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

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

    return;
}

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

```

```

*/
static void PrintHeader(void)
{
    cls();

    printf("*****\n");
    printf("*
printf("* Microsoft SQL Server 6.5
printf("*
printf("* HTML TPC-C BENCHMARK KIT: Delivery Report
printf("* Version %d.%2.2d.%3.3d
*\n", versionMS, versionMM, versionLS);
    printf("*
printf("*****\n\n");
;

    return;
}

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

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

    hConsole = GetStdHandle(STD_OUTPUT_HANDLE);

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

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

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

```

```

    GetConsoleScreenBufferInfo( hConsole, &csbi );

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

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

    return;
}

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

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

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

/* 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
*               -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;
}

```

```

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

return;
}

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

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

    deadlock_count = 0;

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

            if (dbrpcexec(dbproc) == SUCCEED)
            {

```

```

while ((rc = dbresults(dbproc)) !=
NO_MORE_RESULTS) && (rc != FAIL))
    {
        while ((rc = dbnextrow(dbproc))
!= NO_MORE_ROWS) && (rc != FAIL))
            {
                for (i=0;i<10;i++)
                {
                    if(pData=dbdata(dbproc, i+1))
                        pDelivery->
>o_id[i] = *((DBINT *)pData);
                    else
                        pDelivery->
>o_id[i] = 0;
                }
            }
        }
    }
    if ( !SQLDetectDeadlock(dbproc) )
        break;
    deadlock_count++;
    Sleep(10 * deadlock_count);
}
GetLocalTime(&trans_end);

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

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

if ( bFlush )
    fflush(fpLog);

```



```

        return ERR_SUCCESS;
    }
}
/* FUNCTION: BOOL SQLDetectDeadlock(DBPROCESS *dbproc)
*
* PURPOSE:   This function is used to check for deadlock conditions.
*
* ARGUMENTS: DBPROCESS          *dbproc          DBPROCESS to check
*
* RETURNS:   BOOL    FALSE          No lock
            condition present
*
*           Lock condition detected          TRUE
*
* 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 *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.
    DBSETLHOST(login, app);
    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(PECBINFO));
    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)_vsnprintf( buf+ len, sizeof( buf) - len - 1, format,
args);
    buf[sizeof( buf )- 1]= '\0';
    va_end( args );
    userlog( buf );
}

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

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

    return;
}

/* FUNCTION: int err_handler(DBPROCESS *dbproc, int severity, int
dberr, int oserr, char *dberrstr, char *oserrstr)
*
* PURPOSE: This function handles DB-Library errors
*
* ARGUMENTS: DBPROCESS *dbproc DBPROCESS
id pointer

```

```

*          int          severity
severity of error
*          int          dberr
error id
*          int          oserr
operating system specific error code
*          char         *dberrstr
printable error description of dberr
*          char         *oserrstr
printable error description of oserr
* RETURNS:          int          INT_CONTINUE
continue if error is SQLETIME else INT_CANCEL action
* COMMENTS:  None
*/

```

```

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

    pEcbInfo = NULL;

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

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

    if ( pEcbInfo && pEcbInfo->bFailed )
    {

```

```

        bError == FALSE;
        return INT_CANCEL;
    }

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

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

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

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

        fclose(fp);
    }

    return INT_CANCEL;
}

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

```

```

*
*                               INT_CANCEL
*
*           cancel operation
*
* * COMMENTS:  This function also sets the dead lock dbproc variable if
* necessary.
*
*/

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

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

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

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

```

```

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

    if ( pEcbInfo )
        pEcbInfo->bFailed = TRUE;

    bError = TRUE;

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

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

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

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

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

```

```

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

flush when written.

return FALSE;
}

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

static void PrintParameters(void)
{
    TMLog("Performance Tuning Corporation Tuxedo Kit");
    TMLog(" www.perftuning.com (281) 251-3495 ");
    TMLog("Delivery: -S Server [-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\DBLIB
!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
DBLIB           = $(SQL_LOC)
DBLIBINC        = $(DBLIB)\include
DBLIBDIR        = $(DBLIB)\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
$(LDEBUG) /def:"$(LINK32_DEF)" /out:"$(OUTDIR)\tpcc.dll"

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

ALL: $(OBJDIR)\. $(OUTDIR)\. $(OUTDIR)\tpcc.dll
$(OUTDIR)\Neworder.exe $(OUTDIR)\Payment.exe \
$(OUTDIR)\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
*             int      -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
*             int      -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
                                {
                                    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
 *          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 = SQLAllocStmnt((*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;

        SQLFreeStmnt((*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;
        SQLFreeStmnt((*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;

        SQLFreeStmnt((*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");

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
*
* RETURNS: BOOL FALSE if successfull
* TRUE if an error occurs
*
* COMMENTS: None
*
*/

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

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

BOOL SQLDetectDeadlock(DBPROCESS *dbproc)

```

```

{
    PECBINFO      pEcbInfo;

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

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

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

static void UtilStrCpy(char *pDest, char *pSrc, int n)

```

```

{
    strncpy(pDest, pSrc, n);
    pDest[n] = '\0';

    return;
}

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

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

    pEcbInfo = NULL;

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

    if ( !(pEcbInfo = (PECBINFO)dbgetuserdata(dbproc)) )

```



```

{
    pECB = gpECB;
    iTermId = 0;
    iSyncId = 0;
}
else
{
    pECB = pEcbInfo->pECB;
    iTermId = pEcbInfo->iTermId;
    iSyncId = pEcbInfo->iSyncId;
}

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

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

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

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

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

    fclose(fp);
}

return INT_CANCEL;
}

/* FUNCTION: int msg_handler(DBPROCESS *dbproc, DBINT msgno, int
msgstate, int severity, char *msgtext)
*
* PURPOSE: This function handles DB-Library SQL Server error
messages

```

```

*
* ARGUMENTS: DBPROCESS *dbproc DBPROCESS
id pointer DBINT msgno
*
message number DBINT msgno
*
message state int msgstate
*
message severity int severity
*
char *msgtext
printable message description
*
* RETURNS: int INT_CONTINUE
continue if error is SQLETIME else INT_CANCEL action INT_CANCEL
*
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];
char     //SQL server name
szDatabase[32];
char     //tpcc database name
szUser[32];
char     //user name
szPassword[32];
int      spId;

#ifdef LOCAL_ALLOC
ORDER_STATUS_DATA OrderStatusData;
#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: 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
*           -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
*           -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.
*
* iTermId int
terminal id of browser
*
* iSyncId int
sync id of browser
*
* *dbproc DBPROCESS
connection db process id
*
* *pOrderStatus ORDER_STATUS_DATA
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
successful
*
* 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) == SUCCEEDED)
        {
            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) == SUCCEEDED)
            {
                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);
2)
        if (pData=dbdata (dbproc,
            pOrderStatus-
>OlOrderStatusData[i].ol_i_id = *(DBINT *) pData);
3)
            if (pData=dbdata (dbproc,
                pOrderStatus-
>OlOrderStatusData[i].ol_quantity = (*(DBSMALLINT *) pData);
4)
                if (pData=dbdata (dbproc,
                    dbconvert (dbproc,
SQLNUMERIC, pData, dbdatlen(dbproc,4), SQLFLT8, (BYTE *)&pOrderStatus-
>OlOrderStatusData[i].ol_amount, 8);
5)
                    if (pData=dbdata (dbproc,
                        {
                            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));
3)
                                    if (pData=dbdata (dbproc,
                                        UtilStrCpy (pOrderStatus->c_first, pData, dbdatlen(dbproc,3));
4)
                                            if (pData=dbdata (dbproc,

```

```

        UtilStrCpy (pOrderStatus->c_middle, pData, dbdatlen(dbproc, 4));
5)
            if (pData=dbdata (dbproc,
                {
                    datetime =
*(DBDATETIME *) pData);
                    dbdatecrack (dbproc, &pOrderStatus->o_entry_d, &datetime);
6)
                }
                if (pData=dbdata (dbproc,
                    pOrderStatus-
>o_carrier_id = (*(DBSMALLINT *) pData);
7)
                    if (pData=dbdata (dbproc,
                        dbconvert (dbproc,
SQLNUMERIC, pData, dbdatlen(dbproc,7), SQLFLT8, (BYTE *)&pOrderStatus-
>c_balance, 8);
8)
                        if (pData=dbdata (dbproc,
                            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
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
 *          char *database SQL
server name
 *
 *          char *user
server database
 *
 *          char *password
 *
 *          char *app
 *          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 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");

        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

/* 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("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
*             int      -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
*             int      -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) == SUCCEED)
        {
            dbrpcparam(dbproc, NULL, 0, SQLINT2, -1, -1,
(BYTE *) &pStockLevel->w_id);
            dbrpcparam(dbproc, NULL, 0, SQLINT1, -1, -1,
(BYTE *) &pStockLevel->d_id);
            dbrpcparam(dbproc, NULL, 0, SQLINT2, -1, -1,
(BYTE *) &pStockLevel->thresh_hold);

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

                            if (pData=dbdata(dbproc, 1))
                                pStockLevel->low_stock = *((long *) pData);
                        }
                    }
                }
            }
            if (SQLDetectDeadlock(dbproc))
            {
                pStockLevel->num_deadlocks++;
                sprintf(printbuf, "deadlock: retry:
%d", pStockLevel->num_deadlocks);
                Sleep(10 * tryit);
            }
            else
            {
                strcpy(pStockLevel->execution_status,
"Transaction committed.");
                return TRUE;
            }
        }

        // If we reached here, it means we quit after MAX_RETRY
        deadlock_retry
        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 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");

        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");

    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
*
* RETURNS: BOOL FALSE if successfull
TRUE if an error occurs
*
* COMMENTS: None
*/

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

```

```

static BOOL SQLCloseConnection(EXTENSION_CONTROL_BLOCK *pECB,
DBPROCESS *dbproc)
{
    if (dbclose(dbproc) == FAIL)
        return TRUE;
    return FALSE;
}

#endif

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

BOOL SQLDetectDeadlock(DBPROCESS *dbproc)
{
    PECBINFO pEcbInfo;

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

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

```

```

        va_end( args );
        userlog( buf );
    }

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

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

    return;
}

/* FUNCTION: int err_handler(DBPROCESS *dbproc, int severity, int
 * dberr, int oserr, char *dberrstr, char *oserrstr)
 *
 * PURPOSE:   This function handles DB-Library errors
 *
 * ARGUMENTS: DBPROCESS      *dbproc          DBPROCESS
 *            id pointer
 *            int            severity
 *            severity of error
 *            int            dberr
 *            error id
 *            int            oserr
 *            operating system specific error code
 *            char          *dberrstr
 *            printable error description of dberr
 *            char          *oserrstr
 *            printable error description of oserr
 *
 * RETURNS:   int            INT_CONTINUE
 *            continue if error is SQLETIME else INT_CANCEL action

```

```

 *
 * COMMENTS:  None
 *
 */

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

    pEcbInfo = NULL;

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

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

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

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

```

```

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

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

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

        fclose(fp);
    }

    return INT_CANCEL;
}

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

int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int
severity, char *msgtext)
{
    PECBINFO pEcbInfo;
    EXTENSION_CONTROL_BLOCK *pECB;

```

```

FILE *fp;
SYSTEMTIME systemTime;
char szTmp[256];
int iTermId;
int iSyncId;

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

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

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

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

    if ( pEcbInfo )
        pEcbInfo->bFailed = TRUE;

    bError = TRUE;

    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_WEBDLL
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

    //TERMINID= key in the HTTP string.
    void       (*Delete)(EXTENSION_CONTROL_BLOCK *pECB, int
id); //API to free resources used by a terminal array entry
} TERM;

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

//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 WINAPIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call, LPVOID
lpReserved);
static void DeliveryDisconnect(void *ptr);
static BOOL IsValidTermId(int TermId);
BOOL ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd, int
*pFormId, int *pTermId, int *pSyncId);
void NewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId);
void PaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId);
void DeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId);
void OrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId);
void StockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId);
void Exitcmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId);
void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId);
void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId);
void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId);
void ClearCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId);
void MenuCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId,
int iSyncId);
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, UWORD 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;

```

```

    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;
#ifdef USE_ODBC
    short                o_commit_flag;
    TIME_STAMP_STRUCT   o_entry_d;
#else
    DBDATE_REC          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
    short                h_date;
    TIME_STAMP_STRUCT   h_date;
#else
    DBDATE_REC          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
    short                c_since;
    TIME_STAMP_STRUCT   c_since;
#else
    DBDATE_REC          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
    short                ol_delivery_d;
    TIME_STAMP_STRUCT   ol_delivery_d;
#else
    DBDATE_REC          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;
}

```

## 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.
*

```

```

*****/
#ifdef _HTTPEXT_H_
#define _HTTPEXT_H_

#include <windows.h>

#ifdef __cplusplus
extern "C" {
#endif

```

```

#define HSE_VERSION_MAJOR      1      // major version of this
spec
#define HSE_VERSION_MINOR      0      // minor version of this
spec
#define HSE_LOG_BUFFER_LEN     80
#define HSE_MAX_EXT_DLL_NAME_LEN 256

typedef LPVOID HCONN;

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

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

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

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

//
// These are Microsoft specific extensions
//

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

//
// passed to GetExtensionVersion
//

typedef struct _HSE_VERSION_INFO {
    DWORD dwExtensionVersion;
    CHAR lpszExtensionDesc[HSE_MAX_EXT_DLL_NAME_LEN];
} HSE_VERSION_INFO, *LPHSE_VERSION_INFO;

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

```

```

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

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

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

    LPSTR lpszContentType; // Content type of client data

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

LPVOID

lpvBuffer,
LPDWORD lpdwSize );

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

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

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

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

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

```

```

// the following type declarations is for the server side
typedef BOOL (WINAPI * PFN_GETEXTENSIONVERSION)( HSE_VERSION_INFO *pVer
);
typedef DWORD (WINAPI * PFN_HTTPEXTENSIONPROC )( EXTENSION_CONTROL_BLOCK
*pECB );

#ifdef __cplusplus
}
#endif

#endif // end definition _HTTPTEXT_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     StartWWWebService(void);
static BOOL     StopWWWebService(void);
static void     UpdateDialog(HWND hDlg);

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

    hInst = hInstance;

    InitCommonControls();

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

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

    return 0;
}

BOOL CALLBACK UpdatedDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM
lParam)
{
    switch(uMsg)
    {
        case WM_INITDIALOG:
            if ( lParam == 1 )
                SetDlgItemText(hwnd, IDC_RESULTS, "HTML
TPCC Installation Successfull");
            else

```

```

        SetDlgItemText(hwnd, IDC_RESULTS, "HTML
TPCC Registry Updated");
        return TRUE;
    case WM_COMMAND:
        if ( wParam == IDOK )
            EndDialog(hwnd, TRUE);
        break;
    default:
        break;
    }
    return FALSE;
}

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

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

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

            bLog                = FALSE;
            iThreads            = 4;
            iMaxWareHouse       = 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) occured
when creating tpcc.dll", NULL, MB_ICONSTOP | MB_OK);
            EndDialog(hwnd, 0);
            return TRUE;
        }
        SetDlgItemText(hDlg, IDC_STATUS, "Updating
Registry.");
        SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);

        UpdateDialog(hDlg);

        WriteRegistrySettings(szDllPath);

        Sleep(100);

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

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

static void ReadRegistrySettings(void)
{
    HKEY    hKey;

```

```

    DWORD    size;
    DWORD    type;
    char    szTmp[256];

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

        bLog = FALSE;
        if ( RegQueryValueEx(hKey, "LOG", 0, &type, szTmp, &size)
== ERROR_SUCCESS )
            if ( !stricmp(szTmp, "ON") )
                bLog = TRUE;

        iThreads = 4;
        size = sizeof(szTmp);
        if ( RegQueryValueEx(hKey, "NumberOfDeliveryThreads", 0,
&type, szTmp, &size) == ERROR_SUCCESS )
            iThreads = atoi(szTmp);
        if ( iThreads == 0 )
            iThreads = 4;

        iMaxWareHouse = 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);
    StartWWWService();
}

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, 0) == 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
#ifdef 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 AFX_RESOURCE_DLL || defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif // _WIN32

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

IDD_DIALOG1 DIALOGEX 0, 0, 234, 174
STYLE DS_MODALFRAME | DS_CENTER | WS_MINIMIZEBOX | WS_POPUP | WS_CAPTION
|
    WS_SYSMENU
CAPTION "TPCC Web Client Installation Utility"
FONT 8, "MS Sans Serif"
BEGIN
    EDITTEXT        ED_MAXWARE,199,37,21,12,ES_NUMBER,WS_EX_RTLREADING
    CONTROL         "",BN_LOG,"Button",BS_AUTOCHECKBOX | BS_LEFTTEXT |
        BS_LEFT | BS_VCENTER | WS_TABSTOP,205,51,15,13,
        WS_EX_STATICEDGE
    EDITTEXT        ED_THREADS,205,66,15,12,ES_NUMBER,WS_EX_RTLREADING
    EDITTEXT        ED_MAXCONNECTION,186,80,34,12,ES_NUMBER,WS_EX_RTLREADING
    EDITTEXT        ED_IIS_MAX_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"

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

IDR_TPCCDLL1       TPCCDLL DISCARDABLE     "tpcc.dll"

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

VS_VERSION_INFO VERSIONINFO
FILEVERSION 0,3,0,2
PRODUCTVERSION 0,3,0,2
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x40004L
FILETYPE 0x1L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904b0"

```

```

BEGIN
    VALUE "CompanyName", "Microsoft\0"
    VALUE "FileDescription", "install\0"
    VALUE "FileVersion", "0, 3, 0, 2\0"
    VALUE "InternalName", "install\0"
    VALUE "LegalCopyright", "Copyright © 1996\0"
    VALUE "OriginalFilename", "install.exe\0"
    VALUE "ProductName", "Microsoft install\0"
    VALUE "ProductVersion", "0, 3, 0, 2\0"
END
END
BLOCK "VarFileInfo"
BEGIN
    VALUE "Translation", 0x409, 1200
END
END
#endif // !_MAC

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

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

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

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

//{{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 <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>

static TPINIT *tpinf;
static DWORD  TLSIsTpInitedKey;
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    iQSlots = 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>"
ACTION="tpcc.dll" METHOD="GET">"
TYPE="hidden" NAME="STATUSID" VALUE="0">"
TYPE="hidden" NAME="ERROR" VALUE="0">"
TYPE="hidden" NAME="FORMID" VALUE="1">"
TYPE="hidden" NAME="TERMINID" VALUE="-2">"
TYPE="hidden" NAME="SYCNID" VALUE="0">"
<INPUT NAME="w_id" SIZE=4<BR>"
<INPUT NAME="d_id" SIZE=2<BR>"
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 WINAPI 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 WINAPI 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. TERMID= from http
command line
*
            int
iSyncId      sync id of calling browser
            EXTENSION_CONTROL_BLOCK      *pECB
*
            structure pointer to passed in internet
*
            service information.
* RETURNS:   None
*
* COMMENTS:  None
*
*/

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

    UNUSEDPARAM(iFormId);

}

/* FUNCTION: void DeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int
iFormId, int iTermId, int iSyncId)
*

```

```

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

void DeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId)
{
    // WriteZString(pECB, MakeDeliveryForm(iTermId, iSyncId, TRUE) );
    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. TERMID= from http
command line
*
            int
iSyncId      sync id of calling borwser
            EXTENSION_CONTROL_BLOCK      *pECB
*
            structure pointer to passed in internet
*
            service information.
* RETURNS:   None
*
* COMMENTS:  None
*
*/

```

```

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

```

```

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

```

```

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

```

```

/* FUNCTION: void Exitcmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId, int iSyncId)
*
* PURPOSE: This function removes a terminal id from use, the
allocated structure however remains
*
valid so the next request for a new client will not
require a new memory allocation.
*
* ARGUMENTS: int iFormId
unused
*
int iTermId id of calling browser, i.e. TERMID= from http
command line

```

```

*
int iSyncId sync id of calling browser
*
EXTENSION_CONTROL_BLOCK *pECB
structure pointer to passed in internet
*
service information.
* RETURNS: None
*
* COMMENTS: None
*
*/

```

```

void Exitcmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId, int
iSyncId)
{
    (*Term.Delete)(pECB, iTermId);
    WriteZString(pECB, MakeWelcomeForm() );
    UNUSEDPARAM(iFormId);
    UNUSEDPARAM(iSyncId);
    return;
}

```

```

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

```

```

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

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

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

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

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

    return;
}

/* FUNCTION: void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId,
int iTermId, int iSyncId)
*
* PURPOSE:   This function is the first command executed. It is
executed with the command
*           CMD=Begin?Server=xxx from the http command line.
*
* ARGUMENTS: int          iFormId
            unused
*           int
            iTermId      id of calling browser, i.e. TERMID= from http
command line
*           int
            iSyncId      sync id of calling browser
*           EXTENSION_CONTROL_BLOCK *pECB
            structure pointer to passed in internet
*
            service information.

