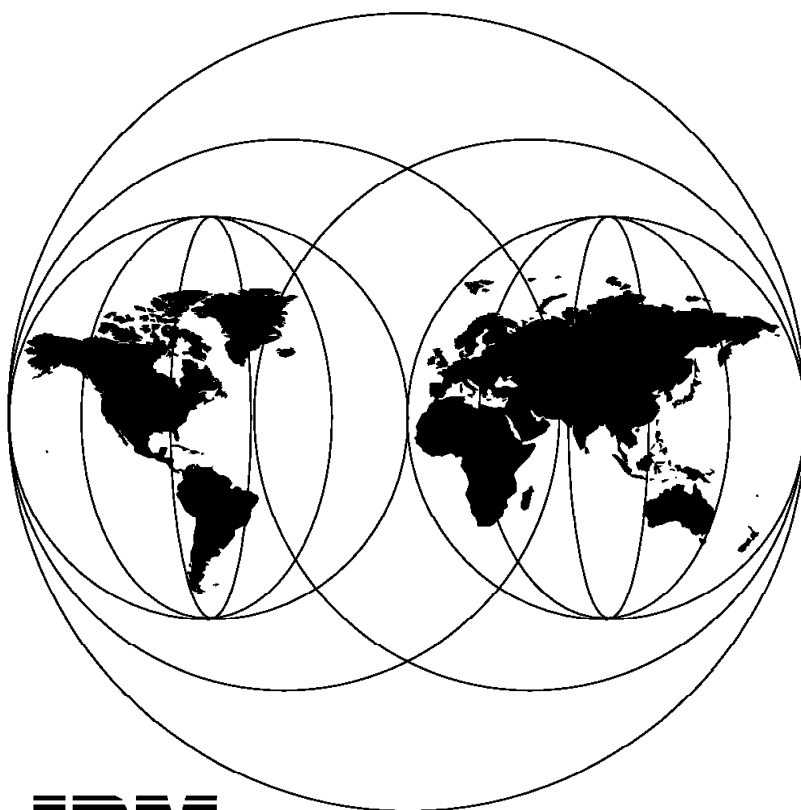


TME 10 Deployment Cookbook: Inventory and Company

October 1997



**International Technical Support Organization
Raleigh Center**



International Technical Support Organization

SG24-2120-00

TME 10 Deployment Cookbook: Inventory and Company

October 1997

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix B, "Special Notices" on page 243.

First Edition (October 1997)

This edition applies to Version 3.1 of TME 10 Inventory, 5697-INV for use with the AIX, Windows NT, and OS/2 operating systems.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 678
P.O. Box 12195
Research Triangle Park, NC 27709-2195

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1997. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	vii
Tables	xiii
Preface	xv
The Team That Wrote This Redbook	xv
Comments Welcome	xvi
Chapter 1. Introduction	1
1.1 TME 10 Deployment Products	1
1.1.1 TME 10 Inventory	2
1.1.2 RDBMS Systems	5
1.1.3 Data Models	5
1.1.4 What Is RIM?	6
1.1.5 Integration of LAN Management Suites	8
1.2 Products We Work With	11
1.3 Where to Find What?	11
1.4 Where to Find Code Examples	12
Chapter 2. Using TME 10 Inventory with Sybase on Windows NT	13
2.1 Overview and Objective	13
2.2 Installing Sybase on Windows NT	15
2.3 Configuring Sybase	21
2.3.1 Changing the System Administrator Password	22
2.3.2 Installing the Sybase Client	24
2.4 Installing TME 10 Inventory in the TMR	28
2.5 Configuring Sybase on Windows NT for Use with TME 10 Inventory	30
2.6 An Example Using Sybase on Windows NT and TME 10 Inventory	37
Chapter 3. Using TME 10 Inventory with Sybase on AIX	47
3.1 Installing Sybase for AIX	48
3.2 Configuring Sybase for AIX	50
3.3 Configuring Sybase for AIX for Use with TME 10 Inventory	53
3.4 Configuring TME 10 Inventory to Work with Sybase for AIX	57
3.5 Database Creation Problem	58
Chapter 4. Using TME 10 Inventory with Oracle for AIX	63
4.1 Overview and Objective	63
4.1.1 Scenario 1: RDBMS Server and RIM Host on the TME Server	64
4.1.2 Scenario 2: RDBMS Server Located Outside the TMR	64
4.1.3 Scenario 3: RIM Host Not on the TME Server	65
4.2 Installing the TME Management Framework (TMF)	66
4.2.1 Installing TME on an AIX Server	67
4.2.2 Installing AIX Managed Nodes as TME Clients	69
4.3 Installing Oracle for AIX	70
4.4 Installing TME 10 Inventory 3.1	73
4.4.1 TME 10 Inventory 3.1 Installation Requirements	74
4.4.2 TME 10 Inventory 3.1 Installation Steps	74
4.5 Configuring Oracle for AIX for Use with TME 10 Inventory 3.1	76
4.5.1 Update Database Tables	76
4.5.2 Erase Database Contents	77

4.6	An Example Using Oracle for AIX and TME 10 Inventory	78
4.7	Moving the RDBMS Server Outside the TMR	81
4.7.1	Removing a Managed Node from the TMR	83
4.7.2	Removing Oracle for AIX from TMR Server	86
4.7.3	Installing Software for the Second Scenario	87
4.7.4	Testing the Scenario	96
4.8	Changing the RIM Host in the TMR	99
Chapter 5. Examples Using Asset Management with TME 10 Inventory		107
5.1	Using TME 10 Inventory with Lotus 1-2-3	107
5.1.1	Installing Lotus 1-2-3 and Lotus Approach	108
5.1.2	Accessing TME 10 Inventory Data from Lotus 1-2-3	109
5.1.3	An Example Using TME 10 Inventory Data with Lotus 1-2-3	117
5.1.4	Summary	120
5.2	Using TME 10 Inventory with License Management and TME 10 Enterprise Console	121
5.2.1	Configure Signature Files	122
5.2.2	Create Asset Database	123
5.2.3	Update Licenses in the License Table	125
5.2.4	Compare Licenses to Actual Installed Programs	128
5.2.5	Create a New Baroc File to Use with TME 10 Enterprise Console	131
5.2.6	Create a New Rule Base	132
5.2.7	Copy Default Rule Base to Licences_Base Rule Base	135
5.2.8	Import Baroc File to Licences_Base Rule Base	136
5.2.9	Compile the Licences_Base Rule Base	138
5.2.10	Make Licences_Base the Active Rule Base	139
5.2.11	Create a New Event Source	140
5.2.12	Create a New Event Group	141
5.2.13	Assign a New Event Filter	143
5.2.14	Assign the New Event Group to an Administrator	144
5.2.15	Monitoring the Events	146
5.2.16	Include the License Checking Script in a TME 10 Task	149
5.2.17	Execute the TME 10 Task	151
5.2.18	Create a TME 10 Job	154
5.2.19	Schedule the TME 10 Job	155
5.2.20	Summary	158
Chapter 6. Monitoring the TME 10 Inventory Database with TME 10 Distributed Monitoring		159
6.1	Overview and Objective	159
6.2	Prerequisites	160
6.3	Installing the TME 10 Distributed Monitoring Sybase Monitoring Collection	160
6.4	Setting Up a TME 10 Distributed Monitoring Profile to Monitor Sybase	161
Chapter 7. Setting Up TME 10 LAN Access for Use with TME 10 Inventory		173
7.1	Overview and Objective	173
7.2	Restrictions When Using TME 10 LAN Access with TME 10 Inventory	175
7.3	Mapping for Discovered Inventory	175
7.4	Installing TME 10 LAN Access	175
7.4.1	Installing TME 10 LAN Access Provider on the NetFinity 5.0 Server	176
7.4.2	Installing TME 10 LAN Access Provider in the TMR	184
7.4.3	Installing TME 10 LAN Access in the TMR	187
7.5	Creating and Configuring a New LAN Access Policy Region	189
7.6	Performing an Inventory Scan	191

7.6.1 Conclusion	198
7.7 Refreshing a TME 10 LAN Access Provider Scan	198
Chapter 8. Using RDBMS Tools to Access TME 10 Inventory Data	199
8.1 Overview of Our Environment	199
8.2 Working with an Oracle Database on an NT Client	200
8.2.1 Installing Oracle for Windows NT 4.0 on an NT Client	200
8.2.2 Configuring the Oracle Client for a Database Connection	204
8.2.3 Testing the Connection to the Oracle Database	209
8.2.4 Different Ways to Look at Oracle Inventory Data	210
8.3 Working with a Sybase Database on a Windows NT Client	224
8.3.1 Working with the ISQL Interface	224
8.3.2 Writing SQL Scripts	226
8.3.3 Working with the WISQL32 Interface	226
8.3.4 Using the Sybase SQL Central	227
8.4 Working with a Sybase Database on an AIX Client	229
8.4.1 Working with the ISQL interface	229
8.4.2 Writing SQL Scripts	229
8.4.3 Using a GUI	230
8.5 Working with an Oracle Database on an AIX Client	230
8.5.1 Using SQL*PLUS	230
8.5.2 Writing SQL Scripts	230
8.5.3 Using a GUI	230
Chapter 9. Summary	231
9.1 Selecting the TME 10 Inventory Server Platform	231
9.2 Selecting the Database Server Platform	231
9.2.1 Can the RDBMS System Be Shared?	234
9.2.2 RDBMS Scalability	234
9.3 Integrating Existing Inventory Solutions	238
9.3.1 Integrating Inventory Information from NetView Distribution Manager	238
9.4 Signature Scanning vs Custom Scanners	238
9.5 TME 10 Inventory and TME 10 Software Distribution	239
Appendix A. Oracle Network Manager	241
Appendix B. Special Notices	243
Appendix C. Related Publications	245
C.1 International Technical Support Organization Publications	245
C.2 Redbooks on CD-ROMs	245
C.3 Other Publications	245
How to Get ITSO Redbooks	247
How IBM Employees Can Get ITSO Redbooks	247
How Customers Can Get ITSO Redbooks	248
IBM Redbook Order Form	249
Glossary	251
List of Abbreviations	253
Index	255

ITSO Redbook Evaluation 257

Figures

1.	TME 10 Full Life-Cycle Management	2
2.	Inventory Management	3
3.	Structure of TME 10 Inventory	4
4.	RIM Architecture	7
5.	TME 10 LAN Access Overview	9
6.	Multiple Services Broker (MSB) Object	10
7.	TME 10 Inventory in a Sybase Environment on Windows NT	14
8.	Windows NT Product Selection Screen Window	16
9.	Custom Selection Window	17
10.	Install Additional Components Window	17
11.	SQL.INI Dialog Window	18
12.	SQL Server Configuration Window	19
13.	Creating Master Device File	20
14.	Sybase SQL Server Professional Tools Window	20
15.	SYBASE - Services Manager Window	21
16.	SQL Central Window	22
17.	Connect to Sybase SQL Server Window	22
18.	SQL Central Window	23
19.	Change Password for sa User	23
20.	Change Password for sa Window	24
21.	SQLEDIT Window	25
22.	SQL.INI File on victor	26
23.	Windows NT Product Selection Screen Window	27
24.	SQL.INI Dialog Window	28
25.	Install Options Window	29
26.	WISQL32 Window	31
27.	Open Connection from foxtrot to victor	31
28.	Open Connection Window	32
29.	WISQL32 Window	32
30.	Open Scripts Needed to Be Run	33
31.	WISQL32 Window	34
32.	Check Tivoli User ID	35
33.	Open Connection to Inventory Using Tivoli User ID	36
34.	WISQL32 Window	37
35.	Profile Manager Window	38
36.	Create Profile Window	39
37.	Profile Manager Window	40
38.	Inventory Profile Window - Initial Scan Settings	41
39.	Customization Window (Part 1)	42
40.	Customization Window (Part 2)	43
41.	Profile Manager	44
42.	Hardware Inventory for wtr05123	45
43.	Using TME 10 Inventory with Sybase on AIX	47
44.	Sybase Installation Program	49
45.	Installation Status Window	50
46.	Quick Configuration Window	51
47.	Sybinit Status Window	52
48.	/usr/local/sybase/interfaces File	52
49.	SQLEDIT Window	53
50.	Open Connection Window	54
51.	Open Scripts Window	55

52.	Open Connection to Inventory Using Tivoli User ID	56
53.	WISQL32 Window Connected to SYBASE	58
54.	SQL Central Window	59
55.	SQL Central Window	59
56.	Database Devices Window	60
57.	Master Properties Window	60
58.	Master Properties Window	61
59.	Databases - SQL Central Window	62
60.	Scenario 1 - RDBMS Server and RIM Host on the TME Server	64
61.	Scenario 2 - RDBMS Server Located Outside the TMR	65
62.	Scenario 3 - RIM Host Not on the TME Server	66
63.	Shell Script db_setup.sh	71
64.	TMR Server, RDBMS Server, and RIM Host on Same Workstation	79
65.	RDBMS Server Outside the TMR	82
66.	Policy Region aixregion	83
67.	Profile Manager:inventory	84
68.	Deleting a Managed Node from the TMR	85
69.	Confirm the Deletion of a Managed Node from the TMR	85
70.	Oracle Installer: Confirm You Have Already Run the Preinstallation Script	88
71.	Oracle Installer: Relink All Executables?	88
72.	Oracle Installer: SQL*Net Listener	89
73.	Oracle Network Manager: Main Window and Menu	91
74.	Oracle Network Manager: Community Parameters	92
75.	Oracle Network Manager: Node Parameters	93
76.	Oracle Network Manager: SQL*Net Listener Parameters	94
77.	Oracle Network Manager: Database Parameters	95
78.	RIM Host on a Managed Node	100
79.	Verify SQL*Net Connection between RS600026 and RS600024	102
80.	Using Lotus 1-2-3 with TME 10 Inventory	107
81.	Lotus SmartSuite 97 Window	108
82.	Select Lotus SmartSuite Applications	109
83.	Lotus 1-2-3 Worksheet	110
84.	Query Table Assistant Window	110
85.	Query Table Positioning Pop-Up	111
86.	Open Window	111
87.	Logon to System 10 Window	112
88.	Viewing TME 10 Inventory Tables in Lotus 1-2-3	113
89.	Selecting Columns from INVENTORYDATA View to Bring to Lotus 1-2-3	114
90.	Query Table in a Lotus 1-2-3 Worksheet	115
91.	Options when Copying Query Table to a Lotus 1-2-3 Worksheet	116
92.	Having Copied Data from Query Table to Lotus 1-2-3 Worksheet	116
93.	Selecting Columns from the PROCESSOR Table to Bring to Lotus 1-2-3	117
94.	More Data from Query Table to Lotus 1-2-3 Worksheet	118
95.	Using Lotus 1-2-3 Function to Assign Asset Value	119
96.	TME 10 Inventory Data Shown Graphically in Lotus 1-2-3 Worksheet	120
97.	Using TME 10 Enterprise Console with TME 10 Inventory for Checking Licences	121
98.	Adding Signature File to Inventory Profile	122
99.	Script to Create SW_LICENSES (Part 1 of 2)	123
100.	Script to Create SW_LICENSES (Part 2 of 2)	124
101.	Contents of Table SW_LICENSES	125
102.	Script to Update SW_LICENSES (Part 1 of 2)	125
103.	Script to Update SW_LICENSES (Part 2 of 2)	126
104.	Running the Script swlic_upd.sh	127