```

```

* RETURNS:      None
*
* COMMENTS:    SQL server must be specified, however the user and
password parameters are optional.
*           The complete command line is
CMD=Begin&Server=server&User=sa&Psw=&. The & are used
*           to separate parameters which is internet
browser standard.
*/

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

    pQueryString = pECB->lpszQueryString;

    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
*
* 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
*
* 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. 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 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, &lpszSize, 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.
*
* char *format
* printf style format string
*
* other arguments as required by printf style format
string.
*
* RETURNS: None
*
* COMMENTS: This function is mainly used for developmental support.
*/

static void h_printf(EXTENSION_CONTROL_BLOCK *pECB, char *format, ...)
{
// int lpszSize;
char szBuff[512];
char szTmp[512];

va_list marker;
va_start(marker, format);
vsprintf(szTmp, format, marker);
va_end(marker);

// lpszSize = wsprintf(szBuff, "<html>%s</html>", szTmp) + 1;
//
// (*pECB->WriteClient)(pECB->ConnID, szBuff, &lpszSize, 0);

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

WriteZString(pECB, szBuff);

```

```

return;
}

void LogTuxError( int TpErrno, char *ErrMsg )
{
    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, tpstrerror(TpErrno),
ErrMsg);
    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 corrsponding Item_Id."
    },
    {      ERR_NEWORDER_QTY_WITHOUT_ITEMID,      "New
Order Qty entered without a corrsponding 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\">");
        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">", iErrorType);
        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"%d\">", iError);
        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), "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\">");
        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">", iErrorType);
        wsprintf(szForm+strlen(szForm), "<INPUT
TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"%d\">", iError);
        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), "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, iTotalConnections, iTickCount;

    EnterCriticalSection(&CriticalSection);

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

    if ( iTotalConnections >= iMaxConnections )
    {
        for(iCurrent = 1, i=1, iTickCount = 0x7FFFFFFF;
i<iMaxConnections; i++)
        {
            if ( iTickCount > Term.pClientData[i].iTickCount )
            {
                iTickCount =
Term.pClientData[i].iTickCount;
                iCurrent = i;
            }
        }
    }
}

```

```

else
{
    for(i=0; i<Term.iAvailable; i++)
    {
        if ( !Term.pClientData[i].inUse )
            break;
    }
    iCurrent = i;
}

if ( i == Term.iAvailable )
{
    Term.iNext = Term.iAvailable;
    if ( !(*Term.Allocate)() )
        goto TermAddErr1;
    for(i=Term.iNext; i<Term.iAvailable; i++)
        Term.pClientData[i].inUse = 0;
    iCurrent = Term.iNext;
}

Term.pClientData[iCurrent].inUse = 1;

if ( !GetKeyValue(pQueryString, "w_id", szTmp, sizeof(szTmp)) )
    goto TermAddErr1;

Term.pClientData[iCurrent].w_id = (short)atoi(szTmp);

if ( !GetKeyValue(pQueryString, "d_id", szTmp, sizeof(szTmp)) )
    goto TermAddErr1;

Term.pClientData[iCurrent].d_id = atoi(szTmp);

Term.pClientData[iCurrent].iTickCount = GetTickCount();
Term.pClientData[iCurrent].iSyncId = Term.iMasterSyncId++;

if ( Init(pECB, iCurrent, Term.pClientData[iCurrent].iSyncId,
szServer, szUser, szPassword, szDatabase) )
{
    (*Term.Delete)(pECB, iCurrent);
    goto TermAddErr1;
}

LeaveCriticalSection(&CriticalSection);
return iCurrent;

TermAddErr1:
LeaveCriticalSection(&CriticalSection);
return -1; //terminal unsuccessfully added
}

/* FUNCTION: void TermDelete(EXTENSION_CONTROL_BLOCK *pECB, int id)
*

```

```

* PURPOSE: This function makes a terminal entry in the Term array
available for reuse.
*
* ARGUMENTS: EXTENSION_CONTROL_BLOCK *pECB passed in structure
pointer from inetsrv.
*
*          int
*          id Terminal id of client exiting
*
* RETURNS: None
*
* COMMENTS: None
*
*/

static void TermDelete(EXTENSION_CONTROL_BLOCK *pECB, int id)
{
    if ( id >= 0 && id < Term.iAvailable )
    {
        Close(pECB, id, -1);
        Term.pClientData[id].inUse = 0;
    }

#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 the
 *            HTML formatted page being constructed.
 *
 * ARGUMENTS: char          *szDest Destination buffer where formatted
string is to be placed
             char          *szPic picture string which
describes how character value is to be
             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
        sprintf(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\">");

    sprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">", NEW_ORDER_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>
New
Order<BR>");

if ( bInput )
{
    sprintf(szForm+strlen(szForm), "Warehouse: %4.4d
District: <INPUT NAME=\"DID*\" SIZE=1>
Term.pClientData[iTermId].NewOrderData.w_id);
    strcat(szForm, "Customer: <INPUT NAME=\"CID*\" SIZE=4>
Name:
Credit:
%Disc:<BR>"
"Order Number:
D_tax:<BR><BR>"
" Supp_W Item_Id Item Name
Qty Stock B/G Price Amount<BR>"
" <INPUT NAME=\"SP00*\"
SIZE=4> <INPUT NAME=\"IID00*\" SIZE=6>
NAME=\"Qty00*\" SIZE=1><BR>"
" <INPUT NAME=\"SP01*\"
SIZE=4> <INPUT NAME=\"IID01*\" SIZE=6>
NAME=\"Qty01*\" SIZE=1><BR>"
" <INPUT NAME=\"SP02*\"
SIZE=4> <INPUT NAME=\"IID02*\" SIZE=6>
NAME=\"Qty02*\" SIZE=1><BR>"

```

```

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:

Total:<BR><HR>
" <INPUT TYPE=\"submit\"

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

NAME=\"CMD\" VALUE=\"Menu\">
" </FORM>
" </HTML> " );

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

```

```

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), "%Disc: %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);
        }
        else

```

```

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

        strcat(szForm, "</PRE><HR><BR>"
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 *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\">");
    wsprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">", PAYMENT_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>
Payment<BR>");

if ( bInput )
    strcat(szForm, "Date:<BR><BR>");
else
{
    wsprintf(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);
}

```

```

        wsprintf(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>"
                "
                "
                "
                "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 <= l )
                    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
*
* browser sync 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
                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\">");
        }
    }

    sprintf(szForm+strlen(szForm), "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">", DELIVERY_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>
Delivery<BR>");

    sprintf(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
        {
            sprintf(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
        {
            sprintf(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 *)tmalloc("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 = 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
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 *)tpalloc("CARRAY", NULL,
sizeof(DELIVERY_DATA))) == NULL)
    {
        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;
    }

    *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.
*
* 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 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, TPSIGRSTR))
== -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_tpinits=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(TLSIsTpInitedKey))
    {
        if (!once)
        {
            InitializeCriticalSection(&TpCriticalSection);
            once=1;
        }

        if ( dLog )
        {
            FILE *fp;

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

        while ( retry < TP_MAX_RETRIES )
        {
            EnterCriticalSection(&TpCriticalSection);
            if(tpinf == NULL)
            {
                if ((tpinf = ( TPINIT *)tpalloc("TPINIT",
NULL, sizeof(TPINIT))) == NULL)
                {
                    LeaveCriticalSection(&TpCriticalSection);
                    TpRc = tperrno;
                    {
                        FILE *fp;

                        fp = fopen(szErrorLogPath,
"ab");
                        fprintf(fp, ">>>>
ThrTpInit:%d : tmalloc of tpinf failed: %d : %s\r\n",
GetCurrentThreadId(),
TpRc, tpsterror(TpRc));
                        fclose(fp);
                    }
                }
            }
        }
    }
}

```

```

    }
    retry++;
    continue;
}

tpinf->flags|=TPMULTICONTEXTS;
}

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

// Do the TPINIT

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 tpinf if it failed the first time

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

        sprintf(ebuf, ">>>> ThrTpInit %d : Cannot
tpinf 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(TLSIsTpInitKey,&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

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

#ifndef DBINT
    typedef long DBINT;
#endif

#define DEFCLPACKSIZE      4096
#define DEADLOCKWAIT      10

// String length constants
#define SERVER_NAME_LEN    20
#define DATABASE_NAME_LEN 20
#define USER_NAME_LEN     20
#define PASSWORD_LEN     20
#define TABLE_NAME_LEN  20
#define I_DATA_LEN        50
#define I_NAME_LEN        24
#define BRAND_LEN         1
#define LAST_NAME_LEN     16
#define W_NAME_LEN        10
#define ADDRESS_LEN       20
#define STATE_LEN         2

```

```

#define ZIP_LEN 9
#define S_DIST_LEN 24
#define S_DATA_LEN 50
#define D_NAME_LEN 10
#define FIRST_NAME_LEN 16
#define MIDDLE_NAME_LEN 2
#define PHONE_LEN 16
#define DATETIME_LEN 30
#define CREDIT_LEN 2
#define C_DATA_LEN 250
#define H_DATA_LEN 24
#define DIST_INFO_LEN 24
#define MAX_OL_NEW_ORDER_ITEMS 15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN 25
#define OL_DIST_INFO_LEN 24

// transaction structures

typedef struct
{
    short ol_supply_w_id;
    long ol_i_id;
    char ol_i_name[I_NAME_LEN+1];
    short ol_quantity;
    char
ol_brand_generic[BRAND_LEN+1];
    double ol_i_price;
    double ol_amount;
    short ol_stock;
    short num_warehouses;
} OL_NEW_ORDER_DATA;

typedef struct
{
    short w_id;
    short d_id;
    long c_id;
    short o_ol_cnt;
    char c_last[LAST_NAME_LEN+1];
    char c_credit[CREDIT_LEN+1];
    double c_discount;
    double w_tax;
    double d_tax;
    long o_id;
    short o_commit_flag;
#ifdef USE_ODBC
    TIMESTAMP_STRUCT o_entry_d;
#else
    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=19100MB, FILEGROWTH=0),
    (name="MSSQL70_cs2",filename="I:",size=19100MB, FILEGROWTH=0),
    (name="MSSQL70_cs3",filename="J:",size=19100MB, FILEGROWTH=0),
    (name="MSSQL70_cs4",filename="K:",size=19100MB, FILEGROWTH=0),
    (name="MSSQL70_cs5",filename="M:",size=19100MB, FILEGROWTH=0),
    (name="MSSQL70_cs6",filename="N:",size=19100MB, FILEGROWTH=0)
to filegroup MSSQL70_cs_fg

alter database tpcc add file
    (name="MSSQL70_misc1",filename="O:",size=9200MB, FILEGROWTH=0),
    (name="MSSQL70_misc2",filename="P:",size=9200MB, FILEGROWTH=0),
    (name="MSSQL70_misc3",filename="Q:",size=9200MB, FILEGROWTH=0),
    (name="MSSQL70_misc4",filename="R:",size=9200MB, FILEGROWTH=0),
    (name="MSSQL70_misc5",filename="S:",size=9200MB, FILEGROWTH=0),
    (name="MSSQL70_misc6",filename="T:",size=9200MB, 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_c1' )
           drop index customer.customer_c1

create unique clustered index customer_c1 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_c1' )
           drop index district.district_c1

create unique clustered index district_c1 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

        @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,

        @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

```

```

        @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:      ORDDSTAT.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.00
--           Copyright Microsoft, 1996
-- Purpose:   Creates order status transaction stored procedure

use tpcc
go

if exists ( select name from sysobjects where name = "tpcc_orderstatus" )
    drop procedure tpcc_orderstatus
go

create proc tpcc_orderstatus @w_id          smallint,
                             @d_id          tinyint,
                             @c_id          int,
                             @c_last       char(16) = ""
as

declare @c_balance      numeric(12,2),
        @c_first       char(16),
        @c_middle      char(2),
        @o_id          int,
        @o_entry_d     datetime,
        @o_carrier_id  smallint,
        @cnt           smallint

begin tran o

    if (@c_id = 0)
        begin

-- get customer id and info using last name

            select @cnt = (count(*)+1)/2
            from customer (repeatableread)
            where c_last = @c_last and

```

```

        c_w_id = @w_id and
        c_d_id = @d_id

    set rowcount @cnt

    select @c_id = c_id,
           @c_balance = c_balance,
           @c_first = c_first,
           @c_last = c_last,
           @c_middle = c_middle
    from customer (repeatableread)
    where c_last = @c_last and
           c_w_id = @w_id and
           c_d_id = @d_id
    order by c_w_id, c_d_id, c_last, c_first

    set rowcount 0
    end

    else
        begin
-- get customer info if by id

        select @c_balance = c_balance,
               @c_first = c_first,
               @c_middle = c_middle,
               @c_last = c_last
        from customer (repeatableread)
        where c_id = @c_id and
               c_d_id = @d_id and
               c_w_id = @w_id

        select @cnt = @@rowcount

        end

-- if no such customer

        if (@cnt = 0)
            begin
                raiserror("Customer not found",18,1)
                goto custnotfound
            end

-- get order info

        select @o_id = o_id,
               @o_entry_d = o_entry_d,
               @o_carrier_id = o_carrier_id
        from orders (serializable)
        where o_c_id = @c_id and
               o_d_id = @d_id and

```

```

        o_w_id = @w_id
    order by o_id asc

-- select order lines for the current order

        select ol_supply_w_id,
               ol_i_id,
               ol_quantity,
               ol_amount,
               ol_delivery_d
        from order_line (repeatableread)
        where ol_o_id = @o_id and
               ol_d_id = @d_id and
               ol_w_id = @w_id

custnotfound:

commit tran o

-- return data to client

select @c_id,
       @c_last,
       @c_first,
       @c_middle,
       @o_entry_d,
       @o_carrier_id,
       @c_balance,
       @o_id

go

-- 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,

```

```

-- 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

```

```

@c_last char(16) =
""
as
declare @w_street_1 char(20),
        @w_street_2 char(20),
        @w_city char(20),
        @w_state char(2),
        @w_zip char(9),
        @w_name char(10),
        @d_street_1 char(20),
        @d_street_2 char(20),
        @d_city char(20),
        @d_state char(2),
        @d_zip char(9),
        @d_name char(10),
        @c_first char(16),
        @c_middle char(2),
        @c_street_1 char(20),
        @c_street_2 char(20),
        @c_city char(20),
        @c_state char(2),
        @c_zip char(9),
        @c_phone char(16),
        @c_since datetime,
        @c_credit char(2),
        @c_credit_lim numeric(12,2),
        @c_balance numeric(12,2),
        @c_discount numeric(4,4),
        @data char(500),
        @c_data char(500),
        @datetime datetime,
        @w_ytd numeric(12,2),
        @d_ytd numeric(12,2),
        @cnt smallint,
        @val smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local smallint,
        @c_id_local int

select @screen_data = ""

begin tran p

-- get payment date

select @datetime = getdate()

if (@c_id = 0)
begin

```

```

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.**

```

Key Name:          HARDWARE
Class Name:        <NO CLASS>
Last Write Time:   12/17/98 - 8:12 AM

Key Name:          HARDWARE\DESCRIPTION
Class Name:        <NO CLASS>
Last Write Time:   12/17/98 - 8:12 AM

Key Name:          HARDWARE\DESCRIPTION\System
Class Name:        System
Last Write Time:   12/17/98 - 8:12 AM
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 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:             11/05/98

Value 4
Name:             SystemBiosVersion
Type:             REG_MULTI_SZ
Data:             PhoenixBIOS 4.0 Release 6.1.3.5
                  PhoenixBIOS 4.0 Release 6.1.3.5
                  PhoenixBIOS 4.0 Release 6.1.3.5

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

                  ease 6.1.3.5
                  PhoenixBIOS 4.0 Release 6.1.3.5

```

Key Name: HARDWARE\DESCRIPTION\System\CentralProcessor  
Class Name: Processor  
Last Write Time: 12/17/98 - 8:12 AM

Key Name: HARDWARE\DESCRIPTION\System\CentralProcessor\0  
Class Name: Processor  
Last Write Time: 12/17/98 - 8:13 AM

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 3

Value 4  
Name: Update Signature  
Type: REG\_BINARY  
Data:  
00000000 00 00 00 00 04 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: 0x1c2

Key Name: HARDWARE\DESCRIPTION\System\CentralProcessor\1  
Class Name: Processor  
Last Write Time: 12/17/98 - 8:13 AM

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 3

Value 4
  Name:      Update Signature
  Type:      REG_BINARY
  Data:      00000000 00 00 00 00 04 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:      0x1c2

Key Name:      HARDWARE\DESCRIPTION\System\CentralProcessor\2
Class Name:    Processor
Last Write Time: 12/17/98 - 8:13 AM
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 5 Stepping 3

Value 4
  Name:      Update Signature
  Type:      REG_BINARY
  Data:      00000000 00 00 00 00 04 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:      0x1c2

Key Name:      HARDWARE\DESCRIPTION\System\CentralProcessor\3
Class Name:    Processor
Last Write Time: 12/17/98 - 8:13 AM
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 5 Stepping 3

Value 4
Name:          Update Signature
Type:          REG_BINARY
Data:          00000000 00 00 00 00 04 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:          0x1c2

Key Name:      HARDWARE\DESCRIPTION\System\FloatingPointProcessor
Class Name:    Processor
Last Write Time: 12/17/98 - 8:12 AM

Key Name:      HARDWARE\DESCRIPTION\System\FloatingPointProcessor\0
Class Name:    Processor
Last Write Time: 12/17/98 - 8:12 AM
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 3

Key Name:      HARDWARE\DESCRIPTION\System\FloatingPointProcessor\1
Class Name:    Processor
Last Write Time: 12/17/98 - 8:12 AM
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 3

Key Name:      HARDWARE\DESCRIPTION\System\FloatingPointProcessor\2
Class Name:    Processor
Last Write Time: 12/17/98 - 8:12 AM
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 5 Stepping 3

Key Name:      HARDWARE\DESCRIPTION\System\FloatingPointProcessor\3
Class Name:    Processor
Last Write Time: 12/17/98 - 8:12 AM
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 5 Stepping 3

Key Name:      HARDWARE\DESCRIPTION\System\MultifunctionAdapter
Class Name:    Adapter
Last Write Time: 12/17/98 - 8:12 AM

Key Name:      HARDWARE\DESCRIPTION\System\MultifunctionAdapter\0
Class Name:    Adapter
Last Write Time: 12/17/98 - 8:12 AM
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: 12/17/98 - 8:12 AM
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: 12/17/98 - 8:12 AM
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

Value 2
Name: Identifier
Type: REG_SZ
Data: PCI

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\11
Class Name: Adapter
Last Write Time: 12/17/98 - 8:12 AM
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: 11
Version: 0
Revision: 0

Value 2
Name: Identifier
Type: REG_SZ
Data: PCI

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\12
Class Name: Adapter
Last Write Time: 12/17/98 - 8:12 AM
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 - 4c 00 04 00 00 4f a7 00
$PnP.!..L...O..
00000010 f0 6d a7 00 00 0f 00 00 - 00 00 00 40 00 00 04 00
.m.....@....
00000020 00 32 00 00 41 d0 0c 02 - 08 80 00 03 00 47 01 80
.2..A.....G..
00000030 00 80 00 01 01 86 09 00 - 00 00 00 e0 ff 00 00 20
.....
00000040 00 86 09 00 00 00 00 e0 - fe 00 00 10 00 79 00 79
.....Y.Y
00000050 00 79 00 32 00 01 41 d0 - 0c 02 06 01 00 03 00 47
.y.2..A.....G
00000060 01 d0 04 d0 04 01 02 47 - 01 00 10 00 10 01 40 47
.....G.....@G
00000070 01 40 10 40 10 01 10 47 - 01 60 03 60 03 01 03 79
.@...G.'...'y
00000080 00 79 00 79 00 2a 00 02 - 41 d0 0c 02 08 80 00 03
.y.y.*..A.....
00000090 00 86 09 00 00 00 00 c1 - fe 00 00 0f 00 86 09 00
.....
000000a0 00 00 00 f0 fe 00 00 10 - 00 79 00 79 00 79 00 36
.....Y.Y.Y.6
000000b0 00 03 41 d0 0c 01 05 00 - 00 03 00 86 09 00 01 00
..A.....
000000c0 00 00 00 00 0a 00 86 - 09 00 60 00 00 0e 00 00
.....'.....
000000d0 00 02 00 86 09 00 00 00 - 00 10 00 00 00 e0 e0 79
.....Y
000000e0 00 79 00 2d 00 04 - 41 d0 02 00 08 01 01 03 .y.y.-
..A.....
000000f0 00 47 01 00 00 00 00 01 - 10 47 01 81 00 81 00 01
.G.....G.....
00000100 0f 47 01 c0 00 c0 00 01 - 20 2a 10 01 79 00 79 00 .G.....
*..Y.Y.
00000110 79 00 25 00 05 41 d0 00 - 00 08 00 01 03 00 47 01
y.%..A.....G.
00000120 20 00 20 00 01 02 47 01 - a0 00 a0 00 01 02 22 04 .
...G.....".
00000130 00 79 00 79 00 79 00 1d - 00 06 41 d0 01 00 08 02
.y.y.y....A.....
00000140 01 03 00 47 01 40 00 40 - 00 01 04 22 01 00 79 00
...G.@...".y.
00000150 79 00 79 00 1d 00 07 41 - d0 0b 00 08 03 01 03 00
y.y....A.....
00000160 47 01 70 00 70 00 01 02 - 22 00 01 79 00 79 00 79
G.p.p...".y.y.y
00000170 00 25 00 08 41 d0 03 03 - 09 00 00 03 00 47 01 60
.%.A.....G.'

```

```

00000180 00 60 00 01 01 47 01 64 - 00 64 00 01 01 22 02 00
.\...G.d.d..."...
00000190 79 00 79 00 79 00 1d 00 - 09 41 d0 0c 04 0b 80 00
Y.Y.Y...A.....
000001a0 03 00 47 01 f0 00 f0 00 - 01 10 22 00 20 79 00 79
..G.....". Y.Y
000001b0 00 79 00 1a 00 0a 41 d0 - 08 00 04 01 00 03 00 47
.Y...A.....G
000001c0 01 61 00 61 00 01 01 79 - 00 79 00 79 00 1a 00 0b
.a.a...Y.Y.Y...
000001d0 41 d0 0a 03 06 04 00 03 - 00 47 01 f8 0c f8 0c 01
A.....G.....
000001e0 08 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 - e8 0c 00 00 18 00 00 79 .....
.....Y
00000200 00 79 00 79 00 18 00 0f - 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 6e 00 11
..".Y."..Y.Y.n..
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 30
.G.w.w..."@.*..0
00000270 47 01 00 01 f8 0f 08 08 - 47 01 00 00 00 01 00
G.....G.....
00000280 22 ff 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: 12/17/98 - 8:12 AM

```

```

Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 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: 12/17/98 - 8:12 AM

```

```

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\DiskController\0
Class Name: Controller
Last Write Time: 12/17/98 - 8:12 AM

```

```

Value 0
Name: Component Information
Type: REG_BINARY
Data: 00000000 64 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: 12/17/98 - 8:12 AM

Key Name:  
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\DiskController\0\DiskPeripheral\0  
Class Name: Peripheral  
Last Write Time: 12/17/98 - 8:12 AM  
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: 23c10933-13a4ac1c-A

Key Name:  
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\DiskController\0\FloppyDiskPeripheral

Class Name: Peripheral  
Last Write Time: 12/17/98 - 8:12 AM

Key Name:  
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\DiskController\0\FloppyDiskPeripheral\0  
Class Name: Peripheral  
Last Write Time: 12/17/98 - 8:12 AM  
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\13\KeyboardController  
Class Name: Controller  
Last Write Time: 12/17/98 - 8:12 AM

Key Name:  
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\KeyboardController\0  
Class Name: Controller  
Last Write Time: 12/17/98 - 8:12 AM  
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: 12/17/98 - 8:12 AM

Key Name:  
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\KeyboardController\0\KeyboardPeripheral\0  
Class Name: Peripheral  
Last Write Time: 12/17/98 - 8:12 AM

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: 12/17/98 - 8:12 AM

Key Name:  
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\PointerController\0  
Class Name: Controller  
Last Write Time: 12/17/98 - 8:12 AM

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\13\PointerController\0\PointerPeripheral  
Class Name: Peripheral  
Last Write Time: 12/17/98 - 8:12 AM

Key Name:  
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\PointerController\0\PointerPeripheral\0  
Class Name: Peripheral  
Last Write Time: 12/17/98 - 8:12 AM

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\13\SerialController  
Class Name: Controller  
Last Write Time: 12/17/98 - 8:12 AM

Key Name:  
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\13\SerialController\0  
Class Name: Controller  
Last Write Time: 12/17/98 - 8:12 AM  
Value 0  
Name: Component Information  
Type: REG\_BINARY  
Data:

00000000 78 00 00 00 00 00 00 00 - 00 00 00 00 ff ff ff ff  
x.....

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: 0x000003e8  
Length: 0x7  
Type: Port

Partial Descriptor 1  
Resource: Interrupt  
Disposition: Undetermined  
Vector: 4  
Level: 4  
Affinity: 0xffffffff  
Type: Latched

Partial Descriptor 2  
Resource: Device Specific  
Disposition: Undetermined  
Reserved1: 0x00000000  
Reserved2: 0x00000000  
Data:

00000000 00 00 00 00 00 20 1c 00 - ..... ..

Value 2  
Name: Identifier  
Type: REG\_SZ  
Data: COM1

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\2  
Class Name: Adapter  
Last Write Time: 12/17/98 - 8:12 AM

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: 12/17/98 - 8:12 AM
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: 12/17/98 - 8:12 AM
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:    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: 12/17/98 - 8:12 AM
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: 12/17/98 - 8:12 AM
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: 12/17/98 - 8:12 AM
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: 12/17/98 - 8:12 AM
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: 12/17/98 - 8:12 AM
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: 12/17/98 - 8:12 AM

Key Name:    HARDWARE\DEVICEMAP\KeyboardClass
Class Name:  <NO CLASS>
Last Write Time: 12/17/98 - 8:13 AM
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: 12/17/98 - 8:13 AM
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: 12/17/98 - 8:13 AM  
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: 12/17/98 - 8:13 AM  
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: 12/17/98 - 8:12 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:12 AM  
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: 12/17/98 - 8:12 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Initiator Id 7  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:12 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 5  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:12 AM

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: 12/17/98 - 8:12 AM  
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: 12/17/98 - 8:12 AM  
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: 12/17/98 - 8:12 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 1\Scsi Bus  
 0\Initiator Id 7  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:12 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 1\Scsi Bus 0\Target  
 Id 0  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:12 AM

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: 12/17/98 - 8:12 AM

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: 12/17/98 - 8:12 AM

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: 12/17/98 - 8:12 AM

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  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

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: 0xe4410000

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 0  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus  
 0\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 1  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus  
 1\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 2  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus  
 2\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 3  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus  
 3\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 3\Target  
Id 0  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

Value 0  
Name: Identifier  
Type: REG\_SZ  
Data: MYLEX DAC1164P 0506

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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus  
4\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 2\Scsi Bus 4\Target  
Id 6  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM  
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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus  
0\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus 1  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus  
1\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus 2  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus  
2\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus 3  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus  
3\Initiator Id 254

Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus 3\Target  
 Id 0  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

Value 0  
 Name: Identifier  
 Type: REG\_SZ  
 Data: MYLEX DAC1164P 0506

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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus  
 4\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 3\Scsi Bus 4\Target  
 Id 6  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus  
 0\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 1  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus  
 1\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 2  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus  
 2\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 3  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM



Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 3\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 3\Target Id 0  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

Value 0  
 Name: Identifier  
 Type: REG\_SZ  
 Data: MYLEX DAC1164P 0506

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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 4\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 4\Scsi Bus 4\Target Id 6  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 0\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 1  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 1\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 2  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 2\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 3  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 3\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 3\Target Id 0  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM  
Value 0  
Name: Identifier  
Type: REG\_SZ  
Data: MYLEX DAC1164P 0506

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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 4\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 5\Scsi Bus 4\Target Id 6  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM  
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: 12/17/98 - 8:13 AM  
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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 0\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 1  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 1\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 2  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 2\Initiator Id 254

Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 3  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 3\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 3\Target Id 0  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM  
 Value 0  
 Name: Identifier  
 Type: REG\_SZ  
 Data: MYLEX DAC1164P 0506

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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 4\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 6\Scsi Bus 4\Target Id 6  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM  
 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: 12/17/98 - 8:13 AM  
 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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 0\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 1  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 1\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 2  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 2\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 3  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 3\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 3\Target Id 0  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

Value 0  
Name: Identifier  
Type: REG\_SZ  
Data: MYLEX DAC1164P 0506

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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 4\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 7\Scsi Bus 4\Target Id 6  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 0\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 1  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 1\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 2  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 2\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 3  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 3\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 3\Target Id 0  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

Value 0  
Name: Identifier  
Type: REG\_SZ  
Data: MYLEX DAC1164P 0506

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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 4\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 8\Scsi Bus 4\Target Id 6  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 0\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 0\Target Id 8  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM  
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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus  
1\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 1\Target  
Id 8  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM  
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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus  
2\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 3  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus  
3\Initiator Id 254  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 3\Target  
Id 0  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM  
Value 0  
Name: Identifier  
Type: REG\_SZ  
Data: MYLEX DAC1164P 0506

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: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM  
Value 0  
Name: Identifier  
Type: REG\_SZ  
Data: MYLEX DAC1164P 0506

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: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 4\Initiator Id 254  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 9\Scsi Bus 4\Target Id 6  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

Value 0  
 Name: Identifier  
 Type: REG\_SZ  
 Data: MYLEX GAM DEVICE

Value 1  
 Name: Type  
 Type: REG\_SZ  
 Data: OtherPeripheral

Key Name: HARDWARE\DEVICEMAP\SERIALCOMM  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:14 AM

Value 0  
 Name: Serial0  
 Type: REG\_SZ  
 Data: COM1

Key Name: HARDWARE\DEVICEMAP\VIDEO  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:14 AM

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: 12/17/98 - 8:12 AM

Key Name:    HARDWARE\RESOURCEMAP\Hardware Abstraction Layer
Class Name:  <NO CLASS>
Last Write Time: 12/17/98 - 8:12 AM

Key Name:    HARDWARE\RESOURCEMAP\Hardware Abstraction Layer\MPS
1.4 - APIC platform
Class Name:  <NO CLASS>
Last Write Time: 12/17/98 - 8:12 AM
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
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

```

```

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:      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: 0xfec00000  
 Length: 0x400  
 Type: Read / Write

Key Name: HARDWARE\RESOURCEMAP\KeyboardPort/PointerPort  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\RESOURCEMAP\KeyboardPort/PointerPort\i8042prt  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM  
 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:
Type:
Data:

```

```

\Device\KeyboardPort0.Translated
REG_RESOURCE_LIST

Full Resource Descriptor 0
Interface Type: Isa
Bus Number:    0
Version:      0
Revision:     0
Partial Descriptor 0
Resource:      Interrupt
Disposition:  Device Exclusive
Vector:       129
Level:        7
Affinity:     0x0000000f
Type:         Latched

Partial Descriptor 1
Resource:      Interrupt
Disposition:  Device Exclusive
Vector:       145
Level:        8
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\LOADED SERIAL DRIVER RESOURCES
Class Name:    <NO CLASS>
Last Write Time: 12/17/98 - 8:14 AM

```

```

Key Name:      HARDWARE\RESOURCEMAP\LOADED SERIAL DRIVER
RESOURCES\Serial
Class Name:    <NO CLASS>
Last Write Time: 12/17/98 - 8:14 AM

```

```

Value 0
Name:          \Device\Serial0.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:  Device Exclusive
Start:        0x000003e8
Length:       0x7
Type:         Port

Partial Descriptor 1
Resource:      Interrupt
Disposition:  Driver Exclusive
Vector:       4
Level:        4
Affinity:     0x00000000
Type:         Latched

```

```

Value 1
Name:      \Device\Serial0.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:  Device Exclusive
Start:       0x000003e8
Length:      0x7
Type:        Port

Partial Descriptor 1
Resource:     Interrupt
Disposition:  Driver Exclusive
Vector:      177
Level:       10
Affinity:    0x0000000f
Type:        Latched

```

```

Key Name:  HARDWARE\RESOURCEMAP\OtherDrivers
Class Name: <NO CLASS>
Last Write Time: 12/17/98 - 8:13 AM

```

```

Key Name:  HARDWARE\RESOURCEMAP\OtherDrivers\E100B
Class Name: <NO CLASS>
Last Write Time: 12/17/98 - 8:14 AM

```

```

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:      0x1c
Type:        Read / Write

Partial Descriptor 1
Resource:     Port
Disposition:  Device Exclusive

```

```

Start:      0x00003800
Length:     0x1c
Type:      Port

```

```

Partial Descriptor 2
Resource:   Interrupt
Disposition: Shared
Vector:    4
Level:     4
Affinity:  0x00000020
Type:      Level Sensitive

```

```

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:      0x1c
Type:        Read / Write

```

```

Partial Descriptor 1
Resource:     Port
Disposition:  Device Exclusive
Start:       0x00003800
Length:      0x1c
Type:        Port

```

```

Partial Descriptor 2
Resource:   Interrupt
Disposition: Shared
Vector:    179
Level:     10
Affinity:  0x0000000f
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:         0xe5300000
  Length:        0x1c
  Type:          Read / Write

Partial Descriptor 1
  Resource:      Port
  Disposition:   Device Exclusive
  Start:         0x00002020
  Length:        0x1c
  Type:          Port

Partial Descriptor 2
  Resource:      Interrupt
  Disposition:   Shared
  Vector:        48
  Level:         48
  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:         0xe5300000
    Length:        0x1c
    Type:          Read / Write

  Partial Descriptor 1
    Resource:      Port
    Disposition:   Device Exclusive
    Start:         0x00002020
    Length:        0x1c
    Type:          Port

  Partial Descriptor 2
    Resource:      Interrupt
    Disposition:   Shared

```

```

Vector:      163
Level:       9
Affinity:    0x0000000f
Type:        Level Sensitive

```

```

Key Name:      HARDWARE\RESOURCEMAP\OtherDrivers\Floppy
Class Name:    <NO CLASS>
Last Write Time: 12/17/98 - 8:13 AM
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:          161
Level:           9
Affinity:        0x0000000f
Type:            Latched
```

```
Key Name:      HARDWARE\RESOURCEMAP\ScsiAdapter
Class Name:    <NO CLASS>
Last Write Time: 12/17/98 - 8:12 AM
```

```
Key Name:      HARDWARE\RESOURCEMAP\ScsiAdapter\dac960nt
Class Name:    <NO CLASS>
Last Write Time: 12/17/98 - 8:13 AM
```

Value 0

```
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:           0xe4410000
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 1

```
Name:          \Device\ScsiPort2.Translated
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:          81
Level:           4
Affinity:        0x0000000f
Type:            Level Sensitive
```

```
Partial Descriptor 1
Resource:        Memory
Disposition:     Device Exclusive
Start:           0xe4410000
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 2
Name:          \Device\ScsiPort3.Raw
Type:          REG_RESOURCE_LIST
Data:

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 3
Name:          \Device\ScsiPort3.Translated
Type:          REG_RESOURCE_LIST
Data:

Full Resource Descriptor 0
Interface Type: PCI
Bus Number:    4
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:         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 4
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 5
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:          162
  Level:           9
  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 6
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 7

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: 146  
Level: 8  
Affinity: 0x0000000f  
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: Read / Write

Value 8

Name: \Device\ScsiPort6.Raw  
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: 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 9

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: 130  
Level: 7  
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 10
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 11
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:         114
Level:          6
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 12
Name:           \Device\ScsiPort8.Raw
Type:           REG_RESOURCE_LIST
Data:

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
  Length:        0x2000000
  Type:          Write Only

```

```

Value 13
Name:
Type:
Data:

```

```

\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:        98
  Level:         5
  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 14
Name:
Type:
Data:

```

```

\Device\ScsiPort9.Raw
REG_RESOURCE_LIST

Full Resource Descriptor 0
Interface Type:  PCI
Bus Number:     11
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:         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

Name: \Device\ScsiPort0.Raw  
Type: REG\_RESOURCE\_LIST  
Data:

Value 15

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: 82  
Level: 4  
Affinity: 0x0000000f  
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: Read / Write

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: 0xe4302000  
Length: 0x100  
Type: Read / Write

Partial Descriptor 3  
Resource: Memory  
Disposition: Device Exclusive  
Start: 0xe4300000  
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: 113  
Level: 6

Key Name: HARDWARE\RESOURCEMAP\ScsiAdapter\symc8XX  
Class Name: <NO CLASS>  
Last Write Time: 12/17/98 - 8:12 AM  
Value 0

```

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:       0xe4302000
Length:     0x100
Type:       Read / Write

Partial Descriptor 3
Resource:     Memory
Disposition:  Device Exclusive
Start:       0xe4300000
Length:     0x1000
Type:       Read / Write

```

```

Value 2
Name:
Type:
Data:

```

```

\Device\ScsiPort1.Raw
REG_RESOURCE_LIST

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:      0xe4302400
Length:    0x100
Type:     Read / Write

Partial Descriptor 3
Resource:  Memory
Disposition: Device Exclusive
Start:    0xe4301000
Length:  0x1000
Type:    Read / Write