105.	Contents on Table SW_LICENSES, After Update via swlic_upd.sh	128
106.	Script to Check Licenses and Send T/EC Event (Part 1 of 2)	129
107.	Script to Check Licenses and Send T/EC Event (Part 2 of 2)	130
108.	Contents on Table SW_LICENSES, After Update by swlic_cmp.sh	131
109.	BAROC File to Describe License Events	132
110.	Accessing Rule Base Definitions	133
111.	Creating a New Rule Base	133
112.	Rule Base Properties	134
113.	Copy Default Rule Base to Licences_Base Rule Base	135
114.	Copy Rule Base	135
115.	Select Import to Rule Base	136
116.	Import Into Rule Base	137
117.	Select Compile Rule Base	138
118.	View Compile Results	138
119.	Load Rule Base	139
120.	Load Rule Base Options	139
121.	Licences_Base Rule Base Is Active	140
122.	Add T/EC Source	141
123.	Event Group Management	142
124.	Add the Event Group	142
125.	Create Filter for the Event Group	143
126.	Save the Event Group	144
127.	Calling the Assign Event Groups Window	145
128.	Assign Event Groups Window	145
129.	Event Groups and Sources	146
130.	Event Group with Indicator	147
131.	License Event Group Monitor	147
132.	Event Detail	148
133.	Create a TME 10 Task	149
134.	TME 10 Task Properties	150
135.	TME 10 Task Script Location	151
136.	Execute the TME 10 Task	151
137.	TME 10 Task Execute Options	152
138.	TME 10 Task Output	153
139.	Create a TME 10 Job	154
140.	TME 10 Job Properties	155
141.	Schedule the TME 10 Job	156
142.	Scheduling Options	157
143.	Environment for the Installation of TME 10 Distributed Monitoring, Sybase, and TME 10 Inventory	159
144.	TME 10 Distributed Monitoring Talking to Sybase	160
145.	Installing Sybase Monitor Collection	161
146.	Create Profile Manager	162
147.	Give the Profile Manager a Name	162
148.	Profile Manager with Subscriber	163
149.	Create Profile	164
150.	Profile Manager with Profile and Subscriber	165
151.	TME 10 Distributed Monitoring Profile Properties	166
152.	Add Monitor to TME 10 Distributed Monitoring Profile	167
153.	Edit Monitor	168
154.	Set Monitoring Schedule	169
155.	Save and Close Profile	170
156.	Pop-up from the Monitor	171
157.	TME 10 LAN Access Environment	173
158.	TME 10 LAN Access Install Window	176

159.	TME 10 LAN Access Install Window	177
160.	TME 10 LAN Access Install - Installing... Window	177
161.	NetFinity Provider Configuration Window	178
162.	TME 10 LAN Access Install Complete! Window	179
163.	OS/2 Desktop Window	179
164.	Hardware Inventory Data	181
165.	Software Inventory Data	182
166.	Incoming Passwords - KWIELA Window	183
167.	Login Window	184
168.	Setup Window	185
169.	LAN Access Support Program - Authentication Window	186
170.	Network Driver Configuration Window	187
171.	Install Product Window	188
172.	Product Install Window	189
173.	Set Managed Resources Window	190
174.	Create Policy Region Window	190
175.	Set Resource Roles Window	191
176.	LANAccessSite Window	192
177.	LANAccessSite Window	193
178.	Policy Region: Lan Access Region Window	193
179.	LANAccessCollection Window	194
180.	LANAccessColl... Window	195
181.	OS/2 Machines Window	196
182.	Software Inventory Window	197
183.	Hardware Inventory Window	197
184.	RDBMS Tools Environment	199
185.	Oracle Installer Window	200
186.	Language Window	201
187.	Oracle Installation Settings Window	201
188.	Oracle7 Server Window	202
189.	Oracle7 Client Window	202
190.	Oracle Documentation Window	203
191.	Oracle Window	203
192.	Installation Completed Window	204
193.	SQL*Net Easy Configuration Window	205
194.	Choose Database Alias Window	205
195.	Choose Protocol Window	206
196.	Choose TCP/IP Host Name and Database Instance Window	206
197.	Confirm Adding Database Alias Window	207
198.	Content of the TNSNAMES.ora File	208
199.	Logon to Oracle Window	209
200.	Oracle ODBC 32Bit Test - Query1 Window	210
201.	C:ORANTPSPLUS33.exe Window	211
202.	C:ORANTPSPLUS33.exe Window	212
203.	Sample SQL Spool File (Test.log)	213
204.	C:ORANTPSPLUS33.exe Window	214
205.	Sample SQL Script File (Test1.sql)	215
206.	Sample Output List (test1.lst)	216
207.	Sample SQL Script File (Test2.sql)	217
208.	Sample Output List (Test2.lst)	217
209.	Inventory [Test2.sql] - Oracle Worksheet Window	219
210.	Command History Window	220
211.	Login Information Window	221
212.	Oracle Scheme Manager Window	221
213.	Inventory - Oracle Schema Manager Window	222

214.	Inventory - Oracle Schema Manager Window	223
215.	Inventory - Oracle Schema Manager Window	224
216.	Database Connection Established Successfully	225
217.	Select Statement for INVENTORYDATA	225
218.	Output from Select Statement for INVENTORYDATA	225
219.	WISQL32 Window After Successful Connection	226
220.	Start/Programs/Sybase SQL Server Professional\SQL Central Window	227
221.	SQL Central Window	227
222.	Connect to Sybase SQL Server Window	228
223.	SQL Central Window - Inventory Database Tables	228
224.	SQL Central Window - COMPUTER_SYSTEM Details	229
225.	Sample of a Korn Shell Script File	230
226.	RIM Accessing an RDBMS System	232
227.	DB2 for AIX Using a DDCS Connection	235
228.	TME 10 Inventory Scalability with Multiple TMR Servers	236
229.	TME 10 Inventory with Distributed RDBMS Servers	237
230.	tnsname.ora	241
231.	sqlnet.ora	241

Tables

1. wserver Flags	67
2. wpatch Flags	68
3. Oracle Installer Commands	71
4. wcrtrim Flags for Creating the RIM Host Object	104
5. SQL Table SW_LICENSES	124

Preface

This redbook provides guidance on how to understand and use TME 10 Inventory and related applications.

It will help you position TME 10 Inventory as a solution for enterprise inventory and asset management and show practical examples for using it.

We demonstrate the major features in TME 10 Inventory and TME 10 LAN Access as well as related applications, such as TME 10 Enterprise Console and TME 10 Distributed Monitoring.

Also, we show how to implement TME 10 Inventory using different RDBMS systems, namely Oracle and Sybase in several client/server combinations.

The scenarios are documented so that service providers can use the code and examples as a base in client implementations.

This book continues a series of redbooks in the area of TME 10 deployment management that was started with the redbook *The TME 10 Deployment Cookbook: Courier and Friends*, SG24-4976.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the Systems Management and Networking ITSO Center, Raleigh.

Stefan Uelpenich is an Advisory ITSO Representative, working as a project leader at the Systems Management and Networking ITSO Center, Raleigh. He applies his extensive field experience as an I/T architect and project leader into his work at the ITSO, where he writes extensively and teaches IBM classes worldwide on all areas of systems management. Before joining the ITSO, he worked in IBM Germany's Professional Services organization as an Advisory I/T Architect for Systems Management, consulting major IBM customers.

Fred Plassman is a Senior International Technical Support Organization (ITSO) Specialist at the Systems Management and Networking ITSO Raleigh Center. He writes extensively and teaches IBM classes worldwide on all areas of software distribution. Before joining the ITSO in July 1996, Fred provided national technical support for IBM software distribution products, primarily IBM Software Distribution V3.1 for AIX and NetView DM/6000 V1.2 in the IBM Networking Support Center, Gaithersburg, Maryland, USA.

Simone Crucchi is an I/T Specialist working in IBM South EMEA Product Support Services in Italy in System and Network Management Services. He has worked at IBM for 2 years and his areas of expertise include AIX and Windows NT system administration. He is currently providing Tivoli services to Italian customers.

Gordon Love is an Advisory I/T Specialist working in Network Station Management at IOS/IBM South Africa. He has three years of experience in systems management and is currently working as a technical team leader for software distribution. Before joining IBM he worked in product support for software distribution products at a large South African bank. Gordon has also

coauthored the redbook *The TME 10 Deployment Cookbook: Courier and Friends*, SG24-4976.

Wolfgang Milchram is an Advisory Systems Engineer in the Network Support and Service Center in IBM Austria. He is the TME 10 Deployment Specialist for IBM Austria. Wolfgang has 10 years of experience in UNIX environments and has also coauthored the redbook *TME 10 Cookbook for AIX*, SG24-4867.

Ingemar Nilsson is a Systems Management Consultant, working in the IBM Nordic PSS Systems Management team, located in Sweden. The team provides technical support for Tivoli products for the IBM Nordics region. He has worked with IBM for 29 years. Before his current position he provided technical marketing support for the AIX SystemView family of products at the IBM Nordic Lab in Sweden.

Juan Ramses Herrera Soto is an I/T Specialist working at Instituto Tecnológico y de Estudios Superiores de Monterrey. He holds an Electronics and Communications degree in Engineering and is currently finishing his Master's degree in Computer Science. He is working as a systems administrator and Internet services administrator. His areas of expertise include UNIX, Windows NT and networking.

Klaus Wieland is a Senior Systems Engineer, working as a project leader in IBM Germany's Professional Services Organization. He has been with IBM for more than 25 years. Klaus has been the project leader for several large Systems Management projects at IBM customers and is currently involved in a major Tivoli project in Germany.

Thanks to the following people for their invaluable contributions to this project:

Barry Nusbaum
Linda Robinson
Dave Shogren
Shawn Walsh
Gail Doucette Wojton
International Technical Support Organization, Raleigh Center

Raghuram Belur
Adi Mishra
Chip Young
Tivoli Systems Inc., An IBM Company, Austin

Wolfgang Geiger
ITSO Raleigh, at Tivoli Systems Inc., An IBM Company, Austin.

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 257 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

This soft copy for use by IBM employees only.

For Internet users <http://www.redbooks.ibm.com>
For IBM Intranet users <http://w3.itso.ibm.com>

- Send us a note at the following address:
 redbook@vnet.ibm.com

Chapter 1. Introduction

In this redbook, we work with TME 10 Deployment Management products focusing on TME 10 Inventory and related applications.

We show the new functions of these products and show examples of how TME 10 Inventory can be integrated with other TME 10 applications as well as with applications from other vendors.

This book is not meant as a basic tutorial on TME 10 Inventory but rather tries to give advice on how to use TME 10 Inventory in a number of different configurations and scenarios.

In this redbook, we assume that you are familiar with the basic principles of TME 10 and its applications as well as with the basic principles of deployment management.

If you are not familiar with TME 10 Deployment Management, we recommend that you read the redbooks *TME 10 Cookbook for AIX - Systems Management and Networking Applications*, SG24-4867 and *The TME 10 Deployment Cookbook - Courier and Friends*, SG24-4976 first. If you are not familiar with TME 10 at all yet, we recommend that you read the redbook *Understanding Tivoli's TME 3.0 and TME 10*, SG24-4948.

In this chapter, we give a brief introduction to the products involved in inventory management with TME 10 Inventory and describe the high-level function of these products or product components.

Note

This redbook is released in concert with another redbook called *TME 10 Inventory 3.2 New Features and Database Support*, SG24-2135. In order to fully benefit from the introduction and especially the summary chapter in this redbook and to see what's new in TME 10 Inventory 3.2 you should obtain a copy of that redbook, also.

1.1 TME 10 Deployment Products

TME 10 Deployment Management is primarily made up of two main components, TME 10 Software Distribution and TME 10 Inventory.

While speaking of deployment will make you immediately think of software distribution, including TME 10 Inventory in the deployment management area is based on the idea of a full life-cycle management of components.

These life-cycle management components can be anything that is relevant for operating your business, for example, hardware and software components.