```

```

Value 3
Name:
Type:
Data:

```

```

\Device\ScsiPort1.Translated
REG_RESOURCE_LIST

Full Resource Descriptor 0
Interface Type:  PCI
Bus Number:    1
Version:      0
Revision:     0
Partial Descriptor 0
Resource:     Interrupt
Disposition:  Shared
Vector:      97
Level:       5
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:       0xe4302400
Length:     0x100
Type:       Read / Write

Partial Descriptor 3
Resource:     Memory
Disposition:  Device Exclusive
Start:       0xe4301000
Length:     0x1000
Type:       Read / Write

```

Key Name: HARDWARE\RESOURCEMAP\System Resources  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:12 AM

Key Name: HARDWARE\RESOURCEMAP\System Resources\Physical Memory  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:12 AM

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: 0x9c000  
 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: 12/17/98 - 8:12 AM

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: 0x0009d000  
 Length: 0x3000  
 Type: Read / Write

Partial Descriptor 1  
 Resource: Memory  
 Disposition: Device Exclusive  
 Start: 0x000eb000  
 Length: 0x15000  
 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: 0x10000  
 Type: Read / Write

Partial Descriptor 4  
 Resource: Memory  
 Disposition: Device Exclusive  
 Start: 0xfec00000  
 Length: 0x1000  
 Type: Read / Write

Partial Descriptor 5  
 Resource: Memory  
 Disposition: Device Exclusive  
 Start: 0xffffeb000  
 Length: 0x15000  
 Type: Read / Write

Key Name: HARDWARE\RESOURCEMAP\VIDEO  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

Key Name: HARDWARE\RESOURCEMAP\VIDEO\cirrus  
 Class Name: <NO CLASS>  
 Last Write Time: 12/17/98 - 8:13 AM

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: 12/17/98 - 8:13 AM

Key Name:      HARDWARE\RESOURCEMAP\VIDEO\VgaStart
Class Name:    <NO CLASS>
Last Write Time: 12/17/98 - 8:13 AM

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: 12/15/97 - 2:00 PM

Value 0
Name:          DSQUERY
Type:          REG_SZ
Data:          DBMSSOCCN

Key Name:      SOFTWARE\Microsoft\MSSQLServer\Client\DB-Lib
Class Name:    <NO CLASS>

```

Last Write Time: 12/12/97 - 2:19 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: 12/15/98 - 2:07 PM

Value 0  
Name: .  
Type: REG\_SZ  
Data: 7.0

Value 1  
Name: SPACELAB  
Type: REG\_SZ  
Data: 7.0

Key Name: SOFTWARE\Microsoft\MSSQLServer\MSSQLServer  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:19 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: {9529E428-72F3-11D1-9814-0800060D8570}

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: 12/12/97 - 2:17 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 - 37 64 63 33 37 62 34 32  
 50da034b7dc37b42  
 00000020 65 32 62 39 64 36 31 62 - 38 31 31 38 65 63 35 64  
 e2b9d61b8118ec5d  
 00000030 36 37 36 34 35 33 65 61 - 39 34 65 33 64 35 61 31  
 676453ea94e3d5a1  
 00000040 31 63 36 63 34 64 63 39 - 63 37 63 38 30 37 31 30  
 1c6c4dc9c7c80710  
 00000050 38 36 66 61 62 65 39 31 - 39 65 34 34 35 62 36 65  
 86fabe919e445b6e  
 00000060 33 31 61 32 34 30 65 62 - 34 63 38 64 34 34 65 33  
 31a240eb4c8d44e3  
 00000070 31 37 32 34 39 35 36 38 - 66 65 36 38 66 37 65 31  
 17249568fe68f7e1  
 00000080 33 34 64 30 62 31 32 39 - 37 39 32 61 61 65 30 32  
 34d0b129792aae02  
 00000090 35 62 32 63 39 63 63 34 - 65 39 30 33 00  
 5b2c9cc4e903.

Value 1  
 Name: CurrentVersion  
 Type: REG\_SZ  
 Data: 7.00.611

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: 12/12/97 - 2:17 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: 12/12/97 - 2:17 PM

Value 0  
 Name: Security  
 Type: REG\_SZ  
 Data:

Key Name:  
 SOFTWARE\Microsoft\MSSQLServer\Providers  
 Class Name: <NO CLASS>  
 Last Write Time: 12/12/97 - 2:19 PM

Value 0  
 Name: AllowInProcess  
 Type: REG\_DWORD  
 Data: 0x1

Key Name:  
 SOFTWARE\Microsoft\MSSQLServer\Providers\ADSDSOObject  
 Class Name: <NO CLASS>  
 Last Write Time: 12/12/97 - 2:19 PM

Value 0  
 Name: AllowInProcess  
 Type: REG\_DWORD  
 Data: 0x1

Key Name:  
 SOFTWARE\Microsoft\MSSQLServer\Providers\DTSPackageDSO  
 Class Name: <NO CLASS>  
 Last Write Time: 12/12/97 - 2:19 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: 12/12/97 - 2:19 PM  
Value 0  
Name: AllowInProcess  
Type: REG\_DWORD  
Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers\MSDAORA  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:19 PM  
Value 0  
Name: AllowInProcess  
Type: REG\_DWORD  
Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers\MSDASQL  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:19 PM  
Value 0  
Name: AllowInProcess  
Type: REG\_DWORD  
Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers\MSIDXS  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:19 PM  
Value 0  
Name: AllowInProcess  
Type: REG\_DWORD  
Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers\MSQLImpProv  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:19 PM  
Value 0  
Name: AllowInProcess  
Type: REG\_DWORD  
Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers\MSSEARCHSQL  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:19 PM  
Value 0  
Name: AllowInProcess  
Type: REG\_DWORD  
Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Providers\SQLOLEDB  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:19 PM  
Value 0  
Name: AllowInProcess  
Type: REG\_DWORD  
Data: 0x1

Key Name: SOFTWARE\Microsoft\MSSQLServer\Replication  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:19 PM  
Value 0  
Name: DistributionDB  
Type: REG\_SZ  
Data:

Value 1  
Name: IsInstalled  
Type: REG\_DWORD  
Data: 0x1

Value 2  
Name: WorkingDirectory  
Type: REG\_SZ  
Data: D:\MSSQL7\REPLDATA

Key Name: SOFTWARE\Microsoft\MSSQLServer\Replication\ArticleResolver  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:17 PM  
Value 0  
Name: Microsoft SQLServer Stored Procedure Resolver  
Type: REG\_SZ  
Data: {6F31CE30-7BE4-11d1-9B0A-00C04FC2DEB3}

Key Name: SOFTWARE\Microsoft\MSSQLServer\Replication\DatabaseReconciler  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:17 PM

Key Name: SOFTWARE\Microsoft\MSSQLServer\Replication\DatabaseReconciler\7.0  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:17 PM

Key Name: SOFTWARE\Microsoft\MSSQLServer\Replication\DatabaseReconciler\7.0\MSSQLServer

Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:17 PM  
Value 0  
Name: <NO NAME>  
Type: REG\_SZ  
Data: {13D9E578-1CA8-11D0-A11B-00AA003E4672}

Key Name:  
SOFTWARE\Microsoft\MSSQLServer\Replication\MergeReplicationProvider  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:17 PM

Key Name:  
SOFTWARE\Microsoft\MSSQLServer\Replication\MergeReplicationProvider\7.0  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:17 PM

Key Name:  
SOFTWARE\Microsoft\MSSQLServer\Replication\MergeReplicationProvider\7.0\M  
sJet  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:17 PM  
Value 0  
Name: <NO NAME>  
Type: REG\_SZ  
Data: {f159cf30-0db4-11d1-b272-00aa00b8de95}

Key Name:  
SOFTWARE\Microsoft\MSSQLServer\Replication\MergeReplicationProvider\7.0\M  
SSQLServer  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:17 PM  
Value 0  
Name: <NO NAME>  
Type: REG\_SZ  
Data: {777D3984-11A3-11D0-A11A-00AA003E4672}

Key Name: SOFTWARE\Microsoft\MSSQLServer\Setup  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:17 PM  
Value 0  
Name: SourcePath  
Type: REG\_SZ  
Data: E:\sql70EE611

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: 12/12/97 - 2:19 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: 12/12/97 - 2:19 PM

Key Name: SOFTWARE\Microsoft\MSSQLServer\SQLEW\Replication  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:19 PM  
Value 0  
Name: PerfmonFile  
Type: REG\_SZ  
Data: D:\MSSQL7\BINN\REPLMON.PMC

Key Name: SOFTWARE\Microsoft\MSSQLServer\SQLEW\Wizards  
Class Name: <NO CLASS>  
Last Write Time: 12/12/97 - 2:19 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: 12/12/97 - 2:19 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: RestartsSQLServer  
   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: 12/12/97 - 2:19 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: 12/12/97 - 2:19 PM  
 Value 0

Name: {E07FDDA4-5A21-11d2-9DAD-00C04F79D434}  
 Type: REG\_SZ  
 Data:

Value 1  
 Name: {E07FDDA5-5A21-11d2-9DAD-00C04F79D434}  
 Type: REG\_SZ  
 Data:

Value 2  
 Name: {E07FDDA6-5A21-11d2-9DAD-00C04F79D434}  
 Type: REG\_SZ  
 Data:

Value 3  
 Name: {E07FDDA8-5A21-11d2-9DAD-00C04F79D434}  
 Type: REG\_SZ  
 Data:

Value 4  
 Name: {E07FDDA9-5A21-11d2-9DAD-00C04F79D434}  
 Type: REG\_SZ  
 Data:

Value 5  
 Name: {E07FDDAA-5A21-11d2-9DAD-00C04F79D434}  
 Type: REG\_SZ  
 Data:

Value 6  
 Name: {E07FDDAB-5A21-11d2-9DAD-00C04F79D434}  
 Type: REG\_SZ  
 Data:

Value 7  
 Name: {E07FDDAC-5A21-11d2-9DAD-00C04F79D434}  
 Type: REG\_SZ  
 Data:

Value 8  
 Name: {E07FDDAD-5A21-11d2-9DAD-00C04F79D434}  
 Type: REG\_SZ  
 Data:

Value 9  
 Name: {E07FDDAE-5A21-11d2-9DAD-00C04F79D434}  
 Type: REG\_SZ  
 Data:

Value 10  
 Name: {E07FDDAF-5A21-11d2-9DAD-00C04F79D434}  
 Type: REG\_SZ  
 Data:

Value 11  
 Name: {E07FDDBE-5A21-11d2-9DAD-00C04F79D434}  
 Type: REG\_SZ  
 Data:

Value 12  
 Name: {E07FDDBF-5A21-11d2-9DAD-00C04F79D434}  
 Type: REG\_SZ  
 Data:

Value 13  
 Name: {E07FDDC0-5A21-11d2-9DAD-00C04F79D434}  
 Type: REG\_SZ  
 Data:

Value 14  
 Name: {E07FDDC6-5A21-11d2-9DAD-00C04F79D434}  
 Type: REG\_SZ  
 Data:

Value 15  
 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: 12/18/98 - 12:33 PM

Key Name:    HARDWARE\DESCRIPTION
Class Name:  <NO CLASS>
Last Write Time: 12/18/98 - 12:33 PM

Key Name:    HARDWARE\DESCRIPTION\System
Class Name:  System
Last Write Time: 12/18/98 - 12:33 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 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: 12/18/98 - 12:33 PM

Key Name: HARDWARE\DESCRIPTION\System\CentralProcessor\0  
 Class Name: Processor  
 Last Write Time: 12/18/98 - 12:33 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: 12/18/98 - 12:33 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: 12/18/98 - 12:33 PM

Key Name: HARDWARE\DESCRIPTION\System\FloatingPointProcessor\0  
Class Name: Processor  
Last Write Time: 12/18/98 - 12:33 PM

Value 0  
Name: Component Information  
Type: REG\_BINARY  
Data:  
00000000 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: 12/18/98 - 12:33 PM  
Value 0  
Name: Component Information  
Type: REG\_BINARY  
Data:  
00000000 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: 12/18/98 - 12:33 PM

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\0  
Class Name: Adapter  
Last Write Time: 12/18/98 - 12:33 PM

Value 0  
Name: Component Information  
Type: REG\_BINARY  
Data:  
00000000 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: 12/18/98 - 12:33 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: 12/18/98 - 12:33 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: 12/18/98 - 12:33 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 10  
\.....  
000000b0 00 00 00 f0 1f 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.

```

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

```

```

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4
Class Name: Adapter
Last Write Time: 12/18/98 - 12:33 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: 12/18/98 - 12:33 PM

```

```

Key Name: HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\DiskController\0
Class Name: Controller
Last Write Time: 12/18/98 - 12:33 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\4\DiskController\0\DiskPeripheral
Class Name: Peripheral
Last Write Time: 12/18/98 - 12:33 PM

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\DiskController\0\DiskPeripheral\0
Class Name: Peripheral
Last Write Time: 12/18/98 - 12:33 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: 12/18/98 - 12:33 PM

Key Name:
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\DiskController\0\FloppyDiskPeripheral\0
Class Name: Peripheral
Last Write Time: 12/18/98 - 12:33 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  
.....%....1...0  
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: 12/18/98 - 12:33 PM

Key Name:  
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\KeyboardController\0  
Class Name: Controller  
Last Write Time: 12/18/98 - 12:33 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: 12/18/98 - 12:33 PM

Key Name:  
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\KeyboardController\0\KeyboardPeripheral\0  
Class Name: Peripheral  
Last Write Time: 12/18/98 - 12:33 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: 12/18/98 - 12:33 PM

Key Name:  
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\PointerController\0  
Class Name: Controller  
Last Write Time: 12/18/98 - 12:33 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: 12/18/98 - 12:33 PM

Key Name:  
HARDWARE\DESCRIPTION\System\MultifunctionAdapter\4\PointerController\0\PointerPeripheral\0  
Class Name: Peripheral  
Last Write Time: 12/18/98 - 12:33 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: 12/18/98 - 12:33 PM

Key Name: HARDWARE\DEVICEMAP\KeyboardClass  
Class Name: <NO CLASS>  
Last Write Time: 12/18/98 - 12:33 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: 12/18/98 - 12:33 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: 12/18/98 - 12:33 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: 12/18/98 - 12:33 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: 12/18/98 - 12:33 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0  
Class Name: <NO CLASS>  
Last Write Time: 12/18/98 - 3:40 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: 12/18/98 - 12:33 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Initiator Id 7  
Class Name: <NO CLASS>  
Last Write Time: 12/18/98 - 12:33 PM

Key Name: HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 0  
Class Name: <NO CLASS>  
Last Write Time: 12/18/98 - 12:33 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: 12/18/98 - 3:40 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: 12/18/98 - 12:33 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: 12/18/98 - 3:40 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: 12/18/98 - 12:33 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: 12/18/98 - 3:40 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: 12/18/98 - 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: 12/18/98 - 12:33 PM
Value 0
  Name:      PCI_0_c
  Type:      REG_SZ
  Data:      \Device\E100B1

Value 1
  Name:      PCI_2_0
  Type:      REG_SZ
  Data:      \Device\Video0

Value 2
  Name:      PCI_2_1
  Type:      REG_SZ
  Data:      \Device\ScsiPort0

Key Name:    HARDWARE\RESOURCEMAP
Class Name:  <NO CLASS>
Last Write Time: 12/18/98 - 12:33 PM

Key Name:    HARDWARE\RESOURCEMAP\Hardware Abstraction Layer
Class Name:  <NO CLASS>
Last Write Time: 12/18/98 - 12:33 PM

Key Name:    HARDWARE\RESOURCEMAP\Hardware Abstraction Layer\MPS
1.4 - APIC platform
Class Name:  <NO CLASS>
Last Write Time: 12/18/98 - 12:33 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
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

```

```

Value 1
Name:
Type:
Data:

```

```

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:         0fee00000
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: 12/18/98 - 12:33 PM

Key Name:      HARDWARE\RESOURCEMAP\KeyboardPort/PointerPort\i8042prt
Class Name:    <NO CLASS>
Last Write Time: 12/18/98 - 12:33 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: 12/18/98 - 12:33 PM

Key Name:      HARDWARE\RESOURCEMAP\OtherDrivers\E100B
Class Name:    <NO CLASS>
Last Write Time: 12/18/98 - 12:33 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:       0x1c
Type:         Read / Write

Partial Descriptor 1
Resource:      Port
Disposition:  Device Exclusive
Start:        0x0000e400
Length:       0x1c
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:      0x1c
Type:        Read / Write

Partial Descriptor 1
Resource:     Port
Disposition: Device Exclusive
Start:       0x0000e400
Length:      0x1c
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:      0x1c
Type:        Read / Write

Partial Descriptor 1
Resource:     Port
Disposition: Device Exclusive
Start:       0x0000e000
Length:      0x1c
Type:        Port

Partial Descriptor 2
Resource:     Interrupt
Disposition: Shared
Vector:      4
Level:       4
Affinity:    0x00000000
Type:        Level Sensitive