The following figure gives an overview of TME 10's full life-cycle management:

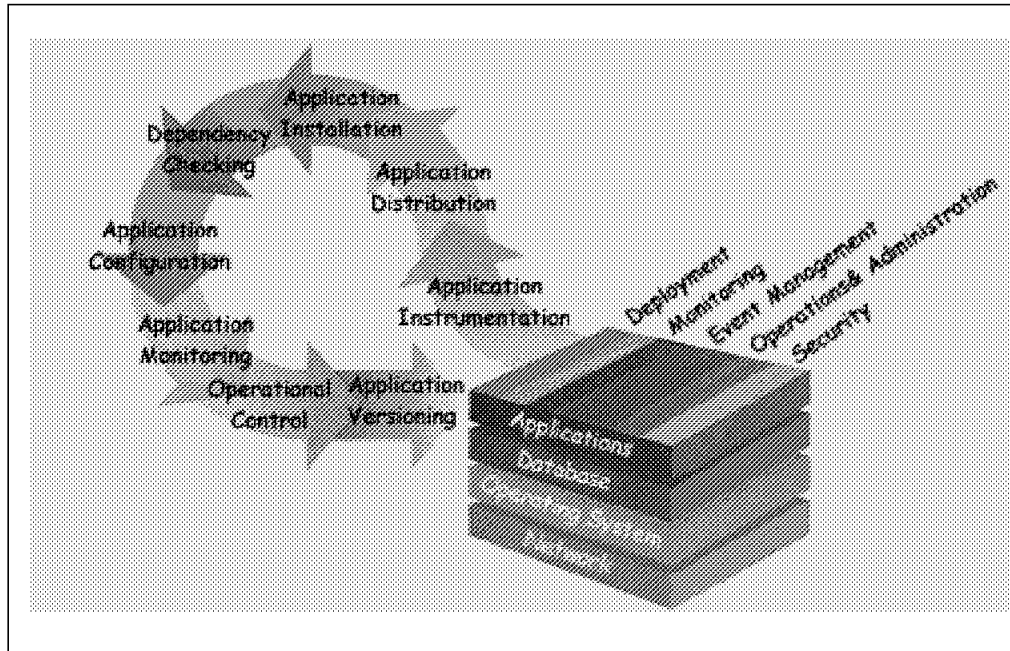


Figure 1. TME 10 Full Life-Cycle Management

For example, the management cycle for a software package could start with developing the application using Tivoli's Application Management Specification (AMS) which will enable it to be managed by TME 10 applications. The AMS-enabled or any other application can then be distributed using TME 10 Software Distribution, later being updated with a newer version, again using TME 10 Software Distribution and eventually be retired (deinstalling it using TME 10 Software Distribution).

During the entire process, TME 10 Inventory can be used to control or assist in this process. The software product that is to be deployed can be included in a reference configuration (stored in TME 10 Inventory) thus planning the deployment. Before the distribution, the current configuration of the endpoint (also stored in TME 10 Inventory) is checked to see if it meets the prerequisites required by the software. The software configuration of all endpoints and the installation status can also be stored in TME 10 Inventory.

TME 10 Software Distribution is covered in detail in the redbook *The TME Deployment Cookbook - Courier and Friends*, SG24-4976. An introduction to the use of the TME 10 Inventory application can be found in *TME 10 Cookbook for AIX - Systems Management and Networking Applications*, SG24-4867.

1.1.1 TME 10 Inventory

TME 10 Inventory is the application in TME 10 that can be used primarily to gather inventory information, such as hardware and software inventory information about the components in your enterprise.

This is achieved through detector components, also called *scanners*, in or outside of the TME 10 Inventory application that collect data about a component and then hand it over to TME 10 Inventory for storing in a database.

One example for an outside detector component is TME 10 LAN Access, as described in 1.1.5, "Integration of LAN Management Suites" on page 8.

The plain data itself, however, does not provide a major benefit. The real advantage of an inventory solution comes when using the data in the inventory database, as shown in the following figure:

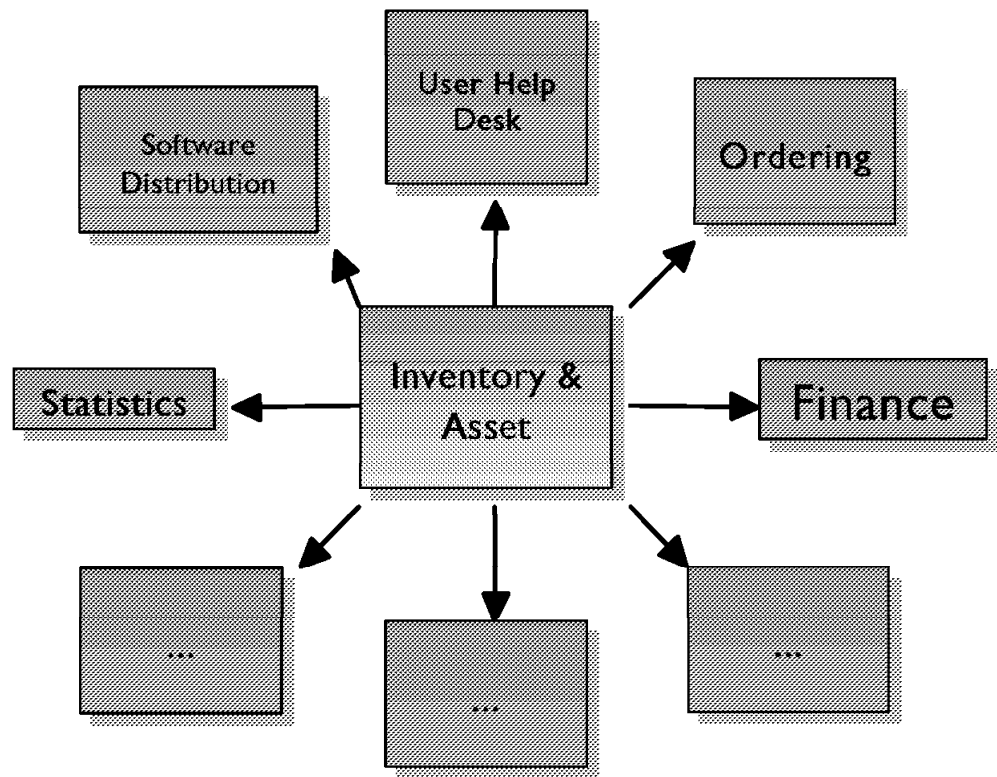


Figure 2. Inventory Management

The above figure shows some possible connections of inventory management. One example for the use of inventory management is the connection with software distribution. For example, before a software distribution to an endpoint takes place, the inventory database can be checked to see if the endpoint meets certain prerequisites that are defined to be necessary for the file package to be distributed.

If the criteria are met, the file package is then distributed and the status change of the endpoint and the file package can be reflected in the inventory database. For example, the inventory database can contain which software packages are installed on the endpoint. Or, for the file package, the inventory database could contain to which endpoints the software has been distributed, in which version and at which time.

For a user help-desk, inventory information can be very useful in solving end-user problems. For example, when the user help-desk receives a problem from a PC user, the person receiving the problem can retrieve information from the inventory database about the PC in question and have a look at the PC's configuration before proceeding further. An enhancement of this process could be a rule engine that correlates certain known problems with stored configurations in the inventory database and therefore automatically handle certain problems.

Besides mainly technical issues, there are other valuable uses for inventory information, for example, in connection with ordering or accounting processes.

For example, a connection to an ordering system may be used to automatically order new licenses of a software product, when the database reports that the number of valid licenses has been exceeded.

Inventory information correlated with financial information can be used, for example, to calculate the book value of equipment or to calculate what percentage of the equipment is written off already.

Having examined a few uses for an inventory management application, what is the advantage of TME 10 Inventory?

First, let's have a look at the structure of TME 10 Inventory:

"Sharing objects"

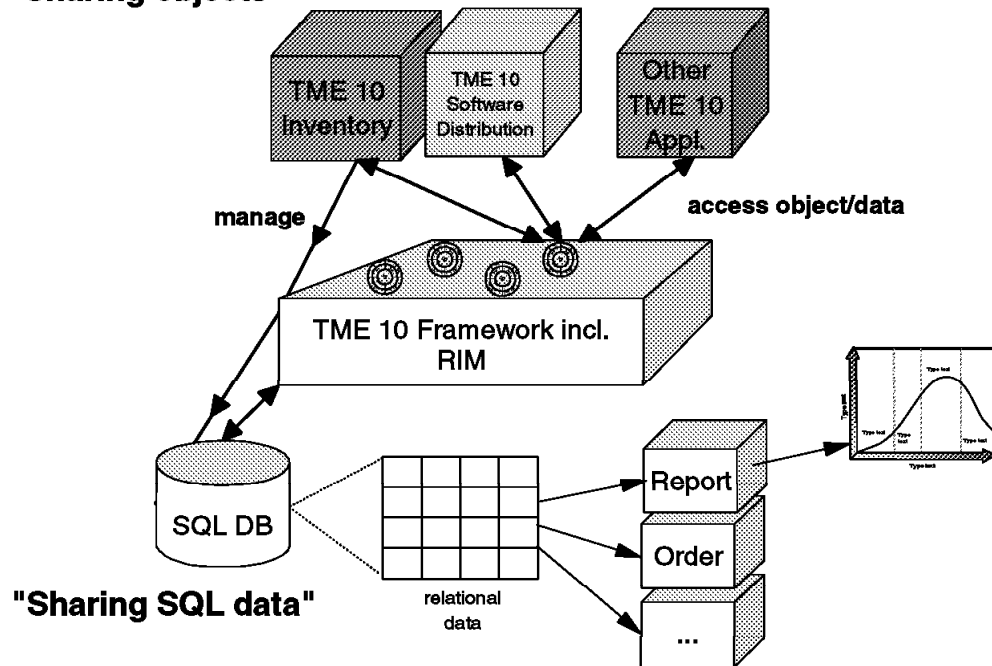


Figure 3. Structure of TME 10 Inventory

TME 10 Inventory has two major advantages:

- TME 10 Inventory is based on the TME 10 Framework and can therefore tightly and automatically be integrated with other TME 10 applications.
- TME 10 Inventory stores inventory information in a standard relational database system and therefore allows many non-TME 10 applications to share this data.

TME 10 Inventory shields access to the data by encapsulating the access in CORBA objects. Therefore, another TME 10 application, for example TME 10 Software Distribution can share that data by just accessing the information in these objects.

While this is a good method for sharing data between TME 10 applications, other applications, that might not be TME 10 applications, may want to share that data as well. Since these applications are most likely not written to run on the TME 10 Framework, they cannot share the data by accessing TME 10 Framework objects.

However, most applications that want to share that data, for example, financial applications, are likely to access data stored in a standard relational (SQL) database, or, store their own data in SQL databases as well.

This is important, because normally you will want to combine the data gathered by TME 10 Inventory with other data, for example, a field containing the current book value of the PC type stored in the inventory database.

1.1.2 RDBMS Systems

As mentioned before, TME 10 Inventory can only run in connection with a Relational DataBase Management System (RDBMS) that stores data on behalf of TME 10 Inventory.

While this approach requires you to buy an RDBMS license in addition to TME 10 Inventory it has major advantages:

Some inventory products store information in a proprietary way, which will not allow effective sharing of information and usually also is not as effective as an RDBMS system (that is specialized on managing large amounts of data).

TME 10 Inventory lets you pick the RDBMS system of your choice, allowing any application that can access SQL data to share it and using the advanced features of the RDBMS system, such as scalability and performance to manage the data.

Thus, TME 10 Inventory can share data with TME 10 applications (Sharing Objects) as well as any other application (Sharing SQL data).

But what kinds of RDBMS systems does TME 10 Inventory support? This is determined by the RIM component, as described in 1.1.4, "What Is RIM?" on page 6.

1.1.3 Data Models

Once the detector components (usually called scanners in TME 10) have gathered information about a particular system component, this information needs to be stored in the TME 10 Inventory database.

As TME 10 Inventory (through the RIM component described in 1.1.4, "What Is RIM?" on page 6) stores the collected information in a relational database system, the RDBMS system needs to be prepared to store that information; that is, it needs to have database tables defined that can hold the gathered information.

The data model determines the structure of these tables. The TME 10 Inventory data model, and therefore the predefined database tables that come with TME 10 Inventory, are used by TME 10 Inventory, as well as, other TME 10 applications.

The data model that comes with TME 10 Inventory 3.1 is used by TME 10 Inventory, as well as, TME 10 Software Distribution (Courier) and Tivoli Manager for Applications (TMA). The data model contains entities that are not actually populated or updated by TME 10 Inventory, although they might be considered inventory information.

The reasons for not actually updating the data model entities are that, on the one hand, the currently used scanners are not able to provide (that is scan) the

information needed to fill the corresponding tables, and on the other hand, there are entities that have been included for future support.

The data model provided with TME 10 Inventory is supposed to be used by TME 10 Inventory itself and some other TME 10 applications. Other applications, for example, business applications, or even other TME 10 or systems management applications, will use other data models.

There have been major efforts in the past to unify data models in the area of systems management, with the eventual goal of having all systems management applications share common data, enabling them to truly cooperate.

While discussion of this topic is beyond the scope of this book it should be noted that the TME 10 Inventory data model alone does not constitute a comprehensive solution to that problem. You should also notice that you have limited ability to extend the data model that comes with TME 10 Inventory, for example, by adding new MIF groups.

1.1.4 What Is RIM?

Previous versions of TME 10 Inventory supported two Relational DataBase Management Systems (RDBMS), namely Sybase and Oracle.

To support these two RDBMS systems, there had to be interface code to support the different RDBMS server systems. In previous versions of TME 10 Inventory, this support was integrated in the TME 10 Inventory product.

To have a clear interface to the RDBMS system and to easily support other RDBMS systems, starting with TME 10 Inventory 3.1, RDBMS specific access code was removed from TME 10 Inventory and encapsulated in a module, called RDBMS Interface Module (RIM).

The principle of RIM is shown below:

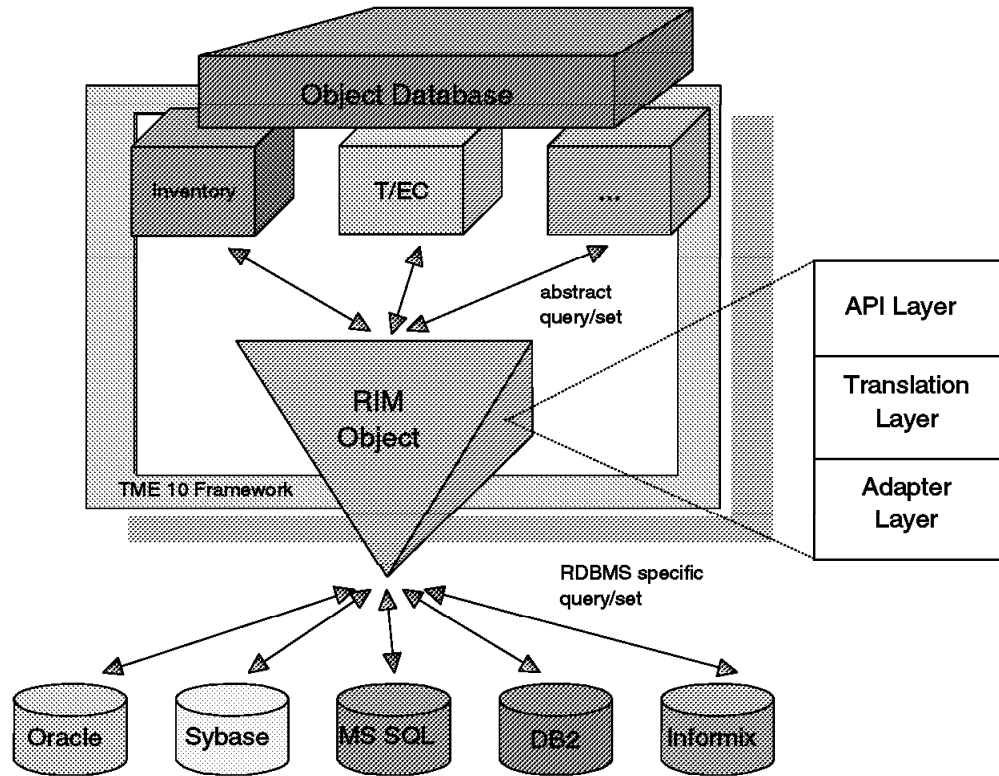


Figure 4. RIM Architecture

Note

You should notice that DB2 and MS SQL are only supported in TME 10 Framework 3.2 and TME 10 Inventory 3.2 or later. Informix, although shown, is not currently supported.

Refer to *TME 10 Inventory 3.2 New Features and Database Support*, SG24-2135 for a detailed explanation of how to use TME 10 Inventory with DB2 and MS SQL.

The idea behind RIM, is that the TME 10 Inventory application always talks to RIM, but not directly to the RDBMS system, and all code for accessing database-specific information is hidden in RIM. Therefore, RIM is used to provide the RDBMS access for TME 10 Inventory.

This architecture has the major advantage that adding support for another RDBMS system does not require changes in TME 10 Inventory but only in RIM.

While with TME 10 Framework 3.0 and TME 10 Inventory 3.0, RIM was part of TME 10 Inventory, starting with TME 10 Framework 3.1 RIM is part of the TME 10 Framework, thus also allowing other TME 10 applications to share the RDBMS support. For example, another TME 10 application that makes use of RIM is TME 10 Enterprise Console 3.1.

The RIM module is divided into three layers. The API layer provides an interface to the TME 10 application that wants to use RIM services. The API is supplied as a set of C library calls.

The translation layer performs the tasks of looking up the RDBMS server to connect to and of translating the data type used in the object representation to a standard SQL string that is used to talk to the RDBMS system.

The adapter layer provides any operations that are specific to a certain RDBMS product and therefore masks the differences of the RDBMS systems to the other layers. Therefore, support for another RDBMS system requires enhancements to this layer.

1.1.5 Integration of LAN Management Suites

While TME 10 Inventory acts as the enterprise-wide inventory and configuration application, the task of gathering inventory data is not restricted to TME 10 Inventory itself.

In particular, one might already be using a LAN or workgroup management tool, such as Microsoft Systems Management Server (SMS), Intel LANDesk or IBM NetFinity. While these tools cannot provide an enterprise-wide solution for systems management or inventory, they can provide, for example, the data collection mechanisms and other functions within a workgroup.

1.1.5.1 TME 10 LAN Access

TME 10 LAN Access is a product to integrate these LAN management suites with TME 10 Inventory and other TME 10 applications. TME 10 LAN Access transports data between the LAN management suite and TME 10. In the case of TME 10 Inventory, for example, TME 10 LAN Access retrieves the software and hardware inventory data that has been gathered by the LAN tool and makes the data available to TME 10 Inventory.

TME 10 LAN Access allows you to continue using the LAN management suite while at the same time providing a central inventory database.

To summarize, TME 10 LAN Access has the following goals:

- Enable TME 10 to manage PC desktop management suites
- Leverage investments of TME 10 customers in these PC desktop management suites
- Allow TME 10 administrators to have a homogeneous view of heterogeneous LANs

This cooperative approach to management leverages the PC desktop suite's strength in managing small to medium workgroups and uses TME 10 LAN Access to integrate these workgroups into enterprise management provided by TME 10.

In the initial release, TME 10 LAN Access supports the following LAN management tools:

- Microsoft SMS
- Intel LANDesk
- IBM NetFinity

However, if you use a LAN management product that is not supported, TME 10 LAN Access also provides an API that can be used to tailor TME 10 LAN Access to other LAN management tools.

The following figure gives an overview of how TME 10 LAN Access is used.

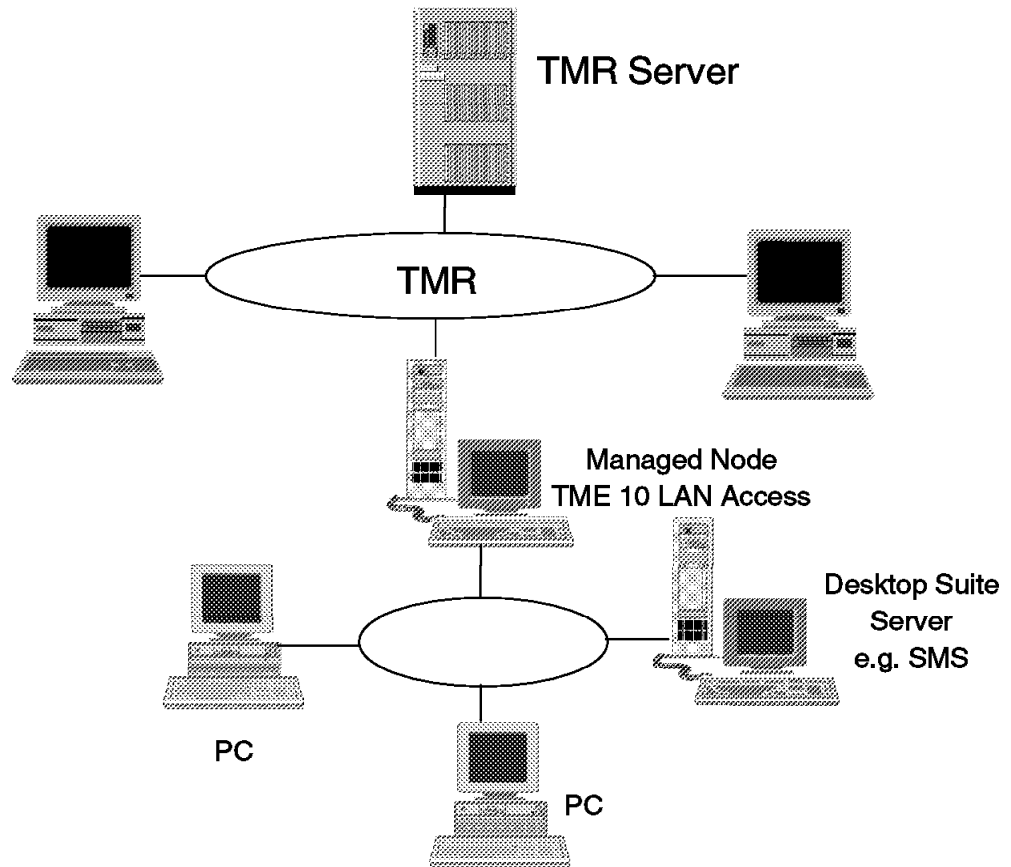


Figure 5. TME 10 LAN Access Overview

In the above figure, a desktop management suite, in this case SMS, manages a number of PC desktops in its management domain. The desktop suite server communicates with a system in the domain that has TME 10 LAN Access installed.

At the same time, the TME 10 LAN Access system is a managed node in the TMR where the desktop suite is integrated.

The integration is provided by a Multiple Services Broker (MSB) object that is managed in the TMR. The following figure shows the structure of such an MSB object:

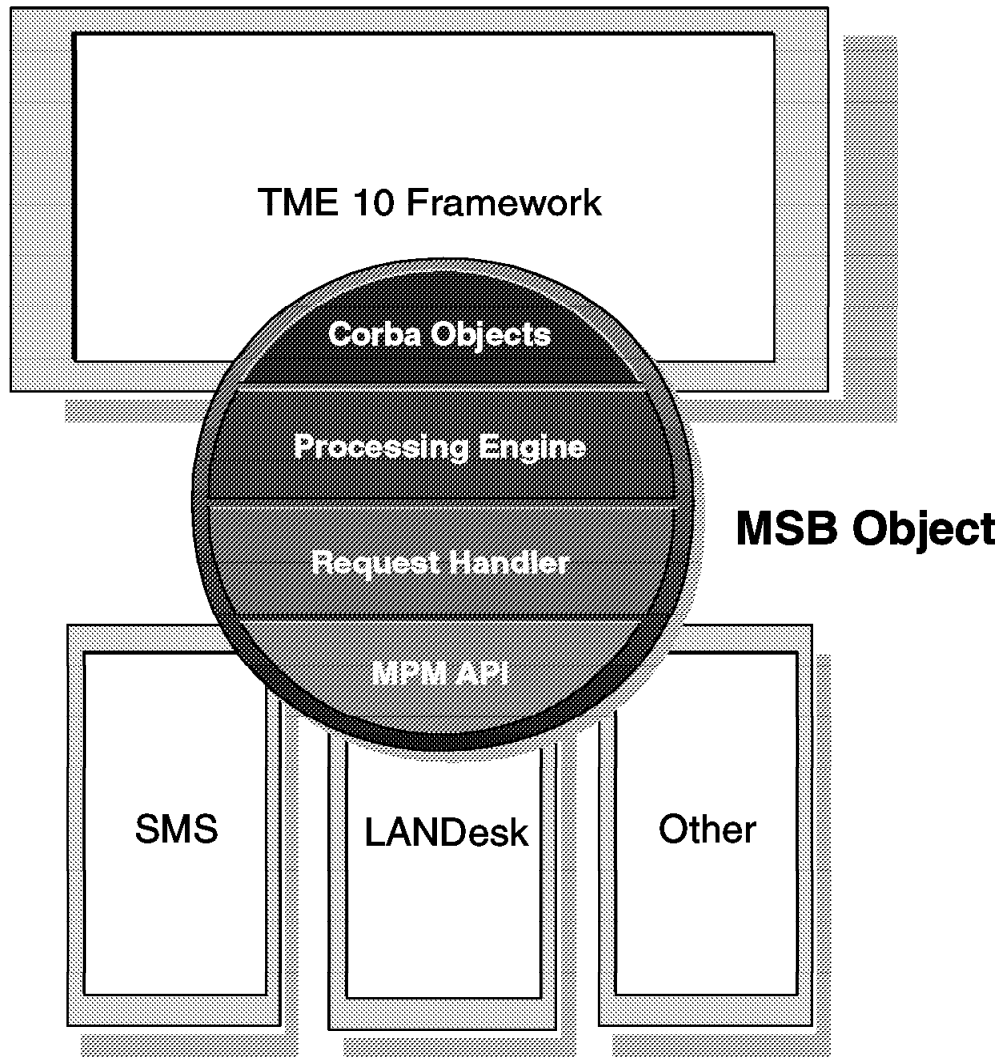


Figure 6. Multiple Services Broker (MSB) Object

The MSB object is used at the managed node that has TME 10 LAN Access installed. On the lowest level, the MSB object utilizes the Multi-Platform Management API (MPM) to shield the specifics of the LAN management suite. On the highest level, the MSB object provides the interface to make the information from the LAN management suite available to the TME 10 Framework as CORBA objects.

The Processing Engine/Request Handler gathers data from and dispatches work to the desktop management suite.

Support for additional desktop management suites can be added by utilizing the MPM API.

1.1.5.2 Multi-Platform Management API (MPM)

Tivoli's Multi-Platform Management API (MPM) provides a set of common procedures that management applications can use to exchange data, initiate management operations and handle alerts in cooperation with desktop management suites.

The following categories of operations are supported by MPM API:

- Discovery of devices and device capabilities
- Inventory of device hardware and software
- Alert management
- Remote file system operations
- Remote program execution

Apparently for us the inventory category is most relevant.

We will not explain MPM API in further detail here; however, it should be noted that TME 10 LAN Access is based on MPM API.

1.2 Products We Work With

In this redbook we work with the following products:

- TME 10 Inventory 3.1 on the following platforms:
 - Windows NT 4.0 (server, client), Windows 95 (client)
 - AIX 4.1.4 (server)
- TME 10 LAN Access 1.0 on the following platforms:
 - Windows NT 4.0 (TME 10 LAN Access node)
 - OS/2 (TME 10 LAN Access Provider)
- Oracle 7.1.6.2.0 on the following platform:
 - AIX 4.1.4
- Sybase 11 on the following platforms:
 - Windows NT 4.0 (server and client)
 - AIX 4.1.4 (server)
- NetFinity 5.0 on the following platform:
 - OS/2 (server and client)
- Lotus 1-2-3 and Lotus Approach on the following platform:
 - Windows NT 4.0

1.3 Where to Find What?

As mentioned before, the redbook *TME 10 Inventory 3.2 New Features and Database Support*, SG24-2135 complements this redbook and has additional scenarios besides the ones shown here.

In this redbook we focus on the following topics:

The first three chapters deal with how to work with TME 10 Inventory in combination with the Oracle and Sybase RDBMS systems in different configurations. This is described in detail in Chapter 2, “Using TME 10 Inventory with Sybase on Windows NT” on page 13, Chapter 3, “Using TME 10 Inventory with Sybase on AIX” on page 47 and Chapter 4, “Using TME 10 Inventory with Oracle for AIX” on page 63.