Value 3
Name:         \Device\E100B2.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:       0xfecfc000
Length:      0x1c
Type:        Read / Write

```

Partial Descriptor 1  
Resource: Port  
Disposition: Device Exclusive  
Start: 0x0000e000  
Length: 0x1c  
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: 0x1c  
Type: Read / Write

Partial Descriptor 1  
Resource: Port  
Disposition: Device Exclusive  
Start: 0x0000e800  
Length: 0x1c  
Type: Port

Partial Descriptor 2  
Resource: Interrupt  
Disposition: Shared  
Vector: 10  
Level: 10  
Affinity: 0x00000000  
Type: Level Sensitive

Value 5  
Name:  
Type:

\Device\E100B3.Translated  
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: 0xfecfe000  
Length: 0x1c  
Type: Read / Write

Partial Descriptor 1  
Resource: Port  
Disposition: Device Exclusive  
Start: 0x0000e800  
Length: 0x1c  
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: 0x1c  
Type: Read / Write

Partial Descriptor 1  
Resource: Port  
Disposition: Device Exclusive  
Start: 0x0000d400  
Length: 0x1c  
Type: Port

Partial Descriptor 2  
Resource: Interrupt  
Disposition: Shared  
Vector: 15  
Level: 15  
Affinity: 0x00000000  
Type: Level Sensitive

Value 7  
Name:  
Type:  
Data:

\Device\E100B4.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: 0xf7ffe000  
Length: 0x1c  
Type: Read / Write  
Partial Descriptor 1  
Resource: Port  
Disposition: Device Exclusive  
Start: 0x0000d400  
Length: 0x1c  
Type: Port  
Partial Descriptor 2  
Resource: Interrupt  
Disposition: Shared  
Vector: 162  
Level: 9  
Affinity: 0x00000003  
Type: Level Sensitive

Value 8  
Name:  
Type:  
Data:

\Device\E100B5.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: 0xf7ffd000  
Length: 0x1c  
Type: Read / Write

Partial Descriptor 1  
Resource: Port  
Disposition: Device Exclusive  
Start: 0x0000d000  
Length: 0x1c  
Type: Port

Partial Descriptor 2  
Resource: Interrupt  
Disposition: Shared  
Vector: 5  
Level: 5  
Affinity: 0x00000000  
Type: Level Sensitive

Value 9  
Name:  
Type:  
Data:

\Device\E100B5.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: 0xf7ffd000  
Length: 0x1c  
Type: Read / Write  
Partial Descriptor 1  
Resource: Port  
Disposition: Device Exclusive  
Start: 0x0000d000  
Length: 0x1c  
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: 0x1c
Type: Read / Write

Partial Descriptor 1
Resource: Port
Disposition: Device Exclusive
Start: 0x0000c800
Length: 0x1c
Type: Port

Partial Descriptor 2
Resource: Interrupt
Disposition: Shared
Vector: 14
Level: 14
Affinity: 0x00001000
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: 0x1c
Type: Read / Write

Partial Descriptor 1
```

```
Resource: Port
Disposition: Device Exclusive
Start: 0x0000c800
Length: 0x1c
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: 12/18/98 - 12:33 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:  
Type:  
Data:

.Translated  
REG\_RESOURCE\_LIST

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: 12/18/98 - 12:33 PM

Key Name: HARDWARE\RESOURCEMAP\ScsiAdapter\symc8XX  
Class Name: <NO CLASS>  
Last Write Time: 12/18/98 - 12:33 PM

Value 0  
Name:  
Type:  
Data:

\Device\ScsiPort0.Raw  
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: 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 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

Key Name: HARDWARE\RESOURCEMAP\System Resources  
 Class Name: <NO CLASS>  
 Last Write Time: 12/18/98 - 12:33 PM  
  
 Key Name: HARDWARE\RESOURCEMAP\System Resources\Physical Memory  
 Class Name: <NO CLASS>  
 Last Write Time: 12/18/98 - 12:33 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: 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: 0x1f000000  
 Type: Read / Write

Key Name: HARDWARE\RESOURCEMAP\System Resources\Reserved  
 Class Name: <NO CLASS>  
 Last Write Time: 12/18/98 - 12:33 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  
 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: 12/18/98 - 12:33 PM

Key Name: HARDWARE\RESOURCEMAP\VIDEO\cirrus  
 Class Name: <NO CLASS>  
 Last Write Time: 12/18/98 - 12:33 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: 12/18/98 - 12:33 PM

Key Name: HARDWARE\RESOURCEMAP\VIDEO\VgaStart  
 Class Name: <NO CLASS>  
 Last Write Time: 12/18/98 - 12:33 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:

Value 9  
Name: application/postscript,ai,,5  
Type: REG\_SZ  
Data:

Value 10  
Name: application/postscript,eps,,5  
Type: REG\_SZ  
Data:

Value 11  
Name: application/postscript,ps,,5  
Type: REG\_SZ  
Data:

Value 12  
Name: application/rtf,rtf,,5

Type: REG\_SZ  
Data:

Value 13  
Name: application/winhelp,hlp,,5  
Type: REG\_SZ  
Data:

Value 14  
Name: application/x-bcpio,bcpio,,5  
Type: REG\_SZ  
Data:

Value 15  
Name: application/x-cpio,cpio,,5  
Type: REG\_SZ  
Data:

Value 16  
Name: application/x-csh,csh,,5  
Type: REG\_SZ  
Data:

Value 17  
Name: application/x-director,dcr,,5  
Type: REG\_SZ  
Data:

Value 18  
Name: application/x-director,dir,,5  
Type: REG\_SZ  
Data:

Value 19  
Name: application/x-director,dxr,,5  
Type: REG\_SZ  
Data:

Value 20  
Name: application/x-dvi,dvi,,5  
Type: REG\_SZ  
Data:

Value 21  
Name: application/x-gtar,gtar,,9  
Type: REG\_SZ  
Data:

Value 22  
Name: application/x-hdf,hdf,,5  
Type: REG\_SZ  
Data:

Value 23  
Name: application/x-latex,latex,,5  
Type: REG\_SZ  
Data:

Value 24  
Name: application/x-msaccess,mdb,,5  
Type: REG\_SZ  
Data:

Value 25  
Name: application/x-mscardfile,crd,,5  
Type: REG\_SZ  
Data:

Value 26  
Name: application/x-msclip,clip,,5  
Type: REG\_SZ  
Data:

Value 27  
Name: application/x-msexcel,xla,,5  
Type: REG\_SZ  
Data:

Value 28  
Name: application/x-msexcel,xlc,,5  
Type: REG\_SZ  
Data:

Value 29  
Name: application/x-msexcel,xlm,,5  
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:	Type: REG_SZ	Data:
Value 56	Name: application/x-tex,tex,,5	Type: REG_SZ	Data:	Value 66	Name: application/x-wais-source,src,,7
Value 57	Name: application/x-texinfo,txi,,5	Type: REG_SZ	Data:	Value 67	Name: application/zip,zip,,9
Value 58	Name: application/x-texinfo,texinfo,,5	Type: REG_SZ	Data:	Value 68	Name: audio/basic,au,,<
Value 59	Name: application/x-troff,roff,,5	Type: REG_SZ	Data:	Value 69	Name: audio/basic,snd,,<
Value 60	Name: application/x-troff,t,,5	Type: REG_SZ	Data:	Value 70	Name: audio/x-aiff,aif,,<
Value 61	Name: application/x-troff,tr,,5	Type: REG_SZ	Data:	Value 71	Name: audio/x-aiff,aifc,,<
Value 62	Name: application/x-troff-man,man,,5	Type: REG_SZ	Data:	Value 72	Name: audio/x-aiff,aiff,,<
Value 63	Name: application/x-troff-me,me,,5	Type: REG_SZ	Data:	Value 73	Name: audio/x-pn-realaudio,ram,,<
Value 64	Name: application/x-troff-ms,ms,,5	Type: REG_SZ	Data:	Value 74	Name: audio/x-wav,wav,,<
Value 65	Name: application/x-ustar,ustar,,5	Type: REG_SZ	Data:	Value 75	Name: image/bmp,bmp,,:

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: 12/18/98 - 12:33 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: 12/8/97 - 12:53 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: 12/8/97 - 12:53 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: 12/14/97 - 10:41 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: 0x2bc

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: 0x2bc

Value 9  
 Name: TUXIPC\_SEM  
 Type: REG\_DWORD  
 Data: 0x1800

Value 10  
 Name: TUXIPC\_SEM\_IDS  
 Type: REG\_DWORD  
 Data: 0x1800

Value 11  
 Name: TUXIPC\_SEM\_UNDO  
 Type: REG\_DWORD  
 Data: 0x1800

Value 12  
 Name: TUXIPC\_SHM\_PROCS  
 Type: REG\_DWORD  
 Data: 0x2bc

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: 12/7/97 - 4:00 PM

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\_QUEUEUES  
 Type: REG\_DWORD  
 Data: 0x2bc

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: 0x2bc

Value 7

Name: TUXIPC\_SEM  
 Type: REG\_DWORD  
 Data: 0x1800

Value 8  
 Name: TUXIPC\_SEM\_IDS  
 Type: REG\_DWORD  
 Data: 0x1800

Value 9  
 Name: TUXIPC\_SEM\_UNDO  
 Type: REG\_DWORD  
 Data: 0x1800

Value 10  
 Name: TUXIPC\_SHM\_PROCS  
 Type: REG\_DWORD  
 Data: 0x2bc

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  
 NTLMSPP

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\inet\_srv\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: 12/18/98 - 12:33 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\inetsrv\sspicfilt.dll  
 Value 13  
   Name: GlobalExpire  
   Type: REG\_DWORD  
   Data: 0xffffffff  
 Value 14  
   Name: InstallPath  
   Type: REG\_SZ  
   Data: C:\WINNT\System32\inetsrv  
 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\inetsrv\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 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 05 - 12 00 00 00 25 02 00 00  
.....%...  
000000c0 01 01 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

**This section discloses the Transaction monitor tunable parameters parameters used on the Primergy 470 client system.**

```
*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=23 MAX=35
        CLOPT="-A -- -Sspacelab"
```

```
RQADDR=newq REPLYQ=Y

payment SRVGRP=GROUPPAY
SRVID=200
MIN=13 MAX=22
CLOPT="-A -- -Sspacelab"
RQADDR=payq REPLYQ=Y

orderstatus SRVGRP=GROUPPOS
SRVID=300
MIN=2 MAX=5
CLOPT="-A -- -Sspacelab"
RQADDR=ordq REPLYQ=Y

stocklevel SRVGRP=GROUPSL
SRVID=400
MIN=23 MAX=32
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 was started with the following command line options:

```
sqlservr -c -x -T3502
```

where:

- c starts 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

Microsoft SQL Server Stack Size:

The default stack size for Microsoft SQL Server 7.0 was changed using the EDITBIN utility:  
editbin /STACK:131072

The following Microsoft SQL Server configuration parameters were used:

name	minimum	maximum	config_value	run_value
affinity mask	0	2147483647		0
allow updates	0	1		1
cost threshold for parallelism	0	32767		5
cursor threshold	-1	2147483647		-1
default language	0	9999		0
default sortorder id	0	255		50
extended memory size (MB)	0	2147483647		0
fill factor (%)	0	100		0
index create memory (KB)	704	1600000		0
language in cache	0	100		3
language neutral full-text	0	1		0
lightweight pooling	0	1		1
locks	0	1		1
max async IO	5000	2147483647		8000
max degree of parallelism	1	255		255
max server memory (MB)	0	32		1
max text repl size (B)	4	2147483647	2147483647	2147483647
max worker threads	10	1024		65536
media retention	0	365		260
min memory per query (KB)	512	2147483647		512
min server memory (MB)	0	2147483647		0
nested triggers	0	1		1
network packet size (B)	512	65535		4096
open objects	0	2147483647		0
priority boost	0	1		0
query governor cost limit	0	2147483647		1
query wait (s)	-1	2147483647		0
recovery interval (min)	0	32767	32767	-1
remote access	0	1		1
remote login timeout (s)	0	2147483647		5
remote proc trans	0	1		0
remote query timeout (s)	0	2147483647		0
resource timeout (s)	5	2147483647		10
scan for startup procs	0	1		0
set working set size	0	1		0
show advanced options	0	1		1
spin counter	1	2147483647		10000
time slice (ms)	50	1000		50
two digit year cutoff	1753	9999		2049
Unicode comparison style	0	2147483647		0
Unicode locale id	0	2147483647		33280
user connections	0	32767		270
user options	0	4095		0

**This section discloses hardware information of the Primergy 870-40 server system.**

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: 11/05/98  
BIOS Version: PhoenixBIOS 4.0 Release 6.1.3.5

Processor list:

0: x86 Family 6 Model 5 Stepping 3 GenuineIntel ~450 Mhz  
1: x86 Family 6 Model 5 Stepping 3 GenuineIntel ~450 Mhz  
2: x86 Family 6 Model 5 Stepping 3 GenuineIntel ~450 Mhz  
3: x86 Family 6 Model 5 Stepping 3 GenuineIntel ~450 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: 2,048,255 KB, Free: 1,536,662 KB  
Serial Number: 8427 - 1D6F  
Bytes per cluster: 512  
Sectors per cluster: 1  
Filename length: 255  
D:\ (Local - NTFS) TPCC Total: 6,835,624 KB, Free: 3,478,584 KB  
Serial Number: B873 - EC63  
Bytes per cluster: 512  
Sectors per cluster: 8  
Filename length: 255  
X:\ (Local - NTFS) BACKUP1 Total: 106,695,660 KB, Free: 31,545,624 KB  
Serial Number: 10AC - 718B  
Bytes per cluster: 512  
Sectors per cluster: 8  
Filename length: 255  
Y:\ (Local - NTFS) BACKUP2 Total: 106,695,660 KB, Free: 31,983,988 KB  
Serial Number: 9CBA - 487A  
Bytes per cluster: 512  
Sectors per cluster: 8  
Filename length: 255

Memory Report

Handles: 4,190  
Threads: 100  
Processes: 15  
Physical Memory (K)  
Total: 3,734,952  
Available: 3,538,624  
File Cache: 12,676  
Kernel Memory (K)  
Total: 15,516  
Paged: 9,476  
Nonpaged: 6,040

Commit Charge (K)  
 Total: 86,088  
 Limit: 4,910,428  
 Peak: 3,010,896

Pagefile Space (K)  
 Total: 1,321,984  
 Total in use: 5,708  
 Peak: 8,000

C:\pagefile.sys  
 Total: 273,408  
 Total in use: 2,848  
 Peak: 4,036

D:\pagefile.sys  
 Total: 1,048,576  
 Total in use: 2,860  
 Peak: 3,964

Services Report

```
-----
----
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
Server                                Stopped
(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
MSDTC (MS Transactions)               Stopped   (Manual)
  C:\WINNTPCC\System32\msdtc.exe
  Service Account Name: LocalSystem
  Error Severity: Normal
  Service Flags: Own Process
```

Service Dependencies:			LanmanWorkstation		
RPCSS			LanmanServer		
NTLMSSP			Remote Procedure Call (RPC) Locator	Stopped	(Manual)
MSSQLServer	Stopped	(Manual)	C:\WINNTPCC\System32\LOCATOR.EXE		
D:\MSSQL7\bin\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	(Manual)	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		
NetDDEDSDM			Error Severity: Normal		
Network DDE DSDM	Stopped	(Manual)	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\bin\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					
Directory Replicator	Stopped	(Manual)	Drivers Report		
C:\WINNTPCC\System32\lmrepl.exe			-----		
Service Account Name: LocalSystem			----		
Error Severity: Normal			Abiosdsk (Primary disk)	Stopped	(Disabled)
Service Flags: Own Process			Error Severity: Ignore		
Service Dependencies:			Service Flags: Kernel Driver, Shared Process		



AFD Networking Support Environment (TDI) (Automatic)	Running		Cdrom (SCSI CDROM Class)	Running	(System)
C:\WINNTPCC\System32\drivers\afd.sys			Error Severity: Ignore		
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process		
Service Flags: Kernel Driver, Shared Process			Group Dependencies:		
Aha154x (SCSI miniport)	Stopped	(Disabled)	SCSI miniport		
Error Severity: Normal			Changer (Filter)	Stopped	(System)
Service Flags: Kernel Driver, Shared Process			Error Severity: Ignore		
Aha174x (SCSI miniport)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process		
Error Severity: Normal			cirrus (Video)	Running	(System)
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal		
aic78xx (SCSI miniport)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process		
Error Severity: Normal			Cpqarray (SCSI miniport)	Stopped	(Disabled)
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal		
Always (SCSI miniport)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process		
Error Severity: Normal			cpqfw2e (SCSI miniport)	Stopped	(Disabled)
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal		
ami0nt (SCSI miniport)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process		
Error Severity: Normal			dac960nt (SCSI miniport)	Running	(Boot)
Service Flags: Kernel Driver, Shared Process			C:\WINNTPCC\System32\drivers\dac960nt.sys		
amsint (SCSI miniport)	Stopped	(Disabled)	Error Severity: Normal		
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process		
Service Flags: Kernel Driver, Shared Process			dce376nt (SCSI miniport)	Stopped	(Disabled)
Arrow (SCSI miniport)	Stopped	(Disabled)	Error Severity: Normal		
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process		
Service Flags: Kernel Driver, Shared Process			Delldsa (SCSI miniport)	Stopped	(Disabled)
atapi (SCSI miniport)	Stopped	(Disabled)	Error Severity: Normal		
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process		
Service Flags: Kernel Driver, Shared Process			Dell_DGX (Video)	Stopped	(Disabled)
Atdisk (Primary disk)	Stopped	(Disabled)	Error Severity: Ignore		
Error Severity: Ignore			Service Flags: Kernel Driver, Shared Process		
Service Flags: Kernel Driver, Shared Process			Disk (SCSI Class)	Running	(Boot)
ati (Video)	Stopped	(Disabled)	Error Severity: Ignore		
Error Severity: Ignore			Service Flags: Kernel Driver, Shared Process		
Service Flags: Kernel Driver, Shared Process			Group Dependencies:		
Beep (Base)	Running	(System)	SCSI miniport		
Error Severity: Normal			Diskperf (Filter)	Stopped	(Disabled)
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal		
BusLogic (SCSI miniport)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process		
Error Severity: Normal			DptScsi (SCSI miniport)	Stopped	(Disabled)
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal		
Busmouse (Pointer Port)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process		
Error Severity: Ignore			dtc329x (SCSI miniport)	Stopped	(Disabled)
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal		
Cdaudio (Filter)	Stopped	(System)	Service Flags: Kernel Driver, Shared Process		
Error Severity: Ignore			Intel EtherExpress PRO Adapter (NDIS)	Running	
Service Flags: Kernel Driver, Shared Process			(Automatic)		
Cdfs (File system)	Running	(Disabled)	C:\WINNTPCC\System32\drivers\e100bnt.sys		
Error Severity: Normal			Error Severity: Normal		
Service Flags: File System Driver, Shared Process			Service Flags: Kernel Driver, Shared Process		
Group Dependencies:			et4000 (Video)	Stopped	(Disabled)
SCSI CDROM Class			Error Severity: Ignore		
			Service Flags: Kernel Driver, Shared Process		

Fastfat (Boot file system)	Running	(Disabled)	mga_mil (Video)	Stopped	(Disabled)
Error Severity: Normal			Error Severity: Ignore		
Service Flags: File System Driver, Shared Process			Service Flags: Kernel Driver, Shared Process		
Fd16_700 (SCSI miniport)	Stopped	(Disabled)	mitsumi (SCSI miniport)	Stopped	(Disabled)
Error Severity: Normal			Error Severity: Normal		
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process		
Fd7000ex (SCSI miniport)	Stopped	(Disabled)	mkecr5xx (SCSI miniport)	Stopped	(Disabled)
Error Severity: Normal			Error Severity: Normal		
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process		
Fd8xx (SCSI miniport)	Stopped	(Disabled)	Modem (Extended base)	Stopped	(Manual)
Error Severity: Normal			Error Severity: Ignore		
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process		
flashpnt (SCSI miniport)	Stopped	(Disabled)	Mouse Class Driver (Pointer Class)	Running	(System)
Error Severity: Normal			System32\DRIVERS\mouclass.sys		
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal		
Floppy (Primary disk)	Running	(System)	Service Flags: Kernel Driver, Shared Process		
Error Severity: Ignore			Msf (File system)	Running	(System)
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal		
Ftdisk (Filter)	Stopped	(Disabled)	Service Flags: File System Driver, Shared Process		
Error Severity: Ignore			Mup (Network)	Running	(Manual)
Service Flags: Kernel Driver, Shared Process			C:\WINNTPCC\System32\drivers\mup.sys		
i8042 Keyboard and PS/2 Mouse Port Driver (Keyboard Port)	Running	(System)	Error Severity: Normal		
System32\DRIVERS\i8042prt.sys			Service Flags: File System Driver, Shared Process		
Error Severity: Normal			Ncr53c9x (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			ncr77c22 (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			Ncrc700 (SCSI miniport)	Stopped	(Disabled)
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal		
Jazzg364 (Video)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process		
Error Severity: Ignore			Ncrc710 (SCSI miniport)	Stopped	(Disabled)
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal		
Jzvxl484 (Video)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process		
Error Severity: Ignore			Microsoft NDIS System Driver (NDIS)	Running	(System)
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal		
Keyboard Class Driver (Keyboard Class)	Running	(System)	Service Flags: Kernel Driver, Shared Process		
System32\DRIVERS\kbdclass.sys			NetBIOS Interface (NetBIOSGroup)	Stopped	(Manual)
Error Severity: Normal			C:\WINNTPCC\System32\drivers\netbios.sys		
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal		
KSecDD (Base)	Running	(System)	Service Flags: File System Driver, Shared Process		
Error Severity: Normal			Group Dependencies:		
Service Flags: Kernel Driver, Shared Process			TDI		
macdisk (Filter)	Running	(Boot)	WINS Client (TCP/IP) (PNP_TDI)	Running	
C:\WINNTPCC\System32\drivers\macdisk.sys			(Automatic)		
Error Severity: Normal			C:\WINNTPCC\System32\drivers\netbt.sys		
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal		
mga (Video)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process		
Error Severity: Ignore			Service Dependencies:		
Service Flags: Kernel Driver, Shared Process			Tcpip		
			NetDetect	Stopped	(Manual)