Chapter 5, “Examples Using Asset Management with TME 10 Inventory” on page 107 shows two examples of how to actually use inventory data gathered and stored by TME 10 Inventory. This is shown in combination with Lotus 1-2-3 and other TME 10 components, such as TME 10 Enterprise Console.

Then we look at another TME 10 application, TME 10 Distributed Monitoring, and show how it can be used to monitor the RDBMS that TME 10 Inventory uses. This is shown in Chapter 6, “Monitoring the TME 10 Inventory Database with TME 10 Distributed Monitoring” on page 159.

In Chapter 7, “Setting Up TME 10 LAN Access for Use with TME 10 Inventory” on page 173 we show how the new TME 10 LAN Access product can be used to include inventory information from LAN management tools, such as NetFinity in TME 10 Inventory.

Then, in Chapter 8, “Using RDBMS Tools to Access TME 10 Inventory Data” on page 199 we show how tools that come with Sybase and Oracle can be used to make working with TME 10 Inventory easier.

Finally, Chapter 9, “Summary” on page 231 concludes and summarizes the experiences gathered and gives some recommendations for planning the use of TME 10 Inventory.

1.4 Where to Find Code Examples

You can retrieve the code examples in this redbook from the International Technical Support Organization Raleigh Center.

IBM employees can point their Web browser to w3.itso.ral.ibm.com/~stefanu and follow the links to code examples.

People outside IBM can send e-mail requesting the code examples to uelpenich@vnet.ibm.com.

Chapter 2. Using TME 10 Inventory with Sybase on Windows NT

This chapter is an overview of necessary installation and configuration steps for using TME 10 Inventory in combination with the Sybase 11 RDBMS system. We show a basic TME 10 Inventory example to verify that the environment works the way we expect.

This chapter has two goals:

1. To show how to get TME 10 Inventory up and running in a Windows NT environment with Sybase.
2. To show how to have the RDBMS server separated from the TME 10 Inventory server.

2.1 Overview and Objective

The objective of this chapter is to successfully install Sybase Server Version 11 for Windows NT in the environment shown in Figure 7 on page 14. Notice in the figure that victor, which is not part of the foxtrot TMR, will act as a remote RDBMS server for the foxtrot TMR; that is, victor will hold the TME 10 Inventory database. We intentionally want the RDBMS server to not be part of any TMR to prove that this can be done. This important configuration option is often misunderstood.

Note

The key to understanding this configuration option is to see the difference between the RIM host and the RDBMS host. The RIM host must be a managed node, but this is just the node where access to the RDBMS host takes place. As a consequence, the RIM host needs to be able to "talk" to the RDBMS host, but does not necessarily have to be the RDBMS host. However, on the RIM host, you must have installed at least the RDBMS client support, in our case a Sybase client, to "talk" to the RDBMS (server) host. The RDBMS host is the node where the RDBMS server support is installed.

We set up the TME 10 Inventory application in the foxtrot TMR where foxtrot will be installed as the TME 10 Inventory server and RIM host. The TMR Managed Node, foxtrot, will also have a Sybase client installed. The Managed Node, romeo, in the foxtrot TMR will have the managed node TME 10 Inventory code installed while the two PC Managed Nodes, juliet and wtr05123, will have the TME 10 Inventory PC agent code installed.

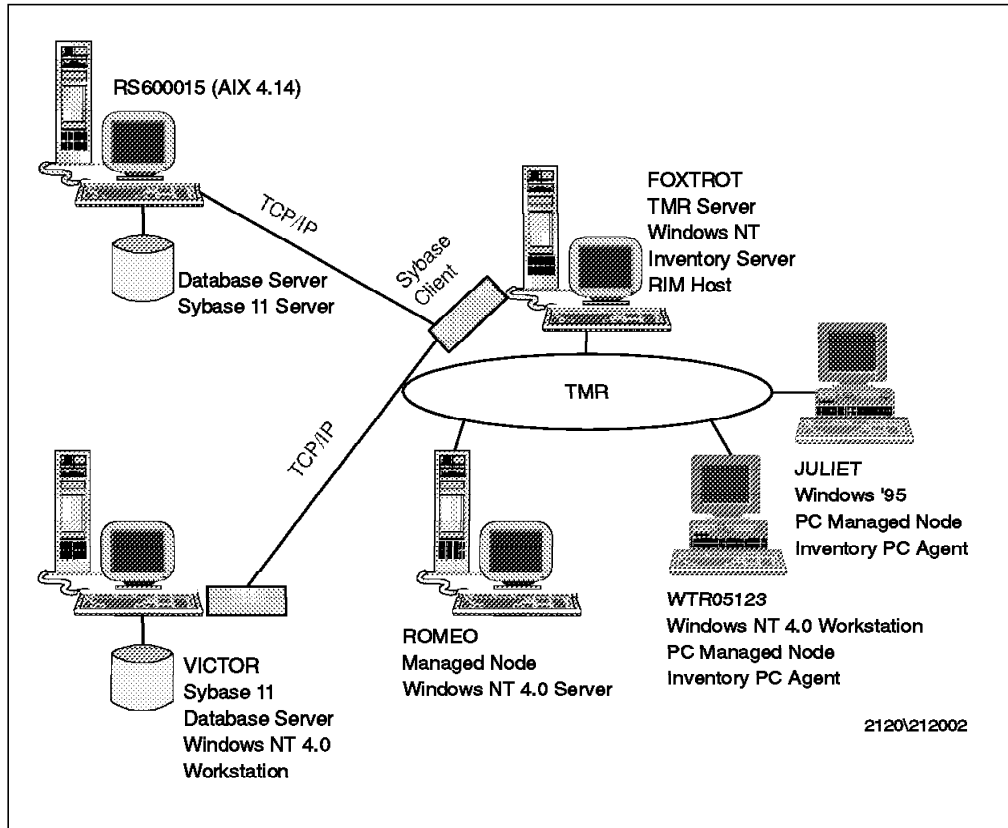


Figure 7. TME 10 Inventory in a Sybase Environment on Windows NT

Note

The machine rs600015 is not used in this chapter but later in Chapter 3, "Using TME 10 Inventory with Sybase on AIX" on page 47.

We perform the following steps:

- Install Sybase on the RDBMS server (victor).
- Configure Sybase on the RDBMS server.
- Install the Sybase client on the TME 10 Inventory server (foxtrot).
- Configure the Sybase client.
- Install TME 10 Inventory in the TMR.
- Configure Sybase for TME 10 Inventory.
- Run a basic inventory scan.

We use the following equipment in our scenario:

Foxtrot

Hardware : PS/2 Model 95, Pentium 60, 80 MB RAM, 1 GB Disk
 OS : Windows NT Server 4.0 with SP 2
 Protocol : TCP/IP
 Software : TME 10 Framework 3.1 Rev C
 TME 10 Inventory 3.1 Rev A
 Sybase Monitor Client

Victor

Hardware : PS/2 Model 95, Pentium 60, 32 MB RAM, 1 GB Disk
OS : Windows NT Workstation 4.0
Protocol : TCP/IP
Software : Sybase Server Version 11

Romeo

Hardware : PS/2 Model 95, Pentium 60, 64 MB RAM, 1 GB Disk
OS : Windows NT Server 4.0 with SP 2
Protocol : TCP/IP
Software : TME 10 Framework 3.1 Rev C
TME 10 Inventory 3.1 Rev A

Wtr05123

Hardware : PS Valuepoint, Pentium 66, 32 MB RAM, 570 MB Disk
OS : Windows NT Workstation 4.0
Protocol : TCP/IP
Software : Tivoli PC agent 4.007
Inventory PC agent

Juliet

Hardware : PS/2 Model 57, 486-SX25, 16 MB RAM, 540 MB Disk.
OS : Windows 95
Protocol : TCP/IP
Software : Tivoli PC agent v4.007
Inventory PC agent

2.2 Installing Sybase on Windows NT

Installing Sybase involves installing the Sybase RDBMS server on victor and installing the Sybase client on foxtrot (the TME 10 Inventory server and RIM host).

We start by installing the RDBMS server on victor.

Note

In this project, we use a version of Sybase SQL Server Professional for Windows NT. If you use a different version of Sybase, the windows, tools, etc. might look slightly different. However, the basic principles explained are still valid.

Run **Setup.exe** from the Sybase server source code directory. This will be most likely a CD-ROM directory. The Sybase SQL Server screen will appear. Click on **Yes** after you have read and agreed to the licensing agreement.

You will be prompted to enter the target path. The default is **c:\Sybase** which we used in our scenario. If you wish to change the target directory, enter it in the field provided and click on **Continue**.

The Sybase installation utility will auto-detect if your Windows is 16 or 32 bit. Since we are running Windows NT Workstation 4.0, it has defaulted to 32 bit. Click on **Continue**.

The Windows NT Product Selection Screen window will now appear. See Figure 8 on page 16.

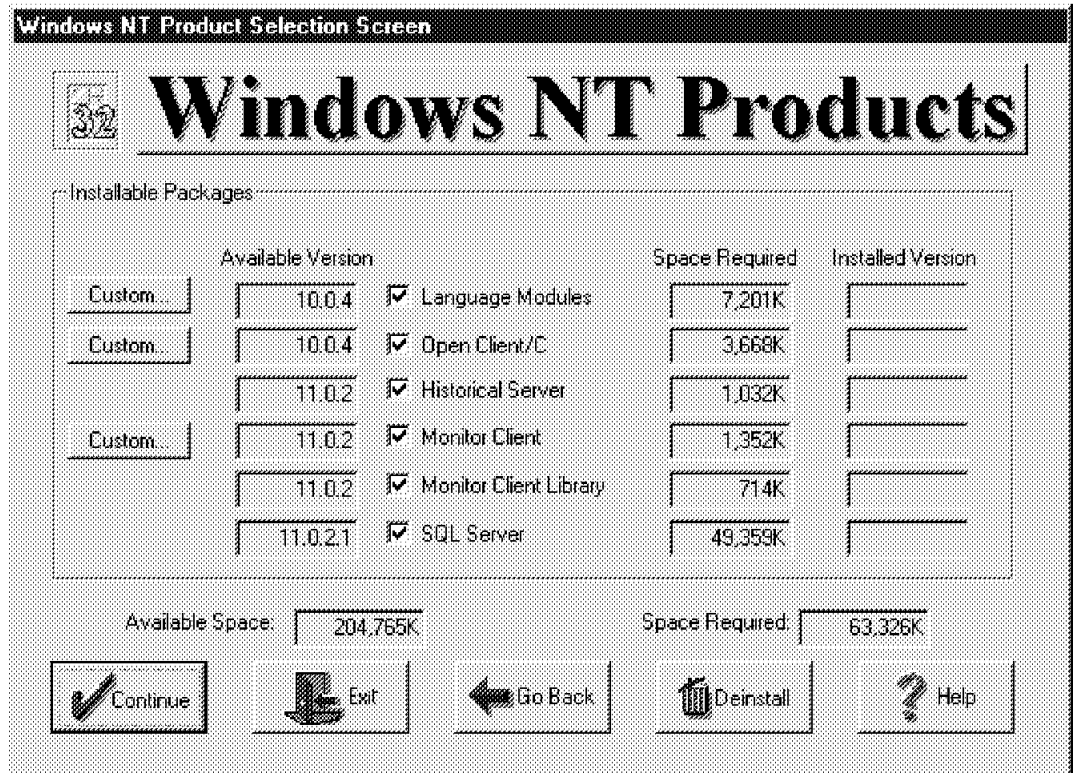


Figure 8. Windows NT Product Selection Screen Window

We leave all the default selections. If you want to know more about the installation defaults, use the online help facility or refer to the Sybase Installation guide.

Click on the **Custom...** button to the left of Monitor Client.

The Custom Selection window will appear. See Figure 9 on page 17.

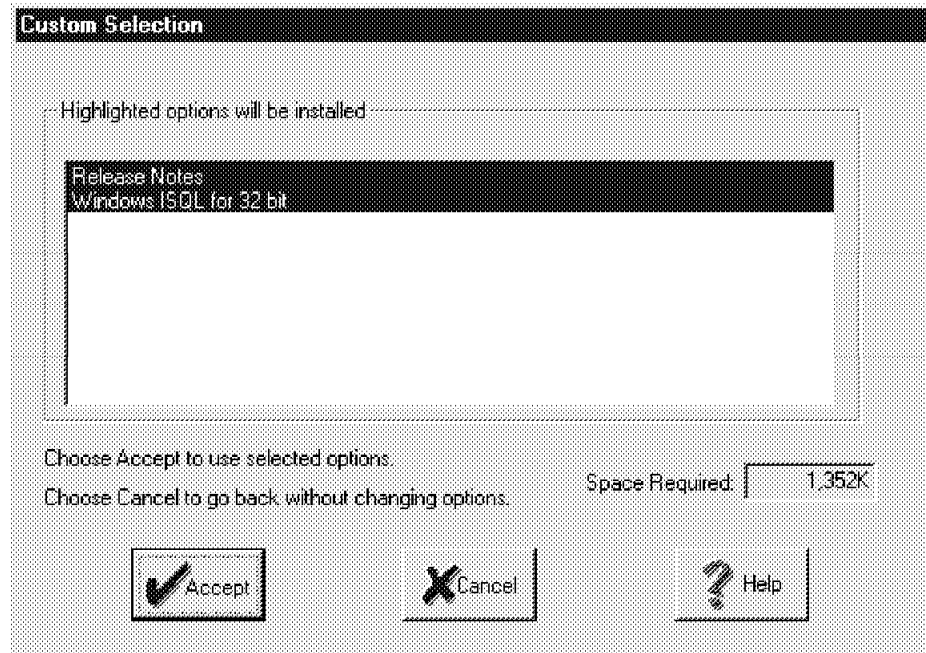


Figure 9. Custom Selection Window

Ensure that **Windows ISQL for 32 bit** is highlighted in the options list. The isql tool is the command line interface that is used to access the RDBMS server from the server itself or from a client. Refer to 9.2, "Selecting the Database Server Platform" on page 231 for a description of how TME 10 Inventory uses this kind of tool to access the RDBMS. Windows ISQL for 32 bit is a Windows-based version of isql (which itself is a simple command line style tool). We will use the Windows isql version, for example, to add the tables needed for inventory.

Click on **Accept**.

The installation will continue until the Install Additional Components window appears. See Figure 10.

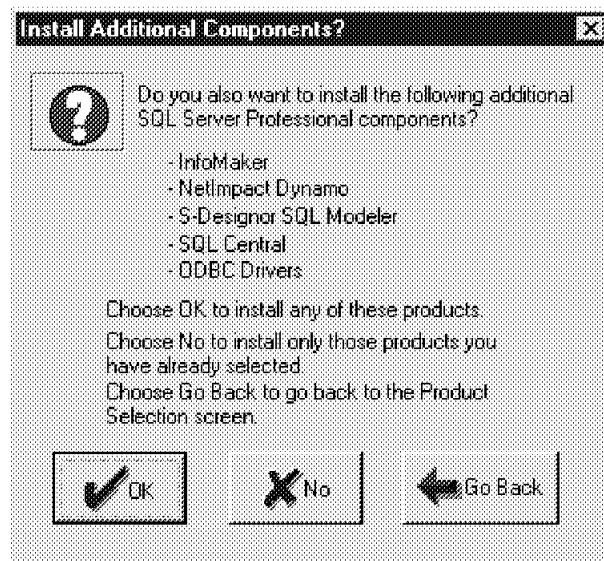


Figure 10. Install Additional Components Window

Click on **OK** to install additional components.

The installation will continue until the SQL.INI Dialog window appears. See Figure 11.

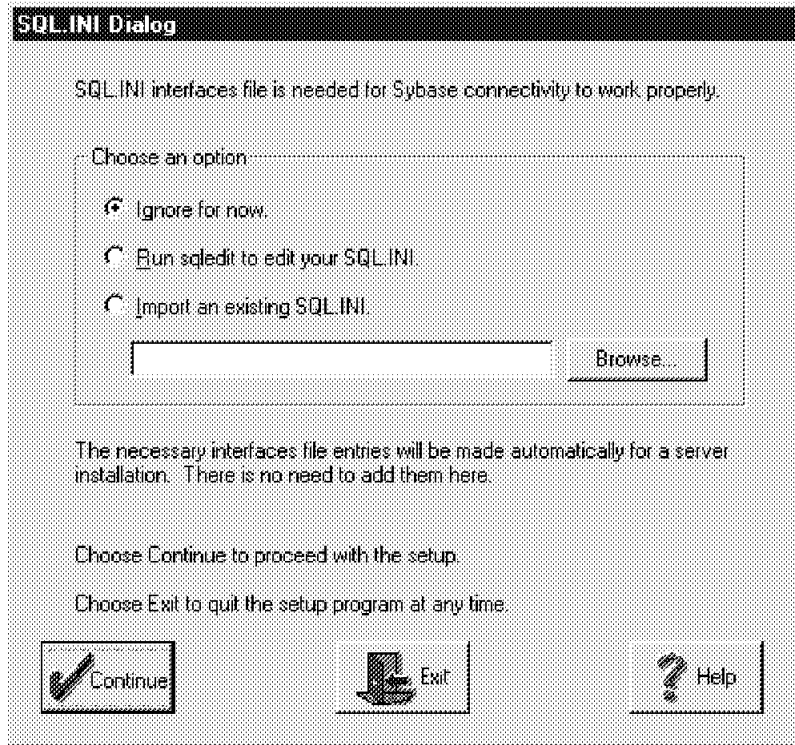


Figure 11. SQL.INI Dialog Window

Leave the radio selection on **Ignore for now** and click on **Continue**. The Server installation will automatically populate the SQL.INI file (also known as the Interfaces file) with entries which contain the server and associated network information.

We will view and edit the SQL.INI information in 2.3, “Configuring Sybase” on page 21.

Click on **Continue**.

The installation will continue until the SQL Server Configuration window appears. See Figure 12 on page 19.

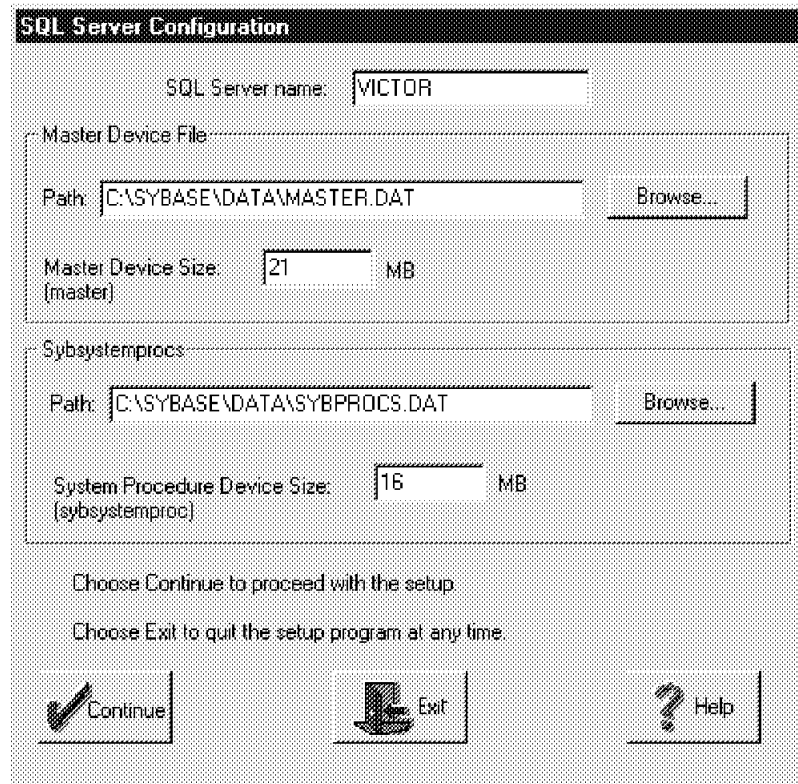


Figure 12. SQL Server Configuration Window

Our SQL Server name is VICTOR which we enter in the SQL Server name field. We choose to leave the Master Device File size at the default value of 21 MB in this example. We recommend, however, that you change the size to at least 70 MB. Otherwise, you are likely to get problems with TME 10 Inventory later as the device runs full.

Note

If you want to increase the Master Device File size, you must do it now because the size cannot be changed once the installation is complete.

The Master Device File contains all the system databases, namely:

- Master** Controls the general operation of the SQL Server as well as any new databases you create after installing SQL Server.
- Model** Provides a template for new databases you create.
- Tempdb** Provides a temporary workspace area for all SQL Server databases that is cleared and rebuilt each time SQL Server is started.

The System Procedure Device Size default is 16 MB and should be increased to accommodate future releases of Sybase and your own stored procedures. We left this size at 16 MB because we will not be upgrading Sybase during the course of this project.

Click on **Continue**.

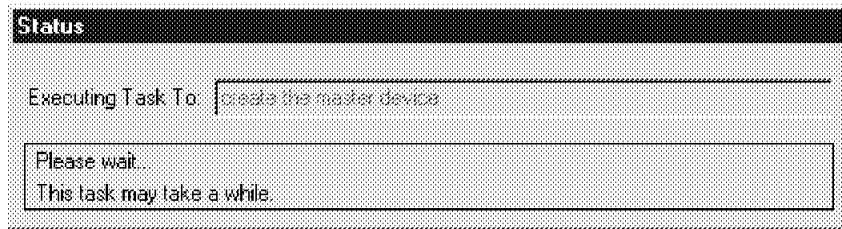


Figure 13. Creating Master Device File

The installation will now continue with the creation of the master device file which will take a few minutes depending on the size specified during the Sybase SQL Server configuration stage.

Click on **Next** to accept the default directories. The install process will now continue.

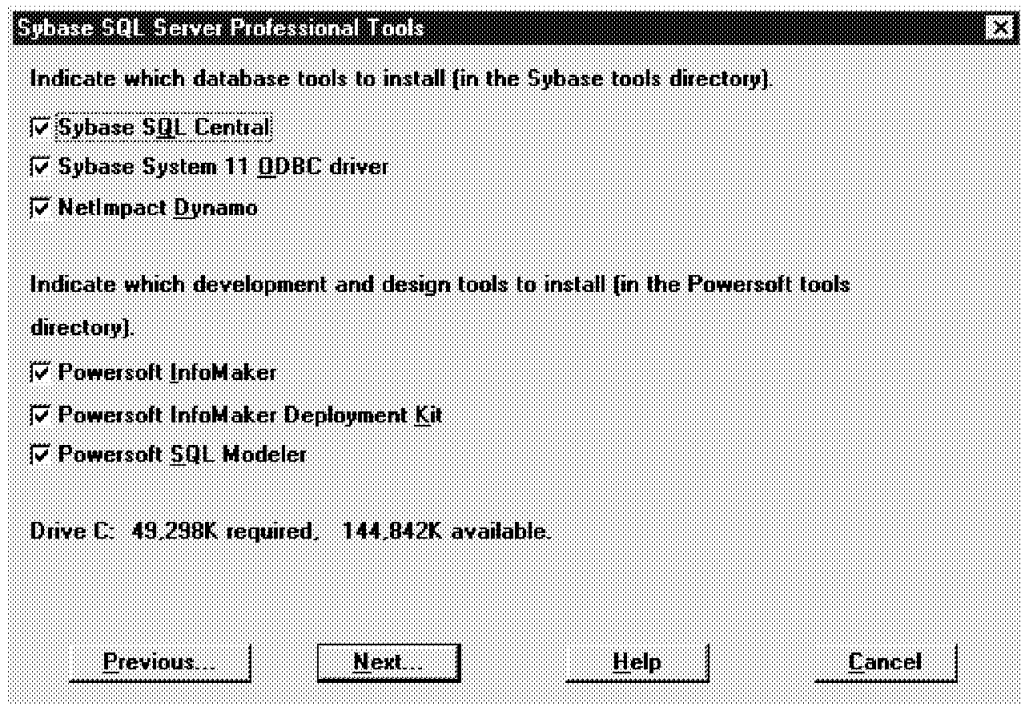


Figure 14. Sybase SQL Server Professional Tools Window

When the Sybase SQL Server Professional Tools window appears, install all the tools available.

Note
You do not have to install all the options, but we chose to do so to make sure we had all the features we actually needed for running with TME 10 Inventory.

Click on **Next**.

Click on **OK**.

Click on **OK** to let the system make all modifications for you and complete the installation.

Shut down and restart the machine.

2.3 Configuring Sybase

After installing Sybase, we can now configure Sybase SQL Server for use within our environment.

Start the SQL Server using the **Services Manager** tool that is supplied with Sybase.

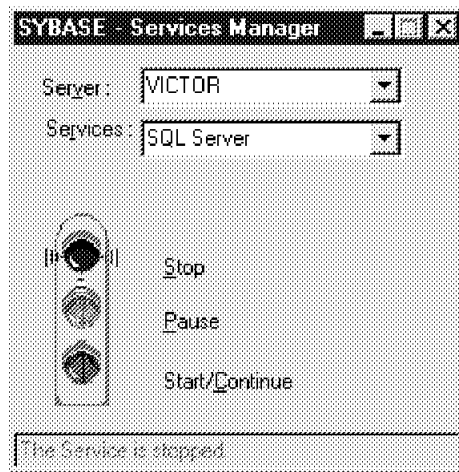


Figure 15. SYBASE - Services Manager Window

The Services Manager tool can be started by clicking on the **Start** menu button, **Programs**, **Sybase SQL Server Professional** and **Services Manager**.

The traffic light icon is a graphical representation of the SQL Server status. You will notice that the service is currently stopped. To start the service double-click on the green light or the **Start/Continue** text.

Possible status can be:

Red Stopped

Yellow Paused, not connected

Yellow with blue background
Paused, connected

Green Running, not connected

Green with blue background
Running, connected

All 3 low-lighted
Unknown, (not local machine), not connected

2.3.1 Changing the System Administrator Password

The installation process creates a system administrator user ID, the sa user, with no password by default. As with any administrator user ID, the default password should be changed for obvious security reasons.

Note

TME 10 Inventory will create its own database user named tivoli to access the database system. See 2.5, "Configuring Sybase on Windows NT for Use with TME 10 Inventory" on page 30 for details.

To change the user ID sa's password, you need to connect to the SQL Server. To connect to the SQL Server, click on the **Start** menu button, **Programs**, **Sybase SQL Server Professional** and **SQL Central**.

The SQL Central window will appear. See Figure 16.

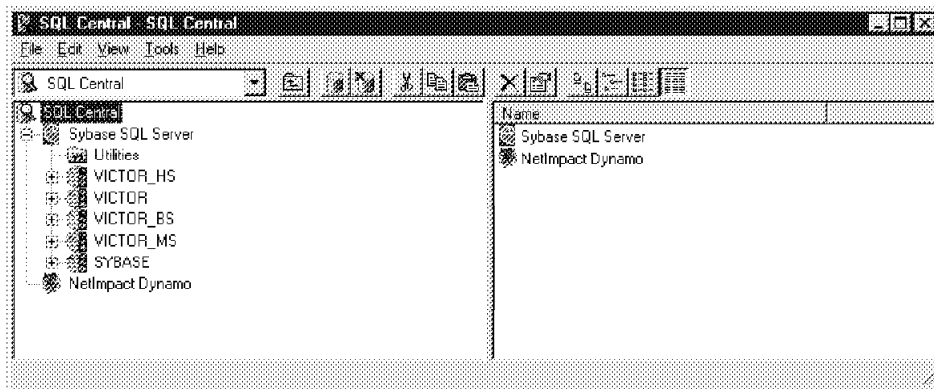


Figure 16. SQL Central Window

Click once with the left mouse button on **VICTOR**. The Connect to Sybase SQL Server window will appear. See Figure 17.