C:\WINNTPCC\system32\drivers\netdtecl.sys			Error Severity: Normal	
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process	
Service Flags: Kernel Driver, Shared Process			qv (Video)	Stopped (Disabled)
Npfs (File system)	Running	(System)	Error Severity: Ignore	
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process	
Service Flags: File System Driver, Shared Process			Rdr (Network)	Running (Manual)
Ntfs (File system)	Running	(Disabled)	C:\WINNTPCC\System32\drivers\rdr.sys	
Error Severity: Normal			Error Severity: Normal	
Service Flags: File System Driver, Shared Process			Service Flags: File System Driver, Shared Process	
Null (Base)	Running	(System)	s3 (Video)	Stopped (Disabled)
Error Severity: Normal			Error Severity: Ignore	
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process	
Ollscsi (SCSI miniport)	Stopped	(Disabled)	Scsiprnt (Extended base)	Stopped
Error Severity: Normal			(Automatic)	
Service Flags: Kernel Driver, Shared Process			Error Severity: Ignore	
Parallel (Extended base)	Stopped		Service Flags: Kernel Driver, Shared Process	
(Automatic)			Group Dependencies:	
Error Severity: Ignore			SCSI miniport	
Service Flags: Kernel Driver, Shared Process			Scsiscan (SCSI Class)	Running (System)
Service Dependencies:			Error Severity: Ignore	
Parport			Service Flags: Kernel Driver, Shared Process	
Group Dependencies:			Group Dependencies:	
Parallel arbitrator			SCSI miniport	
Parport (Parallel arbitrator)	Stopped		Serial (Extended base)	Running
(Automatic)			(Automatic)	
Error Severity: Ignore			Error Severity: Ignore	
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process	
ParVdm (Extended base)	Stopped		Sermouse (Pointer Port)	Stopped (Disabled)
(Automatic)			Error Severity: Ignore	
Error Severity: Ignore			Service Flags: Kernel Driver, Shared Process	
Service Flags: Kernel Driver, Shared Process			Sfloppy (Primary disk)	Stopped (System)
Service Dependencies:			Error Severity: Ignore	
Parport			Service Flags: Kernel Driver, Shared Process	
Group Dependencies:			Group Dependencies:	
Parallel arbitrator			SCSI miniport	
PCIDump (PCI Configuration)	Stopped	(System)	Simbad (Filter)	Stopped (Disabled)
Error Severity: Ignore			Error Severity: Normal	
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process	
Pcmcia (System Bus Extender)	Stopped	(Disabled)	slcd32 (SCSI miniport)	Stopped (Disabled)
Error Severity: Normal			Error Severity: Normal	
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process	
PnP ISA Enabler Driver (Base)	Stopped	(System)	Sparrow (SCSI miniport)	Stopped (Disabled)
Error Severity: Ignore			Error Severity: Normal	
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process	
PROKDD	Stopped	(Manual)	Spock (SCSI miniport)	Stopped (Disabled)
??\C:\WINNTPCC\SYSTEM32\Drivers\PROKDD.SYS			Error Severity: Normal	
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process	
Service Flags: Kernel Driver, Shared Process			Srv (Network)	Stopped (Manual)
psidisp (Video)	Stopped	(Disabled)	C:\WINNTPCC\System32\drivers\srv.sys	
Error Severity: Ignore			Error Severity: Normal	
Service Flags: Kernel Driver, Shared Process			Service Flags: File System Driver, Shared Process	
Q110wnt (SCSI miniport)	Stopped	(Disabled)	symc810 (SCSI miniport)	Stopped (Disabled)

```

Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
symc8XX (SCSI miniport) Running (Boot)
C:\WINNTPCC\system32\drivers\symc8XX.sys
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Tl28 (SCSI miniport) Stopped (Disabled)
Error Severity: Normal
Service Flags: Kernel Driver, Shared Process
Tl3B (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
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 0x0000000f
MPS 1.4 - APIC platform                0      0 0x0000000f
MPS 1.4 - APIC platform                1      1 0x0000000f
MPS 1.4 - APIC platform                2      2 0x0000000f
MPS 1.4 - APIC platform                3      3 0x0000000f
MPS 1.4 - APIC platform                4      4 0x0000000f
MPS 1.4 - APIC platform                5      5 0x0000000f
MPS 1.4 - APIC platform                6      6 0x0000000f
MPS 1.4 - APIC platform                7      7 0x0000000f
MPS 1.4 - APIC platform                8      8 0x0000000f
MPS 1.4 - APIC platform                9      9 0x0000000f
MPS 1.4 - APIC platform               10     10 0x0000000f
MPS 1.4 - APIC platform               11     11 0x0000000f
MPS 1.4 - APIC platform               12     12 0x0000000f
MPS 1.4 - APIC platform               13     13 0x0000000f
MPS 1.4 - APIC platform               14     14 0x0000000f
MPS 1.4 - APIC platform               15     15 0x0000000f
MPS 1.4 - APIC platform               16     16 0x0000000f
MPS 1.4 - APIC platform               17     17 0x0000000f
MPS 1.4 - APIC platform               18     18 0x0000000f
MPS 1.4 - APIC platform               19     19 0x0000000f
MPS 1.4 - APIC platform               20     20 0x0000000f
MPS 1.4 - APIC platform               21     21 0x0000000f
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               25     25 0x0000000f
MPS 1.4 - APIC platform               26     26 0x0000000f
MPS 1.4 - APIC platform               27     27 0x0000000f
MPS 1.4 - APIC platform               28     28 0x0000000f
MPS 1.4 - APIC platform               29     29 0x0000000f
MPS 1.4 - APIC platform               30     30 0x0000000f
MPS 1.4 - APIC platform               31     31 0x0000000f
MPS 1.4 - APIC platform               32     32 0x0000000f

```

MPS 1.4 - APIC platform	33	33	0x0000000f
MPS 1.4 - APIC platform	34	34	0x0000000f
MPS 1.4 - APIC platform	35	35	0x0000000f
MPS 1.4 - APIC platform	36	36	0x0000000f
MPS 1.4 - APIC platform	37	37	0x0000000f
MPS 1.4 - APIC platform	38	38	0x0000000f
MPS 1.4 - APIC platform	39	39	0x0000000f
MPS 1.4 - APIC platform	40	40	0x0000000f
MPS 1.4 - APIC platform	41	41	0x0000000f
MPS 1.4 - APIC platform	42	42	0x0000000f
MPS 1.4 - APIC platform	43	43	0x0000000f
MPS 1.4 - APIC platform	44	44	0x0000000f
MPS 1.4 - APIC platform	45	45	0x0000000f
MPS 1.4 - APIC platform	46	46	0x0000000f
MPS 1.4 - APIC platform	47	47	0x0000000f
MPS 1.4 - APIC platform	61	61	0x0000000f
MPS 1.4 - APIC platform	65	65	0x0000000f
MPS 1.4 - APIC platform	80	80	0x0000000f
MPS 1.4 - APIC platform	193	193	0x0000000f
MPS 1.4 - APIC platform	225	225	0x0000000f
MPS 1.4 - APIC platform	253	253	0x0000000f
MPS 1.4 - APIC platform	254	254	0x0000000f
MPS 1.4 - APIC platform	255	255	0x0000000f
i8042prt	1	1	0xffffffff
i8042prt	12	12	0xffffffff
Serial	4	4	0x00000000
E100B	4	4	0x00000020
E100B	48	48	0x00000000
Floppy	6	6	0x00000000
dac960nt	32	32	0x00000000
dac960nt	32	32	0x00000000
dac960nt	32	32	0x00000000
dac960nt	32	32	0x00000000
dac960nt	32	32	0x00000000
dac960nt	32	32	0x00000000
dac960nt	32	32	0x00000000
dac960nt	32	32	0x00000000
dac960nt	32	32	0x00000000
dac960nt	32	32	0x00000000
symc8XX	8	8	0x00000000
symc8XX	12	12	0x00000000

-----

Devices	Physical Address	Length
-----		
MPS 1.4 - APIC platform	0x00000000	0x0000000010
MPS 1.4 - APIC platform	0x00000020	0x0000000002
MPS 1.4 - APIC platform	0x00000040	0x0000000004
MPS 1.4 - APIC platform	0x00000048	0x0000000004
MPS 1.4 - APIC platform	0x00000061	0x0000000001
MPS 1.4 - APIC platform	0x00000070	0x0000000002
MPS 1.4 - APIC platform	0x00000080	0x0000000010
MPS 1.4 - APIC platform	0x00000092	0x0000000001

MPS 1.4 - APIC platform	0x000000a0	0x0000000002
MPS 1.4 - APIC platform	0x000000c0	0x0000000010
MPS 1.4 - APIC platform	0x000000f0	0x0000000010
i8042prt	0x00000060	0x0000000001
i8042prt	0x00000064	0x0000000001
Serial	0x000003e8	0x0000000007
E100B	0x00003800	0x000000001c
E100B	0x00002020	0x000000001c
Floppy	0x000003f0	0x0000000006
Floppy	0x000003f7	0x0000000001
dac960nt	0x00004000	0x0000000080
dac960nt	0x00005000	0x0000000080
dac960nt	0x00006000	0x0000000080
dac960nt	0x00007000	0x0000000080
dac960nt	0x00008000	0x0000000080
dac960nt	0x00009000	0x0000000080
dac960nt	0x0000a000	0x0000000080
dac960nt	0x0000b000	0x0000000080
symc8XX	0x00003000	0x0000000100
symc8XX	0x00003400	0x0000000100
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	0x0000001c
E100B	0xe5300000	0x0000001c
dac960nt	0xe4410000	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      0xe4302000  0x00000100
symc8XX      0xe4300000  0x00001000
symc8XX      0xe4302400  0x00000100
symc8XX      0xe4301000  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;D:\MSSQL7\BINN
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 5 Stepping 3,
GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=0503
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
Transport: NetBT_E100B2, 00-90-27-23-71-C8, 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: 14,050
SMB's Received: 153
Paged Read Bytes Requested: 0
Non Paged Read Bytes Requested: 0
Cache Read Bytes Requested: 0
Network Read Bytes Requested: 0
Bytes Transmitted: 17,755
SMB's Transmitted: 153
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: 12
Failed Sessions: 0
Reconnects: 0
Core Connects: 0
LM 2.0 Connects: 0
LM 2.x Connects: 0
Windows NT Connects: 12
Server Disconnects: 0
Hung Sessions: 0
Use Count: 0
Failed Use Count: 0
Current Commands: 0
Server File Opens: 2
Server Device Opens: 2
Server Jobs Queued: 1,378,864
Server Session Opens: 1,239,420
Server Sessions Timed Out: 6,619,219
Server Sessions Errored Out: 7,733,362
Server Password Errors: 7,471,205
Server Permission Errors: 5,242,912
Server System Errors: 7,471,205
Server Bytes Sent: 9,007,667,412,795,508
Server Bytes Received: 16,326,046,872,633,445
Server Average Response Time: 6,619,254
Server Request Buffers Needed: 7,536,750
Server Big Buffers Needed: 5,111,840

***** Mylex SCSI configuration
*****

*****
*           MYLEX Disk Array Controller - Configuration Utility
*

```

```

*                               Version 4.78-15
*
*****
CONFIGURATION INFORMATION OF :
=====

3 Channel - 15 Target  DAC1164P  #1    Firmware version
5.06-0-57

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 = 1

Pack 0 : [0:0] [0:1]

SYSTEM DRIVE INFORMATION :
=====

Number of System Drives = 1

```

Sys Drv#	Phy. Size	Raid Level	Eff. Size	Write Policy	State
0	138926 MB	1	69463 MB	Write Thru	Online

Device Information

Chnl/Targ	Vendor	Model	Version	Size	State
0-0	MYLEX	DAC960SX	69464B0 1333	69463 MB	Online
0-1	MYLEX	DAC960SX	69464B0 1333	69463 MB	Online

\*\*\*\*\*

\*\*\*\*

\* MYLEX Disk Array Controller - Configuration Utility

\*

Version 4.78-15

\*

\*\*\*\*\*

\*\*\*\*

CONFIGURATION INFORMATION OF :

=====

3 Channel - 15 Target DAC1164P #2 Firmware version  
5.06-0-57

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	7502	8683 MB	Online
1-3	SEAGATE	ST39102LC	7502	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      7502      8683 MB Online
2-1  SEAGATE ST39102LC      7502      8683 MB Online
2-2  SEAGATE ST39102LC      7503      8683 MB Online
2-3  SEAGATE ST39102LC      7503      8683 MB Online
2-4  SEAGATE ST39102LC      7502      8683 MB Online
2-5  SEAGATE ST39102LC      7502      8683 MB Online
2-10 SEAGATE ST39102LC      7503      8683 MB Online
2-11 SEAGATE ST39102LC      7502      8683 MB Online
2-12 SEAGATE ST39102LC      7502      8683 MB Online
2-13 SEAGATE ST39102LC      7502      8683 MB Online

```

```

*****
****
*           MYLEX Disk Array Controller - Configuration Utility
*
*                               Version 4.78-15
*
*****
****

```

CONFIGURATION INFORMATION OF :  
=====

```

3 Channel - 15 Target DAC1164P #3 Firmware version
5.06-0-57

```

```

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	7502	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 7502 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

```

```

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 7502 8683 MB Online
0-10 SEAGATE ST39102LC 7502 8683 MB Online
0-11 SEAGATE ST39102LC 7502 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

```

```

*****
* MYLEX Disk Array Controller - Configuration Utility
*
* Version 4.78-15
*
*****

```

```

CONFIGURATION INFORMATION OF :
=====

```

```

3 Channel - 15 Target DAC1164P #4 Firmware version
5.06-0-57

```

```

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

```

```

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

```

```

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

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

```

*****
*
* MYLEX Disk Array Controller - Configuration Utility
*
* Version 4.78-15
*
*****

```

CONFIGURATION INFORMATION OF :

```

=====
3 Channel - 15 Target DAC1164P #5 Firmware version
5.06-0-57

```

```

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

```

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
5.06-0-57

```

```

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

```

```

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

```

```

*****
****
* MYLEX Disk Array Controller - Configuration Utility
*
* Version 4.78-15
*
*****
****

```

CONFIGURATION INFORMATION OF :

```

=====
3 Channel - 15 Target DAC1164P #7 Firmware version
5.06-0-57

```

```

Auto Rebuild Management : Disabled
Storage Works Fault Management : Disabled
Rebuild/Add Capacity Rate : 50
Stripe Size : 64K
Cache Segment Size : 8K

```

```

SCSI Transfer Parameters
-----

```

```

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

```

```

SCSI Transfer Parameters
-----
Data Transfer Rate for channel 0: 10 MHz
Data Bus Width for channel 0 : 16 Bit
Command Tags for channel 0 : Enabled

Data Transfer Rate for channel 1: 10 MHz
Data Bus Width for channel 1 : 16 Bit
Command Tags for channel 1 : Enabled

Data Transfer Rate for channel 2: 10 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]

```

```

*****
****
*           MYLEX Disk Array Controller - Configuration Utility
*
*           Version 4.78-15
*
*****
****

```

```

SYSTEM DRIVE INFORMATION :
=====
Number of System Drives = 2

Sys Drv# Phy. Size Raid Level Eff. Size Write Policy State
=====
Online 0 104196 MB 0 104196 MB Write Thru
Online 1 104196 MB 0 104196 MB Write Thru

Device Information
-----
Chnl/Targ Vendor Model Version Size State
-----
0-0 SEAGATE ST19171W B501 8683 MB Online

```

```

CONFIGURATION INFORMATION OF :
=====
3 Channel - 15 Target DAC1164P #8 Firmware version
5.06-0-57

Auto Rebuild Management : Disabled
Storage Works Fault Management : Disabled
Rebuild/Add Capacity Rate : 50
Stripe Size : 64K
Cache Segment Size : 8K

```

```

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

```

SYSTEM DRIVE INFORMATION :

=====

Number of System Drives = 1  
LUN Affinity = Disabled

#	Phy. Size	Raid	Eff. Size	Cache	Packs	Status
=	=====	=====	=====	=====	=====	=====
0	69464 MB	0	69464 MB	Write Back	A...	Onln

SYSTEM DRIVE AFFINITY INFORMATION :

=====

All Affinity (No LUN mapping)

PHYSICAL DRIVE INFORMATION :

=====

Chn	Tgt	Present	Type	Status	Size (MB)	Size (RAW)
===	===	=====	=====	=====	=====	=====
0	0	Absent	00	DEAD	0	00000000
0	1	Absent	00	DEAD	0	00000000
0	2	Absent	00	DEAD	0	00000000
0	3	Absent	00	DEAD	0	00000000
0	4	Absent	00	DEAD	0	00000000
0	5	Absent	00	DEAD	0	00000000
0	6	Absent	00	DEAD	0	00000000
0	8	Absent	00	DEAD	0	00000000
0	9	Absent	00	DEAD	0	00000000
0	a	Absent	00	DEAD	0	00000000
0	b	Absent	00	DEAD	0	00000000
0	c	Absent	00	DEAD	0	00000000
0	d	Absent	00	DEAD	0	00000000
0	e	Absent	00	DEAD	0	00000000
0	f	Absent	00	DEAD	0	00000000
1	0	Present	f1	ONLINE	17366	021eb000
1	1	Present	f1	ONLINE	17366	021eb000
1	2	Present	f1	ONLINE	17366	021eb000
1	3	Present	f1	ONLINE	17366	021eb000
1	4	Absent	00	DEAD	0	00000000
1	5	Absent	00	DEAD	0	00000000
1	6	Absent	00	DEAD	0	00000000
1	8	Absent	00	DEAD	0	00000000
1	9	Absent	00	DEAD	0	00000000
1	a	Absent	00	DEAD	0	00000000
1	b	Absent	00	DEAD	0	00000000
1	c	Absent	00	DEAD	0	00000000
1	d	Absent	00	DEAD	0	00000000
1	e	Absent	00	DEAD	0	00000000
1	f	Absent	00	DEAD	0	00000000
2	0	Absent	00	DEAD	0	00000000

```

*****
****
* MYLEX Disk Array Controller - Configuration Utility
*
* Version 7.11
*
*****
****

```

CONFIGURATION INFORMATION OF :

=====

4 Channel - 16 Target DAC960SX Firmware version 3.33-00

Daughter Card

PHYSICAL PACK INFORMATION :