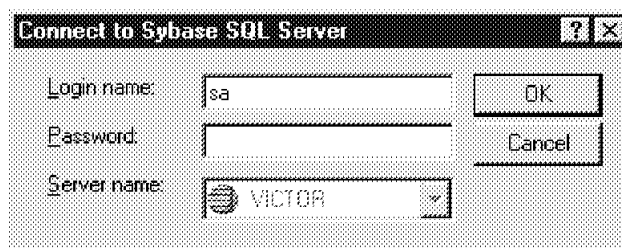


Figure 17. Connect to Sybase SQL Server Window

Enter sa (case sensitive) in the Login name field and leave the password field blank. Since the sa user doesn't have a password, we can log in by just giving the user name.

Click on **OK**.

This will connect you to the RDBMS server. If you encounter problems here, make sure that the RDBMS server is running.

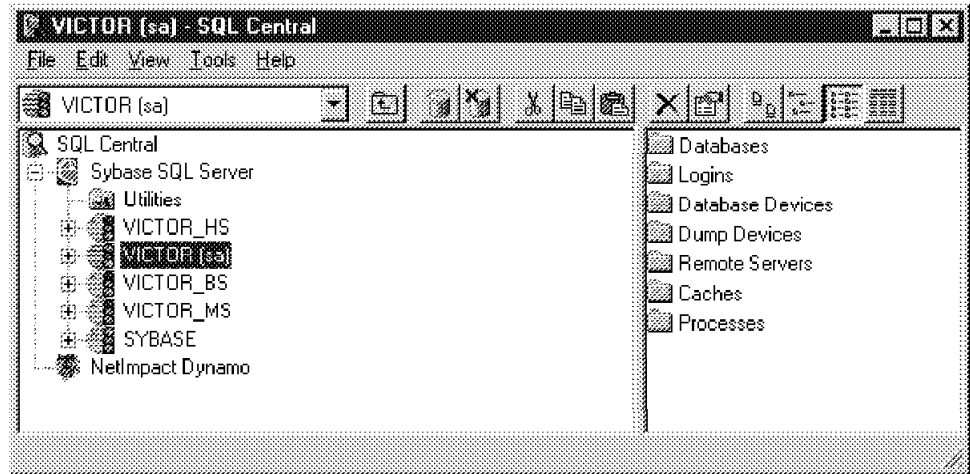


Figure 18. SQL Central Window

Back in the SQL Central window, you will notice that sa has appeared next to VICTOR in the window title. This is a visual reminder of who is currently connected which is useful when you have multiple login IDs defined.

Click on the plus (+) sign next to VICTOR to expand sub-components.

Double-click on the **Logins** folder to display login IDs. You will notice that the Logins folder has been fully expanded on the right-hand side of the window.

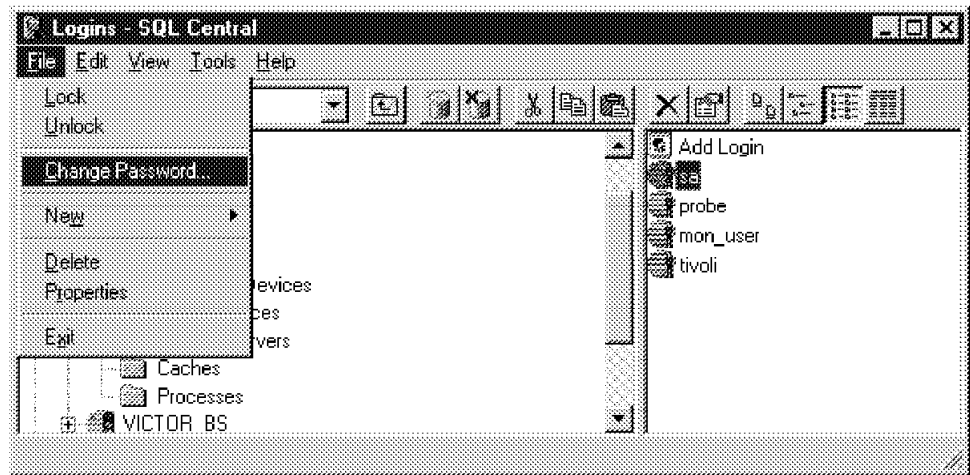


Figure 19. Change Password for sa User

Highlight **sa**. Click on **File, Change Password** from the pull-down menu.

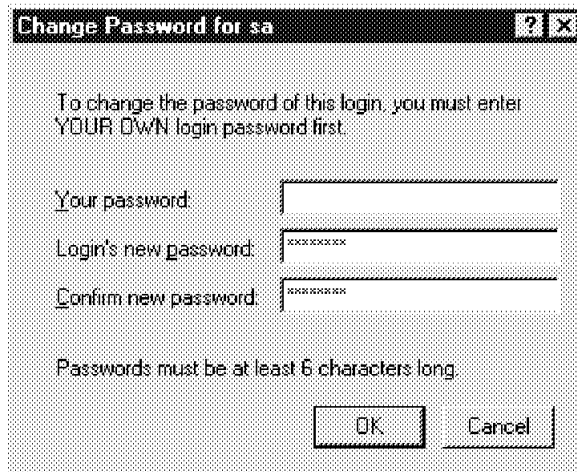


Figure 20. Change Password for sa Window

Change sa's password by entering and confirming the new password in the fields provided.

Click on **OK**.

Click on **Yes**.

2.3.2 Installing the Sybase Client

We need to install the Sybase client on the RIM host so that we can read and write information to and from the RDBMS server. In our scenario, foxtrot is the TME 10 Inventory server and RIM host, and victor is the RDBMS server. Therefore, we will be installing the Sybase client on foxtrot.

For this to work, there needs to be a network connection between foxtrot and victor. In our case, victor and foxtrot can communicate over TCP/IP.

Before we start the installation on foxtrot we need to know some connection information about victor. The required information can be found in the c:\sybase\bin\SQL.INI file on victor.

The Sybase client connects to the RDBMS server using a network connection, for example, a TCP/IP socket connection. To describe this connection on the client, we need to find out the details of the connection to the server.

To view this information on victor, use the SQLEDT tool supplied with Sybase. This step is performed on victor.

From the Windows NT taskbar, click on **Start, Programs, Sybase SQL Server Professional** and **SQLEDT**.

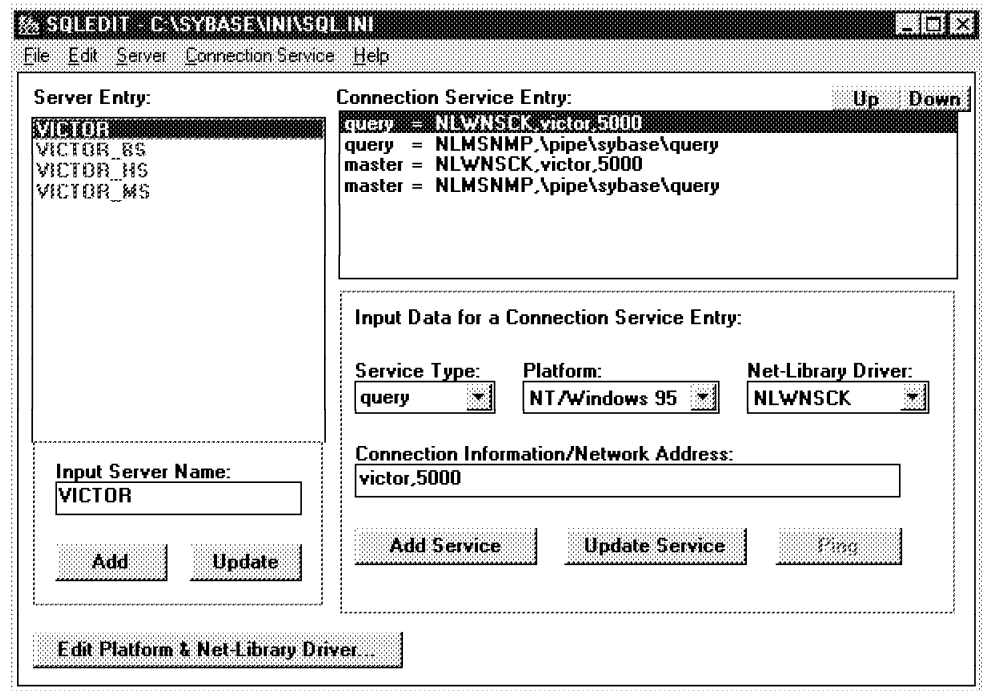


Figure 21. SQLEDT Window

The above window displays the contents of the SQL.INI file and allows you to modify the contents of the file in a convenient and safe fashion.

On the left side you see the RDBMS server processes, VICTOR, which is the RDBMS server, VICTOR_BS (the backup server) and so on. For each of these processes you will find a Connection Service Entry on the right side of the window. You will notice that there are two entries for each process, query and master, with a description of how they are connected to the server process.

After the installation, there are default entries for the RDBMS server that allow us to connect using named pipes and a WINSOCK connection. By selecting the WINSOCK connection, as shown in the above window, you will see the details of the connection.

In the case of VICTOR, the TCP/IP connection is described in the following way: The connection to the VICTOR database is made using the TCP/IP hostname victor (which in our example is the same as the RDBMS server name, but does not necessarily have to be) and port 5000. Similarly, the servers VICTOR_BS, etc. also use the TCP/IP hostname victor, but different port numbers.

For the client to connect to the server, you have to create the SQL.INI file on the client that contains the same kind of connection information. In our case, we will make a TCP/IP connection, so you need to note basically that the TCP/IP hostname of the RDBMS server is victor and the ports used for the different server processes are 5000, 5001, 5002, and 5003.

The same information can also be found in the SQL.INI file on victor which is shown in Figure 22 on page 26.

```

;; Sybase Interfaces file
;;
;; [<database_server_name>]
;; <link_type>=<network_driver>,<connection_info>
;;
;; Examples:
;; [JUPITER]
;; QUERY=NLMSNMP,\\JUPITER\pipe\sybase\query
;; WIN3_QUERY=WNLNMP,\\JUPITER\pipe\sybase\query
;;
[VICTOR_HS]
$BASE$00=NLWNSCK,victor,5003
$BASE$01=NLMSNMP,\pipe\sybase\hist
query=$BASE$00;$BASE$01;
$BASE$02=NLWNSCK,victor,5003
$BASE$03=NLMSNMP,\pipe\sybase\hist
master=$BASE$02;$BASE$03;

[VICTOR]
$BASE$00=NLWNSCK,victor,5000
$BASE$01=NLMSNMP,\pipe\sybase\query
query=$BASE$00;$BASE$01;
$BASE$02=NLWNSCK,victor,5000
$BASE$03=NLMSNMP,\pipe\sybase\query
master=$BASE$02;$BASE$03;

[VICTOR_BS]
$BASE$00=NLWNSCK,victor,5001
$BASE$01=NLMSNMP,\pipe\sybase\backup
query=$BASE$00;$BASE$01;
$BASE$02=NLWNSCK,victor,5001
$BASE$03=NLMSNMP,\pipe\sybase\backup
master=$BASE$02;$BASE$03;

[VICTOR_MS]
$BASE$00=NLWNSCK,victor,5002
$BASE$01=NLMSNMP,\pipe\sybase\monitor
query=$BASE$00;$BASE$01;
$BASE$02=NLWNSCK,victor,5002
$BASE$03=NLMSNMP,\pipe\sybase\monitor
master=$BASE$02;$BASE$03;

[SYBASE]
query=NLWNSCK,rs600015,3000
master=NLWNSCK,rs600015,3000

```

Figure 22. SQL.INI File on victor

The following tasks need to be performed on the Sybase client, which is foxtrot in our environment:

- Install the Sybase client.
- Configure the SQL.INI file.

Since our Sybase installation images reside on victor, we connected a network drive through Windows NT to victor from foxtrot (a shared resource in Windows NT). You can also just use the Sybase installation CD-ROM on foxtrot.

Run **Setup.exe** from the Sybase server source code directory. The Sybase SQL Server screen will appear. Click on **Yes** after you have read and agreed to the licensing agreement.

You will be prompted to enter the target path. The default is **c:\Sybase**, which we use in our scenario. If you wish to change the target directory, enter it in the field provided and click on **Continue**.

The Sybase installation utility will auto-detect if your Windows is 16 or 32 bit. Since we are running Windows NT Server 4.0, it has defaulted to 32 bit. Click on **Continue**.

The Windows NT Product Selection Screen window will appear. See Figure 23.

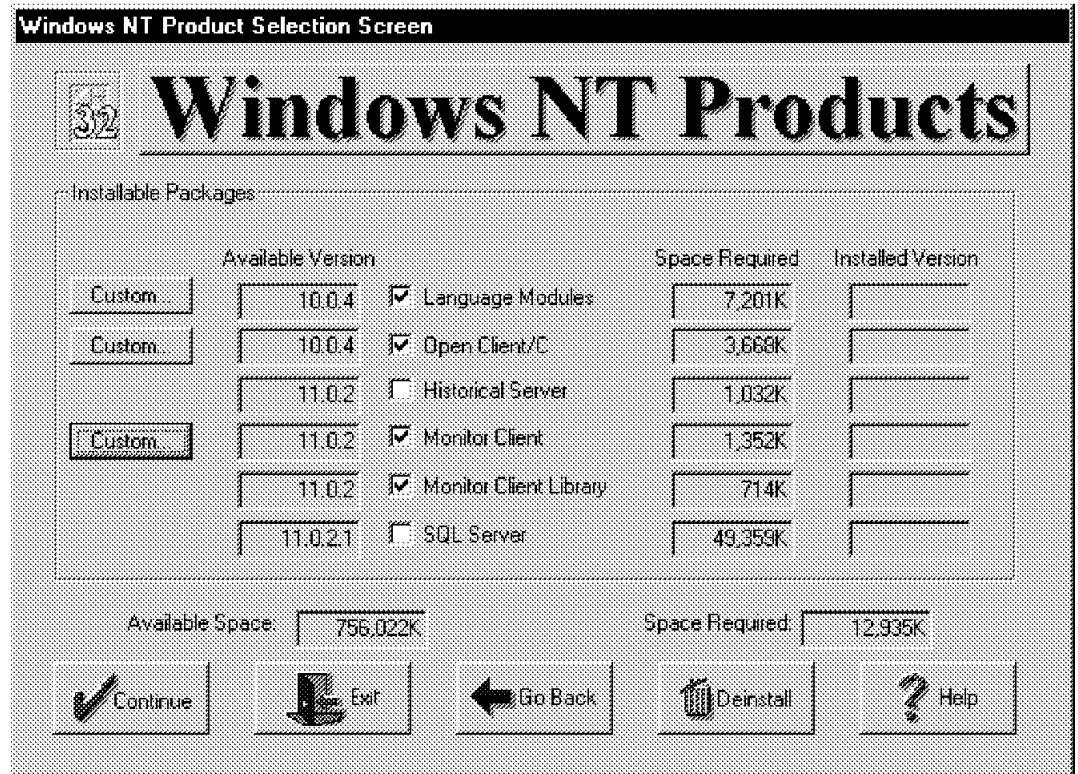


Figure 23. Windows NT Product Selection Screen Window

Select the client products as shown above and click on **Continue**.