=====

Number of Packs = 1

Pack A : [1:0] [1:1] [1:2] [1:3]

```

2 1 Absent 00 DEAD 0 00000000
2 2 Absent 00 DEAD 0 00000000
2 3 Absent 00 DEAD 0 00000000
2 4 Absent 00 DEAD 0 00000000
2 5 Absent 00 DEAD 0 00000000
2 6 Absent 00 DEAD 0 00000000
2 8 Absent 00 DEAD 0 00000000
2 9 Absent 00 DEAD 0 00000000
2 a Absent 00 DEAD 0 00000000
2 b Absent 00 DEAD 0 00000000
2 c Absent 00 DEAD 0 00000000
2 d Absent 00 DEAD 0 00000000
2 e Absent 00 DEAD 0 00000000
2 f Absent 00 DEAD 0 00000000
3 0 Absent 00 DEAD 0 00000000
3 1 Absent 00 DEAD 0 00000000
3 2 Absent 00 DEAD 0 00000000
3 3 Absent 00 DEAD 0 00000000
3 4 Absent 00 DEAD 0 00000000
3 5 Absent 00 DEAD 0 00000000
3 6 Absent 00 DEAD 0 00000000
3 8 Absent 00 DEAD 0 00000000
3 9 Absent 00 DEAD 0 00000000
3 a Absent 00 DEAD 0 00000000
3 b Absent 00 DEAD 0 00000000
3 c Absent 00 DEAD 0 00000000
3 d Absent 00 DEAD 0 00000000
3 e Absent 00 DEAD 0 00000000
3 f Absent 00 DEAD 0 00000000

```

CONTROLLER PARAMETERS :  
=====

Hardware

```

Automatic Rebuild Management Enabled
Operational Fault Management Enabled
Disconnect on First Command Disabled

```

Physical

```

Default rebuild rate 50
Controller read ahead Enabled
Super read ahead Disabled
True Verification of Data Disabled
Stripe Size(K bytes) 8
Installation Abort Disabled
Reassign Restricted to 1 blk Enabled
Write Through Verify Enabled
RAID 5 Algorithm Right Asym

```

Disk Side Parameters

```

Per Channel:          0    1    2    3    4
Data Transfer Rate - 20MHz 20MHz 20MHz 20MHz
Command tagging     - Enabl Enabl Enabl Enabl
Data bus width      - 16bit 16bit 16bit 16bit

Per Device:
Elevator           - Disab
Coalescing         - Disab
Queue Limit        - 0

Global:
Max IOPs           - 64
Spin Up Option     - Automatic / 2 / 6 / 0

```

Host Side

```

Disable Wide Operation Disabled
Vendor Unique TUR Disabled
Disable CC for Invalid LUN Enabled
No Pause on ctrlr not ready Disabled
On Queue Full give Busy Disabled

```

Serial port 0

Active - Active

```

Conservative Cache Disabled
Auto Failback Disabled
Force Simplex Enabled
Host Bus Reset Delay 0
Ctrlr Pres/Flt Signals Disabled
Ctrlr Pres/Flt Select A
Simplex no RSTCOM Disabled

```

Fibre

```

PCI Latency Control Short
Frame Control Long
Smart Large Transfers Enabled
Port 0 Disabled
Port 0 ID 0
Port 1 Disabled
Port 1 ID 0

```

```

*****
****
*           MYLEX Disk Array Controller - Configuration Utility
*

```



\* Version 7.11  
 \*  
 \*\*\*\*\*  
 \*\*\*\*\*

CONFIGURATION INFORMATION OF :  
 =====

4 Channel - 16 Target DAC960SX Firmware version 3.33-00

Daughter Card

PHYSICAL PACK INFORMATION :  
 =====

Number of Packs = 1

Pack A : [1:0] [1:1] [1:2] [1:3]

SYSTEM DRIVE INFORMATION :  
 =====

Number of System Drives = 1  
 LUN Affinity = Disabled

#	Phy. Size	Raid	Eff. Size	Cache	Packs	Status
=	=====	=====	=====	=====	=====	=====
0	69464 MB	0	69464 MB	Write Back	A...	Onln

SYSTEM DRIVE AFFINITY INFORMATION :  
 =====

All Affinity (No LUN mapping)

PHYSICAL DRIVE INFORMATION :  
 =====

Chn	Tgt	Present	Type	Status	Size (MB)	Size (RAW)
==	==	=====	=====	=====	=====	=====
0	0	Absent	00	DEAD	0	00000000
0	1	Absent	00	DEAD	0	00000000
0	2	Absent	00	DEAD	0	00000000
0	3	Absent	00	DEAD	0	00000000
0	4	Absent	00	DEAD	0	00000000
0	5	Absent	00	DEAD	0	00000000
0	6	Absent	00	DEAD	0	00000000
0	8	Absent	00	DEAD	0	00000000
0	9	Absent	00	DEAD	0	00000000
0	a	Absent	00	DEAD	0	00000000
0	b	Absent	00	DEAD	0	00000000

0	c	Absent	00	DEAD	0	00000000
0	d	Absent	00	DEAD	0	00000000
0	e	Absent	00	DEAD	0	00000000
0	f	Absent	00	DEAD	0	00000000
1	0	Present	f1	ONLINE	17366	021eb000
1	1	Present	f1	ONLINE	17366	021eb000
1	2	Present	f1	ONLINE	17366	021eb000
1	3	Present	f1	ONLINE	17366	021eb000
1	4	Absent	00	DEAD	0	00000000
1	5	Absent	00	DEAD	0	00000000
1	6	Absent	00	DEAD	0	00000000
1	8	Absent	00	DEAD	0	00000000
1	9	Absent	00	DEAD	0	00000000
1	a	Absent	00	DEAD	0	00000000
1	b	Absent	00	DEAD	0	00000000
1	c	Absent	00	DEAD	0	00000000
1	d	Absent	00	DEAD	0	00000000
1	e	Absent	00	DEAD	0	00000000
1	f	Absent	00	DEAD	0	00000000
2	0	Absent	00	DEAD	0	00000000
2	1	Absent	00	DEAD	0	00000000
2	2	Absent	00	DEAD	0	00000000
2	3	Absent	00	DEAD	0	00000000
2	4	Absent	00	DEAD	0	00000000
2	5	Absent	00	DEAD	0	00000000
2	6	Absent	00	DEAD	0	00000000
2	8	Absent	00	DEAD	0	00000000
2	9	Absent	00	DEAD	0	00000000
2	a	Absent	00	DEAD	0	00000000
2	b	Absent	00	DEAD	0	00000000
2	c	Absent	00	DEAD	0	00000000
2	d	Absent	00	DEAD	0	00000000
2	e	Absent	00	DEAD	0	00000000
2	f	Absent	00	DEAD	0	00000000
3	0	Absent	00	DEAD	0	00000000
3	1	Absent	00	DEAD	0	00000000
3	2	Absent	00	DEAD	0	00000000
3	3	Absent	00	DEAD	0	00000000
3	4	Absent	00	DEAD	0	00000000
3	5	Absent	00	DEAD	0	00000000
3	6	Absent	00	DEAD	0	00000000
3	8	Absent	00	DEAD	0	00000000
3	9	Absent	00	DEAD	0	00000000
3	a	Absent	00	DEAD	0	00000000
3	b	Absent	00	DEAD	0	00000000
3	c	Absent	00	DEAD	0	00000000
3	d	Absent	00	DEAD	0	00000000
3	e	Absent	00	DEAD	0	00000000
3	f	Absent	00	DEAD	0	00000000

CONTROLLER PARAMETERS :  
 =====

Hardware

Automatic Rebuild Management	Enabled
Operational Fault Management	Enabled
Disconnect on First Command	Disabled

Physical

Default rebuild rate	50
Controller read ahead	Enabled
Super read ahead	Disabled
True Verification of Data	Disabled
Stripe Size(K bytes)	8
Installation Abort	Disabled
Reassign Restricted to 1 blk	Enabled
Write Through Verify	Enabled
RAID 5 Algorithm	Right Asym

Disk Side Parameters

Per Channel:	0	1	2	3	4
Data Transfer Rate -	20MHz	20MHz	20MHz	20MHz	
Command tagging -	Enabl	Enabl	Enabl	Enabl	
Data bus width -	16bit	16bit	16bit	16bit	
Per Device:					
Elevator -	Disab				
Coalescing -	Disab				
Que Limit -	0				
Global:					
Max IOPs -	64				
Spin Up Option -	Automatic	/	2	/	6 / 0

Host Side

Disable Wide Operation	Disabled
Vendor Unique TUR	Disabled
Disable CC for Invalid LUN	Enabled
No Pause on ctrlr not ready	Disabled
On Queue Full give Busy	Disabled

Serial port 0

Active - Active

Conservative Cache	Disabled
Auto Failback	Disabled
Force Simplex	Enabled
Host Bus Reset Delay	0
Ctrlr Pres/Flt Signals	Disabled

Ctrlr Pres/Flt Select	A
Simplex no RSTCOM	Disabled

Fibre

PCI Latency Control	Short
Frame Control	Long
Smart Large Transfers	Enabled
Port 0	Disabled
Port 0 ID	0
Port 1	Disabled
Port 1 ID	0

**This section discloses hardware information of the Primergy 460 client system.**

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: 666,087 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: 1,803,396 KB  
Serial Number: 2C85 - 6841  
Bytes per cluster: 512  
Sectors per cluster: 8  
Filename length: 255

Memory Report

-----

Handles: 16,978  
Threads: 127  
Processes: 16

Physical Memory (K)

Total: 523,696  
Available: 443,200  
File Cache: 19,864

Kernel Memory (K)

Total: 29,636  
Paged: 15,640  
Nonpaged: 13,996

Commit Charge (K)

Total: 56,428  
Limit: 1,543,632  
Peak: 57,368

Pagefile Space (K)

Total: 1,048,576  
Total in use: 2,436  
Peak: 2,448

C:\pagefile.sys

Total: 1,048,576  
Total in use: 2,436  
Peak: 2,448

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			C:\WINNT\system32\netdde.exe	
EventLog (Event log)	Running		Service Account Name: LocalSystem	
(Automatic)			Error Severity: Normal	
C:\WINNT\system32\services.exe			Service Flags: Shared Process	
Service Account Name: LocalSystem			Net Logon (RemoteValidation)	Stopped (Manual)
Error Severity: Normal			C:\WINNT\System32\lsass.exe	
Service Flags: Shared Process			Service Account Name: LocalSystem	
Server	Running		Error Severity: Normal	
(Automatic)			Service Flags: Shared Process	
C:\WINNT\System32\services.exe			Service Dependencies:	
Service Account Name: LocalSystem			LanmanWorkstation	
Error Severity: Normal			LmHosts	
Service Flags: Shared Process			NT LM Security Support Provider	Running (Manual)
Group Dependencies:			C:\WINNT\System32\SERVICES.EXE	
TDI			Service Account Name: LocalSystem	
Workstation (NetworkProvider)	Running		Error Severity: Normal	
(Automatic)			Service Flags: Shared Process	
C:\WINNT\System32\services.exe			Plug and Play (PlugPlay)	Stopped (Manual)
Service Account Name: LocalSystem			C:\WINNT\system32\services.exe	
Error Severity: Normal			Service Account Name: LocalSystem	
Service Flags: Shared Process			Error Severity: Normal	
Group Dependencies:			Service Flags: Shared Process	
TDI			Protected Storage	Running
License Logging Service	Stopped	(Manual)	(Automatic)	
C:\WINNT\System32\llssrv.exe			C:\WINNT\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	
TCP/IP NetBIOS Helper	Stopped	(Manual)	Service Dependencies:	
C:\WINNT\System32\services.exe			RpcSs	
Service Account Name: LocalSystem			Directory Replicator	Stopped (Manual)
Error Severity: Normal			C:\WINNT\System32\lmrepl.exe	
Service Flags: Shared Process			Service Account Name: LocalSystem	
Group Dependencies:			Error Severity: Normal	
NetworkProvider			Service Flags: Own Process	
Messenger	Running		Service Dependencies:	
(Automatic)			LanmanWorkstation	
C:\WINNT\System32\services.exe			LanmanServer	
Service Account Name: LocalSystem			Remote Command Service	Running
Error Severity: Normal			(Automatic)	
Service Flags: Shared Process			c:\benchcrf\rsys.exe	
Service Dependencies:			Service Account Name: LocalSystem	
LanmanWorkstation			Error Severity: Normal	
NetBios			Service Flags: Own Process, Interactive	
Network DDE (NetDDEGroup)	Stopped	(Manual)	Remote Procedure Call (RPC) Locator	Stopped (Manual)
C:\WINNT\system32\netdde.exe			C:\WINNT\System32\LOCATOR.EXE	
Service Account Name: LocalSystem			Service Account Name: LocalSystem	
Error Severity: Normal			Error Severity: Normal	
Service Flags: Shared Process			Service Flags: Own Process	
Service Dependencies:			Service Dependencies:	
NetDDEDSDM			LanmanWorkstation	
Network DDE DSDM	Stopped	(Manual)	Rdr	

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)
Service Flags: File System Driver, Shared Process			Stopped (Disabled)
Group Dependencies:			Error Severity: Ignore
SCSI CDROM Class			Service Flags: Kernel Driver, Shared Process
Cdrom (SCSI CDROM Class)	Running	(System)	Fastfat (Boot file system)
Error Severity: Ignore			Stopped (Disabled)
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal
Group Dependencies:			Service Flags: File System Driver, Shared Process
SCSI miniport			Fdl6_700 (SCSI miniport)
Changer (Filter)	Stopped	(System)	Stopped (Disabled)
Error Severity: Ignore			Error Severity: Normal
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process
cirrus (Video)	Running	(System)	Fd7000ex (SCSI miniport)
Error Severity: Normal			Stopped (Disabled)
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal
Cpqarray (SCSI miniport)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process
Error Severity: Normal			Fd8xx (SCSI miniport)
Service Flags: Kernel Driver, Shared Process			Stopped (Disabled)
cpqfw2e (SCSI miniport)	Stopped	(Disabled)	Error Severity: Normal
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process
Service Flags: Kernel Driver, Shared Process			flashpnt (SCSI miniport)
dac960nt (SCSI miniport)	Stopped	(Disabled)	Stopped (Disabled)
Error Severity: Normal			Error Severity: Normal
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process
dce376nt (SCSI miniport)	Stopped	(Disabled)	Floppy (Primary disk)
Error Severity: Normal			Running (System)
Service Flags: Kernel Driver, Shared Process			Error Severity: Ignore
Delldsa (SCSI miniport)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process
Error Severity: Normal			Ftdisk (Filter)
Service Flags: Kernel Driver, Shared Process			Stopped (Disabled)
Dell_DGX (Video)	Stopped	(Disabled)	Error Severity: Ignore
Error Severity: Ignore			Service Flags: Kernel Driver, Shared Process
Service Flags: Kernel Driver, Shared Process			i8042 Keyboard and PS/2 Mouse Port Driver (Keyboard Port)
Disk (SCSI Class)	Running	(Boot)	Running (System)
Error Severity: Ignore			System32\DRIVERS\i8042prt.sys
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal
Group Dependencies:			Service Flags: Kernel Driver, Shared Process
SCSI miniport			Inport (Pointer Port)
Diskperf (Filter)	Stopped	(Disabled)	Stopped (Disabled)
Error Severity: Normal			Error Severity: Ignore
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process
DptScsi (SCSI miniport)	Stopped	(Disabled)	Jazzg300 (Video)
Error Severity: Normal			Stopped (Disabled)
Service Flags: Kernel Driver, Shared Process			Error Severity: Ignore
dtc329x (SCSI miniport)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process
Error Severity: Normal			Jazzg364 (Video)
Service Flags: Kernel Driver, Shared Process			Stopped (Disabled)
Intel EtherExpress PRO Adapter (NDIS)	Running		Error Severity: Ignore
(Automatic)			Service Flags: Kernel Driver, Shared Process
C:\WINNT\System32\drivers\ei100bnt.sys			Jzvxl484 (Video)
Error Severity: Normal			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
			mga (Video)
			Stopped (Disabled)
			Error Severity: Ignore
			Service Flags: Kernel Driver, Shared Process

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)
Error Severity: Normal			Running (System)
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal
mkecr5xx (SCSI miniport)	Stopped	(Disabled)	Service Flags: File System Driver, Shared Process
Error Severity: Normal			Ntfs (File system)
Service Flags: Kernel Driver, Shared Process			Running (Disabled)
Modem (Extended base)	Stopped	(Manual)	Error Severity: Normal
Error Severity: Ignore			Service Flags: File System Driver, Shared Process
Service Flags: Kernel Driver, Shared Process			Null (Base)
Mouse Class Driver (Pointer Class)	Running	(System)	Running (System)
System32\DRIVERS\mouclass.sys			Error Severity: Normal
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process
Service Flags: Kernel Driver, Shared Process			Oliscsi (SCSI miniport)
Msfs (File system)	Running	(System)	Error Severity: Normal
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process
Service Flags: File System Driver, Shared Process			Parallel (Extended base)
Mup (Network)	Running	(Manual)	Stopped (Automatic)
C:\WINNT\System32\drivers\mup.sys			Error Severity: Ignore
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process
Service Flags: File System Driver, Shared Process			Service Dependencies:
Ncr53c9x (SCSI miniport)	Stopped	(Disabled)	Parport
Error Severity: Normal			Group Dependencies:
Service Flags: Kernel Driver, Shared Process			Parallel arbitrator
ncr77c22 (Video)	Stopped	(Disabled)	Parport (Parallel arbitrator)
Error Severity: Ignore			Stopped (Automatic)
Service Flags: Kernel Driver, Shared Process			Error Severity: Ignore
Ncr700 (SCSI miniport)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process
Error Severity: Normal			ParVdm (Extended base)
Service Flags: Kernel Driver, Shared Process			Stopped (Automatic)
Ncr710 (SCSI miniport)	Stopped	(Disabled)	Error Severity: Ignore
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process
Service Flags: Kernel Driver, Shared Process			Service Dependencies:
Microsoft NDIS System Driver (NDIS)	Running	(System)	Parport
Error Severity: Normal			Group Dependencies:
Service Flags: Kernel Driver, Shared Process			Parallel arbitrator
NetBIOS Interface (NetBIOSGroup)	Running	(Manual)	PCIDump (PCI Configuration)
C:\WINNT\System32\drivers\netbios.sys			Stopped (System)
Error Severity: Normal			Error Severity: Ignore
Service Flags: File System Driver, Shared Process			Service Flags: Kernel Driver, Shared Process
Group Dependencies:			Pcmcia (System Bus Extender)
TDI			Stopped (Disabled)
WINS Client (TCP/IP) (PNP_TDI)	Running	(Automatic)	Error Severity: Normal
(Automatic)			Service Flags: Kernel Driver, Shared Process
C:\WINNT\System32\drivers\netbt.sys			PnP ISA Enabler Driver (Base)
Error Severity: Normal			Stopped (System)
Service Flags: Kernel Driver, Shared Process			Error Severity: Ignore
Service Dependencies:			Service Flags: Kernel Driver, Shared Process
Tcpip			psidisp (Video)
NetDetect	Stopped	(Manual)	Stopped (Disabled)
			Error Severity: Ignore
			Service Flags: Kernel Driver, Shared Process
			Q110wnt (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			Error Severity: Normal		
Rdr (Network)	Running	(Manual)	Service Flags: Kernel Driver, Shared Process		
C:\WINNT\System32\drivers\rdr.sys			T128 (SCSI miniport)	Stopped	(Disabled)
Error Severity: Normal			Error Severity: Normal		
Service Flags: File System Driver, Shared Process			Service Flags: Kernel Driver, Shared Process		
s3 (Video)	Stopped	(Disabled)	T13B (SCSI miniport)	Stopped	(Disabled)
Error Severity: Ignore			Error Severity: Normal		
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process		
Scsiprnt (Extended base)	Stopped		TCP/IP Service (PNP_TDI)	Running	
(Automatic)			(Automatic)		
Error Severity: Ignore			C:\WINNT\System32\drivers\tcpip.sys		
Service Flags: Kernel Driver, Shared Process			Error Severity: Normal		
Group Dependencies:			Service Flags: Kernel Driver, Shared Process		
SCSI miniport			tga (Video)	Stopped	(Disabled)
Scsiscan (SCSI Class)	Running	(System)	Error Severity: Ignore		
Error Severity: Ignore			Service Flags: Kernel Driver, Shared Process		
Service Flags: Kernel Driver, Shared Process			tmv1 (SCSI miniport)	Stopped	(Disabled)
Group Dependencies:			Error Severity: Normal		
SCSI miniport			Service Flags: Kernel Driver, Shared Process		
Serial (Extended base)	Stopped		Ultra124 (SCSI miniport)	Stopped	(Disabled)
(Automatic)			Error Severity: Normal		
Error Severity: Ignore			Service Flags: Kernel Driver, Shared Process		
Service Flags: Kernel Driver, Shared Process			Ultra14f (SCSI miniport)	Stopped	(Disabled)
Sermouse (Pointer Port)	Stopped	(Disabled)	Error Severity: Normal		
Error Severity: Ignore			Service Flags: Kernel Driver, Shared Process		
Service Flags: Kernel Driver, Shared Process			Ultra24f (SCSI miniport)	Stopped	(Disabled)
Sfloppy (Primary disk)	Stopped	(System)	Error Severity: Normal		
Error Severity: Ignore			Service Flags: Kernel Driver, Shared Process		
Service Flags: Kernel Driver, Shared Process			update (Base)	Stopped	(System)
Group Dependencies:			Error Severity: Ignore		
SCSI miniport			Service Flags: Kernel Driver, Shared Process		
Simbad (Filter)	Stopped	(Disabled)	v7vram (Video)	Stopped	(Disabled)
Error Severity: Normal			Error Severity: Ignore		
Service Flags: Kernel Driver, Shared Process			Service Flags: Kernel Driver, Shared Process		
slcd32 (SCSI miniport)	Stopped	(Disabled)	VgaSave (Video Save)	Stopped	(System)
Error Severity: Normal			C:\WINNT\System32\drivers\vga.sys		
Service Flags: Kernel Driver, Shared Process			Error Severity: Ignore		
Sparrow (SCSI miniport)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process		
Error Severity: Normal			VgaStart (Video Init)	Stopped	(System)
Service Flags: Kernel Driver, Shared Process			C:\WINNT\System32\drivers\vga.sys		
Spock (SCSI miniport)	Stopped	(Disabled)	Error Severity: Ignore		
Error Severity: Normal			Service Flags: Kernel Driver, Shared Process		
Service Flags: Kernel Driver, Shared Process			Wd33c93 (SCSI miniport)	Stopped	(Disabled)
Srv (Network)	Running	(Manual)	Error Severity: Normal		
C:\WINNT\System32\drivers\srv.sys			Service Flags: Kernel Driver, Shared Process		
Error Severity: Normal			wd90c24a (Video)	Stopped	(Disabled)
Service Flags: File System Driver, Shared Process			Error Severity: Ignore		
symc810 (SCSI miniport)	Stopped	(Disabled)	Service Flags: Kernel Driver, Shared Process		
Error Severity: Normal			wdvga (Video)	Stopped	(Disabled)
Service Flags: Kernel Driver, Shared Process			Error Severity: Ignore		
symc8XX (SCSI miniport)	Running	(Boot)	Service Flags: Kernel Driver, Shared Process		
C:\WINNT\system32\drivers\symc8XX.sys			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 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 0x00001000
E100B                       4      4 0x00000000
Floppy                      6      6 0x00000000
symc8XX                     7      7 0x00000000
```

```
-----
----
Devices                Physical Address Length
-----
-----
MPS 1.4 - APIC platform      0x00000000 0x0000000010
MPS 1.4 - APIC platform      0x00000020 0x0000000002
MPS 1.4 - APIC platform      0x00000040 0x0000000004
MPS 1.4 - APIC platform      0x00000048 0x0000000004
MPS 1.4 - APIC platform      0x00000061 0x0000000001
MPS 1.4 - APIC platform      0x00000070 0x0000000002
MPS 1.4 - APIC platform      0x00000080 0x0000000010
MPS 1.4 - APIC platform      0x00000092 0x0000000001
MPS 1.4 - APIC platform      0x000000a0 0x0000000002
MPS 1.4 - APIC platform      0x000000c0 0x0000000010
MPS 1.4 - APIC platform      0x000000f0 0x0000000010
i8042prt                    0x00000060 0x0000000001
i8042prt                    0x00000064 0x0000000001
E100B                       0x0000e400 0x000000001c
E100B                       0x0000e800 0x000000001c
E100B                       0x0000d400 0x000000001c
E100B                       0x0000d000 0x000000001c
E100B                       0x0000c800 0x000000001c
```

```

E100B          0x0000e000  0x000000001c
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  0xfef00000  0x00000400
E100B                   0xfecfd000  0x0000001c
E100B                   0xfecfe000  0x0000001c
E100B                   0xf7ffe000  0x0000001c
E100B                   0xf7ffd000  0x0000001c
E100B                   0xf7ffc000  0x0000001c
E100B                   0xfecfc000  0x0000001c
symc8XX                 0xfe8ff800  0x00000100
symc8XX                 0xfe8fd000  0x00001000
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: 38  
Server Device Opens: 0  
Server Jobs Queued: 0  
Server Session Opens: 1  
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: 822,199  
Server Bytes Received: 12,166  
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	1800	tpm C	22349	tpm C/W	12.42		
Table	Rows	Data	Index	5% Space	8H Space	Total Space	
Warehouse	1,920	208	40	12		260	
District	19,200	2,136	56	110		2,302	
Item	100,000	9,528	80	221		9,829	
New-order	17,280,000	273,208	752		38,400	312,360	
History	57,600,000	3,200,056	0		493,491	3,693,547	
Orders	57,600,000	1,765,520	975,168		422,650	3,163,338	
Customer	57,600,000	41,890,912	2,689,928	1,025,359		45,606,199	
Order-line	575,998,743	35,999,928	89,664		5,565,489	41,655,081	
Stock	192,000,000	61,440,000	137,704	1,416,287		62,993,991	
<b>Totals</b>		144,581,496	3,893,392	2,441,999	6,520,030	157,436,908	
Segment	LogDev Cnt.	Seg. Size	Needed	Overhead	Not Needed		
misc	6	117,350,400	49,325,084	493,251	67,532,065		
customer/stock	6	56,524,800	109,686,192	1,096,862	-54,258,254		
<b>Totals</b>		173,875,200	159,011,277	1,590,113	13,273,811		
<b>Dynamic spa</b>	39,859,435	Sum of Data for Order, Order-Line and History (excluding free extents)					
<b>Static space</b>	112,647,555	Data + Index + 5% Space + Overhead - Dynamic space					
<b>Free space</b>	8,094,399	Total Seg. Size - Dynamic Space - Static Space - Not Needed					
<b>Daily growth</b>	7,423,488	(Dynamic space/W * 62.5)* tpmC					
<b>Daily spread</b>	-3,040,832	Free space - 1.5 * Daily growth (zero if negative)					
<b>180 day (KB)</b>	1,448,875,337	Static space + 180 (daily growth + daily spread)					
<b>180 day (GB)</b>	1,381,76	Excludes OS, Paging and RDBMS Logs					
<b>Log size (MB)</b>	45,000	Total size of log file					
<b>% Log used</b>	31,7220	% of log file used during entire run					
<b>Total N-O Tx</b>	2871816	Total count of NO transactions during entire run					
<b>Log per N-O Tx</b>	5,0900	K per New-Order transaction					
<b>8 Hour Log (GB)</b>	52,07						
<b>Disk Capacity</b>	<b>MB</b>	<b>GB</b>	<b>sk's needed</b>	<b>disks priced</b>			
9 GB 10000 rpm	8,676	8,47	163,08	192			
180 day (GB)		1,381,76					
<b>Disk Capacity</b>	<b>MB</b>	<b>GB</b>	<b>sk's needed</b>	<b>disks priced</b>			
18 GB 10000 rpm	17,366	16,96	3,07	4+4			
8 Hour Log (RAID 1)		52,07					

**Price/Performance Spreadsheet**    *The following detailed worksheet was used to calculate the price/performance of the system.*

Description	Part Number	Third Party	Unit Price	Qty	Extended Price	\$/WATT Price
<b>Base System</b>						
Pentium III Xeon Processor 400MHz/2MB	SNP/SY4K60A/101P	Brand	1	\$11,284	1	\$11,284
2 Memory Board	SNP/SY-F182E4K3P	Brand	1	\$7,988	4	\$31,832
Memory 512MB EDD/DIMM1	SNP/SY-F182E3E1P	Brand	1	\$788	1	\$788
DAT Drive DDS3 12GB, SE SCSI	SNP/SY-F1540E54P	Brand	1	\$2,189	8	\$17,512
Fast-Ether-Ethernet-Port/100+-Server (PCI)	SNP/SY-F1730E1P	Brand	1	\$1,158	1	\$1,158
FP 9GB/10K/LVD/SCSI, Ht Plug	SNP/SY-F1888E501P	Brand	1	\$111	1	\$111
Power Supply (add)	SNP/SY-F1888E109P	Brand	1	\$688	1	\$688
Keyboard	SNP/PSE234E1P	Brand	1	\$463	1	\$463
Cursor/Pad	SNP/SY-F1888E213P	Brand	1	\$25	1	\$25
Monitor MD11510	SNP/SY-F1888E213P	Brand	1	\$34	1	\$34
	SNP/SY-F1888E213P	Brand	1	\$219	1	\$219
<b>Server Hardware Subtotal</b>						
						\$51,394
<b>RAID Controller:3 Chan, 320k, LVD</b>						
RAID Controller:3 Chan, 320k, LVD (100% spare)	SNP/SY-D1124V1	Brand	1	\$2,838	7	\$19,866
FP 9GB/10K/LVD/SCSI, Ht Plug	SNP/SY-F1124V1	Brand	1	\$2,838	2	\$5,676
FP 9GB/10K/LVD/SCSI, Ht Plug (100% spare)	SNP/SY-F1888E109P	Brand	1	\$688	192	\$185,856
FP 9GB/10K/LVD/SCSI, Ht Plug (100% spare)	SNP/SY-F1888E109P	Brand	1	\$688	20	\$19,680
Power Supply \$40 ES1 demand stack	SNP/SY4K538V201P	Brand	1	\$2,382	18	\$42,516
Power Supply 502, 2H-hot, 400w, 60A/B	SNP/SY-F182E180	Brand	1	\$7,394	2	\$14,788
FP 18GB SCSI 3-WIDE SCSI, Ht Plug	SNP/SY-F132E180	Brand	1	\$1,474	8	\$11,792
Cursor/Pad, Power Supply 502 (1N)	SNP/SY-F1478E30P	Brand	1	\$34	2	\$68
SCSI SE/DE Converter - 3 channels	SNP/SY-F1478E30P	Brand	1	\$1,032	1	\$1,032
SCSI Cable HD-D5m	T26139-V627-V205	Brand	1	\$85	2	\$170
SCSI Cable UH-D LVD 5m	SNP/SY-F1947L50P	Brand	1	\$147	18	\$2,646
<b>Storage Subtotal</b>						
						\$304,070
<b>Mount Server+Storage</b>						
Power Supply 400 FS/PL 460	SNP/SY4K482V714A	Brand	1	\$2,316	4	\$9,264
PL 460MHz	SNP/SY-F1727E40A	Brand	1	\$1,474	4	\$5,896
Keyboard	SNP/SY-F182E180	Brand	1	\$25	4	\$100
Cursor/Pad, International	SNP/SY-F1699B153A	Brand	1	\$34	4	\$136
Memory 256MB SDRAM/100MHz	SNP/SY-F1867E504A	Brand	1	\$1,221	4	\$4,884
Memory 128MB SDRAM/100MHz	SNP/SY-F1867E503A	Brand	1	\$889	8	\$4,712
HD 4GB, SCSI-3, Ht Plug	SNP/SY-F1724E40	Brand	1	\$675	4	\$2,020
Fast-Ether-Ethernet-Port/100+-Server (PCI)	SNP/SY-F1888E501A	Brand	1	\$111	24	\$2,664
Monitor MD11510	SNP/SY-F1888E501A	Brand	1	\$219	4	\$876
						\$825
<b>Client Hardware Subtotal</b>						
						\$30,592
<b>Microsoft Windows NT-Server Enterprise Edition 4.0, Ind 25 CALs</b>						
MS SQL-Server 7.0 Enterprise Edition Utiln License	Microsoft	Microsoft	2	\$3,999	1	\$3,999
<b>Server Software Subtotal</b>						
						\$28,999
<b>Microsoft Windows NT-Server 4.0, Ind 5 CALs</b>						
Tuxedo 6.4 Core Functionality Services for NT	Microsoft	Microsoft	2	\$809	4	\$3,236
Microsoft Visual C++-Professional 5.0	Microsoft	BEA	4	\$3,000	4	\$12,000
		Microsoft	2	\$499	1	\$499
<b>Client Software Subtotal</b>						
						\$15,735
<b>Complex 8-port 10BaseT Hub (8+1) ports (Ind, 1 2500033</b>						
Kingston Switch 8 port 10/100 Ind, 10% spare	732084	Kingston	5	\$675	3	\$1,515
<b>User Creativity Subtotal</b>						
						\$73,330
<b>Subtotal</b>						
						\$271,109
<b>Five-Year Cost of Ownership</b>						
						\$577,493.00
						22,349.47
						\$25,894

# Appendix E - Price Quotations



ENTERPRISE MIDDLEWARE SOLUTIONS

December 14, 1998

Miguel Isenberg  
Database Benchmarking  
ICP CS PS DS 51  
Siemens AG  
Paderborne, Germany

Fax: +49-5251-8-15149

Dear Mr. Isenberg:

Per your request I am enclosing the pricing information regarding TUXEDO 6.x that you requested. This pricing applies to Tuxedo 6.1, 6.2, 6.3 and 6.4. 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 classified in one of 5 tiers based on CPU type and capacity. Xeon systems with 2 CPU 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 basic level of middleware support for distributed computing, and is best used by organizations with substantial resources and knowledge for advanced distributed computing implementations.

TUX-CFS prices are server only and are based on the overall performance characteristics of the server and uses the same five tier computer classification as TUXEDO 6.x. Prices range from \$3,000 for Tier 1 to \$250,000 for Tier 5. Under this pricing option EVERY system running TUX-CFS at the user site must have a TUXEDO license installed and pay the appropriate per server license fees.

### **BEA Tux/CFS Unlimited User License Fees Per Server**

<b>Unlimited User License fees per server</b>	<b>Number of Users</b>	<b>Dollar Amount</b>	<b>Maintenance (5 x 8) per Year</b>	<b>Maintenance (7 x 24) per Year</b>
Tier 1 -- PC Servers with 1 or 2 CPUs, entry level RISC Uni-processor workstations and servers	Unlimited	\$3,000.00	\$450.00	\$560.00
Tier 2 -- PC Servers with 3 or 4 CPUs, Midrange RISC Uni-processor servers and workstations	Unlimited	\$12,000.00	\$1,800.00	\$2,640.00
Tier 3 -- Midrange Multiprocessors, up to 8 CPUs per system capacity	Unlimited	\$30,000.00	\$4,500.00	\$6,600.00

10/3/197

BEA SYSTEMS, INC.

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	\$55,000.00

**Intel based server tier classifications:**

Platform	Operating System	Tier 1	Tier 1	Tier 2	Tier 3	Tier 3
Intel Pentium/ Pentium Pro PCs	Interactive R3.2 ESIX SVR 4.0 SCO UNIX 3.2.2 and 3.2.4 SCO CDDT 2.x,3.x Solaris x86 2.X UnixWare, Windows NT 3.5/4.0	All 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



**Software House International**  
 Pricing Proposal

Quotation #SI-881206-57316  
 12/28/98

**SIEMENS**

Sabharwal

Price Guard for Epic Days

Phone: Fax: 408-784-9100

**S&I Account Entry: Matthew O. Magrino**

Telephones: (408) 522-4105  
 Fax: (408) 525-1222

**References:**

Product	Part #	Qty	Unit	Unit Price	Total
Combox 10B, Spot 110	Z97031	2500		\$28.75	\$71,875.00
8 Input B Port 107105	T13134	3		\$485.00	\$1,455.00
Switch					
<b>Total</b>					<b>\$73,330.00</b>

Additional Comments:

10/10/98 0852 212 859

TUHQ 8800H 881111111111

PR:121 0161-660 110

One Microsoft Way  
Redmond, WA 98052-6399

Fax 425 336 1340  
<http://www.microsoft.com/>

**Microsoft**

December 21, 1998

Mr. Franz-Josef Balhe  
Siemens Nixdorf Informationssysteme AG  
Heinz-Nixdorf-Ring 1  
D-33106 Paderborn  
Germany

Via FAX # 011-49-5251-820409


Dear Mr. Balhe,

Here is the information you requested regarding US pricing of certain Microsoft products:

Microsoft SQL Server 7.0, Enterprise Edition, unlimited user license	\$28999
Microsoft Windows NT Server, Enterprise Edition 4.0, incl 25 CALs	\$3999
Windows NT Server 4.0, incl 5 CALs	\$809
Visual C++ Professional 5.0	\$499
5-yr maintenance for above software @ \$2095/yr	\$10475

The prices quoted above are valid for the next 60 days. Please let me know if I can be of any further assistance.

Sincerely,

  
Sid Arora  
Product Manager, Microsoft SQL Server  
Applications Marketing

\*\*\* TOTAL PRICE: 03 \*\*\*

One Microsoft Way  
Redmond, WA 98052-6399

1-800-425-5049 FAX  
http://www.microsoft.com/

**Microsoft**

December 21, 1998

Mr. Franz-Josef Balke  
Siemens Nixdorf Informationssysteme AG  
Heinz-Nixdorf Ring 1  
D-33106 Paderborn  
Germany

Via FAX # 011-49-5251-820409

Dear Mr. Balke,


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 Primary: 870, 4-processor, Pentium II Xeon-based, 450 MHz  
Test results: 22300 ipm/C @ \$26/ipm/C approximately

Microsoft hereby grants SNI permission to disclose these results and acknowledge that SNI has formally requested permission to do so in accordance with the license agreement for Microsoft SQL Server software.

Best Regards,



  
Sid Aurora  
Product Manager, Microsoft SQL Server  
Applications Marketing



# Appendix F - Attestation Letter

12-22-1998 10:41AM FROM

TO 01149525180447 P. 01



**Information Paradigm**



Sponsor: Franz-Josef Bathe  
 Siemens AG  
 Heinz-Nixdorf-Ring 1  
 D-33106 Paderborn, Germany

December 21, 1998

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	New Order 90% Response Time	pmC
Server: Siemens Primergy 870-40				
4 x Pentium II Xeon ( 450 Mhz )	4.0 GB (2 MB L2 cache per processor)	193 x 9 GB 8 x 18 GB	.91 Seconds	22,349.47
Four Clients: Siemens Primergy 470 ( Specification for each )				
2 x Pentium II (450 Mhz)	512 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 3.3 of the benchmark. The following verification items were given special attention:

- The transactions were correctly implemented
- The database records were the proper size
- The database was properly scaled and populated
- The ACID properties were met
- Input data was generated according to the specified percentages

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

- 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,



Francois Raab  
President

Primergy 870-4

1979 North Franklin Street - Colorado Springs, CO 80903-2527 • Office: 719/473-7555 • Fax: 719/473-7554

TDTRL P.02