Note

You should notice that SQL Server is not selected this time, since we are installing a Sybase client and not the RDBMS server.

The installation will continue until the SQL.INI Dialog window appears. See Figure 24 on page 28.

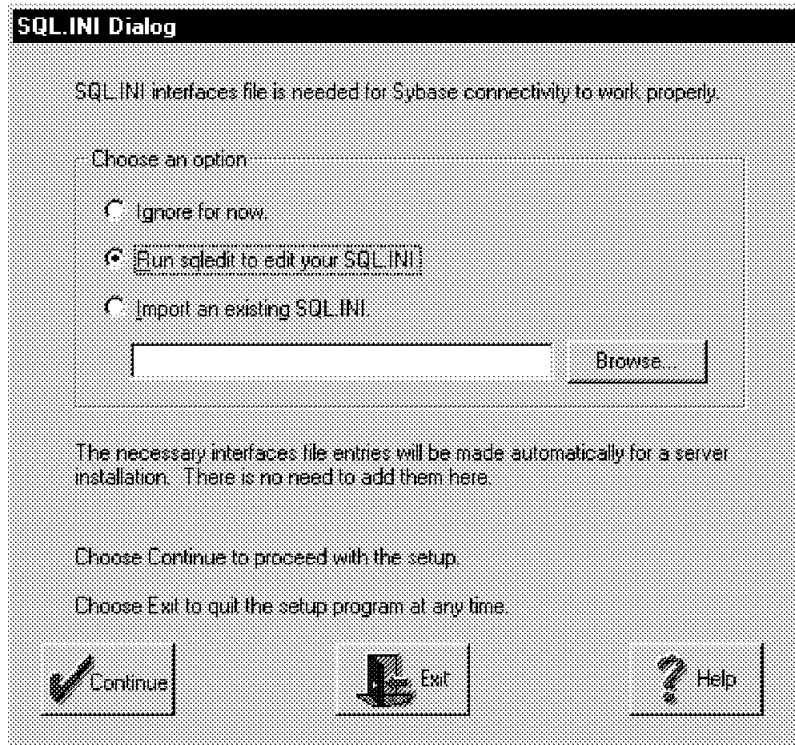


Figure 24. SQL.INI Dialog Window

When the SQL.INI dialog appears, select **Run Sqledit to edit your SQL.INI** and then click on the **OK** button. This will start the SQLEDIT tool.

Enter the information you recorded from victor's SQL.INI into foxtrot's SQL.INI and continue with the installation which is very straight forward.

The information we enter in our scenario is adding query and master connection to VICTOR, VICTOR_BS, VICTOR_HS and VICTOR_MS. We use a WINSOCK connection (NLWNSCK) with victor as the TCP/IP hostname and 5000 as the port for VICTOR.

When you have finished entering all definitions, select **File** from the menu bar and **Save** from the pull-down menu. Then close the SQLEDIT tool.

After the installation of the client has finished, you might want to test if the Sybase client is able to connect to the server. There are several tools to do this, for example, you can use the SQLPING tool that comes with Sybase or use the Windows ISQL tool to connect to the database.

2.4 Installing TME 10 Inventory in the TMR

We give only an overview of how to install TME 10 Inventory in this section and point out the configuration steps that are specific for our environment.

If you want a detailed explanation, refer to the *TME 10 Cookbook for AIX Systems Management and Networking Applications*, SG24-4867.

Note

It is always a good idea to take a database backup before installing any new products in your TMR.

We install TME 10 Inventory 3.1 for NT Only on foxtrot, which will be the TME 10 Inventory server and RIM host in our environment.

Figure 25 shows the TME 10 Inventory install options for the environment we used during the installation.

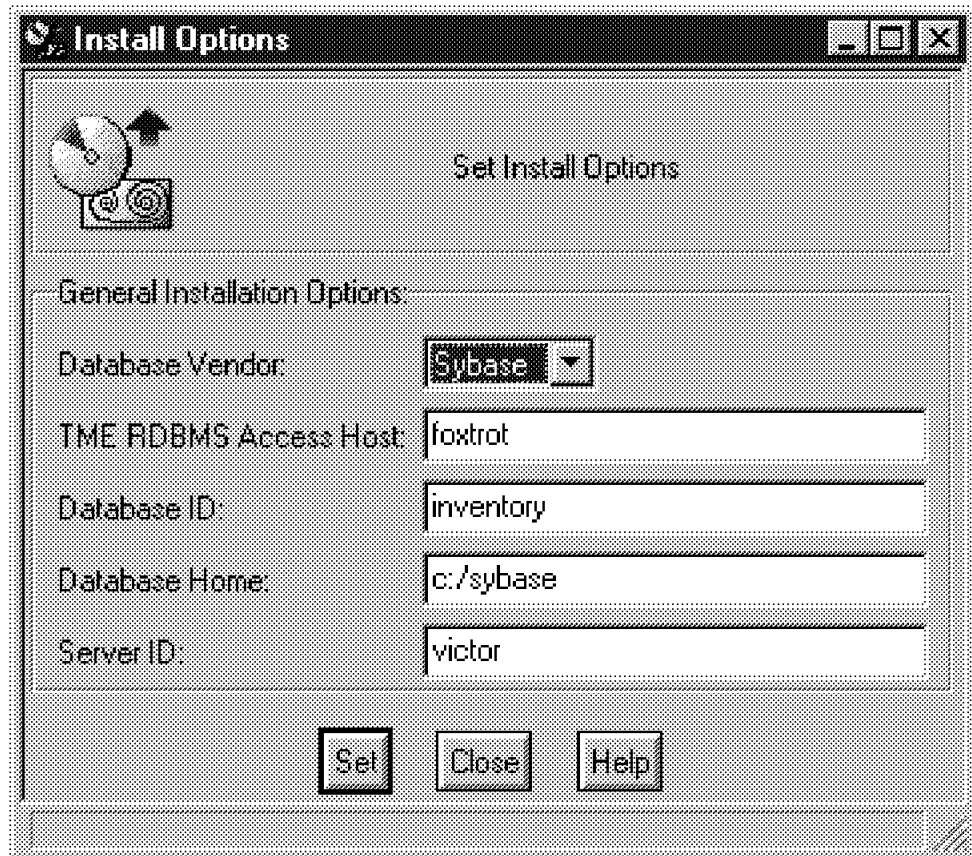


Figure 25. Install Options Window

The parameters you specify in the above window are sometimes confusing, especially if you do not know exactly how TME 10 Inventory accesses the database.

The TME RDBMS Access Host (which is sometimes also called the RIM host) is *not* the RDBMS server but the host on which RIM resides and is executed. Although in a lot of cases this is the same host (when the TME 10 Inventory server and the RDBMS are on the same host), this scenario is a good example where this is different.

We specify foxtrot here, as this is the managed node where the RDBMS access takes place (through the Sybase client), but foxtrot is not the database server.

Note

In theory, the RIM host does not even need to be the same as the TME 10 Inventory server, but just needs to be any managed node. However, the current version of TME 10 Inventory for Windows NT (3.1) requires that the TME 10 Inventory server and the RIM host are the same node.

The Database Home in our case is C:sybase. This is the database home directory of the Sybase client in our case, as this is where RIM tries to find isql to access the database. In our case this is also the database home of the server, but that might be different in other configurations.

In Chapter 3, "Using TME 10 Inventory with Sybase on AIX" on page 47, we show an example where the TME 10 Inventory server (RIM host) is on Windows NT and the Sybase server on AIX. In this case, Database Home would still be C:sybase although the database home directory on the server is most likely /usr/local/sybase. This shows that Database Home is the directory used on the RIM host and not on the RDBMS server necessarily. We also install TME 10 Inventory 3.1 for NT Only on romeo since romeo is a Windows NT Managed Node. All Windows NT managed nodes in your environment must have this portion installed if you want a hardware and software inventory from them.

We then install the TME 10 Inventory PC Agent on juliet and wtr05123 which are PC Managed Nodes in our environment.

Note

If you are installing the TME 10 Inventory PC Agent on multiple PC managed node types at the same time, we recommend that you install Windows 95 and Windows NT 3.51 separately from Windows NT 4.0. We tried to install to Windows NT 4.0 many times, but the product installation never reports back that it is registered and complete, although when you initiate an inventory scan, it is successful.

Once the TME 10 Inventory installation is complete, you can add the InventoryProfile managed resource to your policy region and create an Inventory profile manager. Details of how to do this can be found in the *TME 10 Cookbook for AIX: Systems Management and Networking Applications*, SG24-4867.

2.5 Configuring Sybase on Windows NT for Use with TME 10 Inventory

Now that Sybase SQL Server, the Sybase client and TME 10 Inventory have been installed in our scenario, we can integrate their functionality.

We have to create the TME 10 Inventory database and tablespace from foxtrot on victor using the WISQL32 tool supplied with Sybase. This will also serve as a test to ensure foxtrot can communicate with victor.

Note

You can also use the isql command, which is a simpler version and is what is recommended in the TME 10 Inventory User Guide; however, we think that WISQL32 is easier and more convenient to use.

Open the WISQL32 interface from a Windows command prompt by typing **Wisql32**. The WISQL32 window will appear. See Figure 26 on page 31.

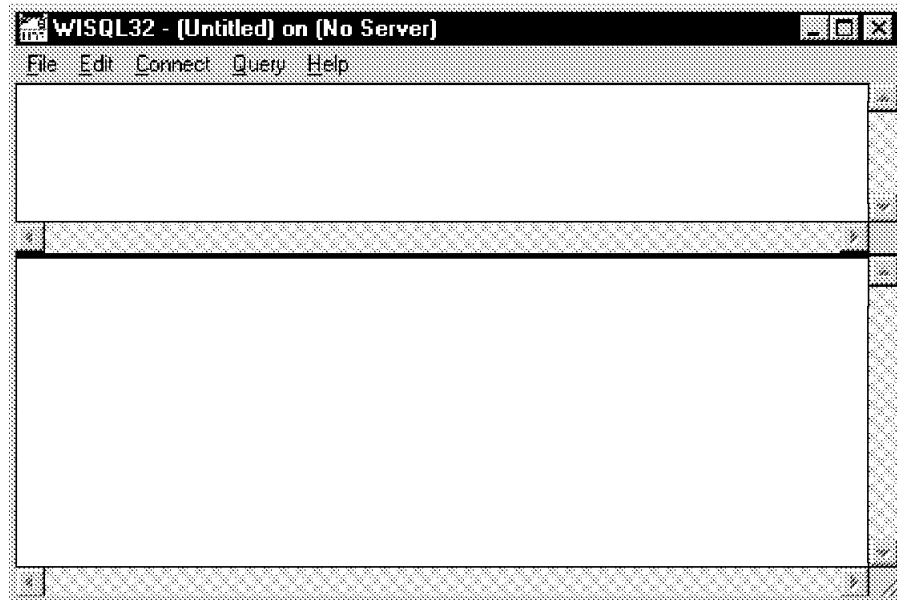


Figure 26. WISQL32 Window

We first need to connect to the RDBMS server.

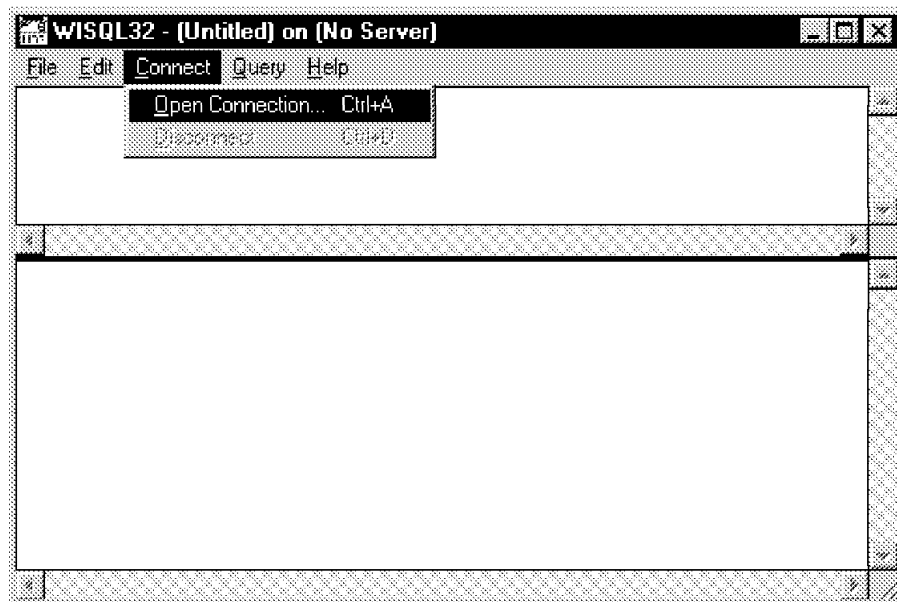


Figure 27. Open Connection from foxtrot to victor

Click on **Connect, Open Connection** from the menu bar. You will be asked for connection information.

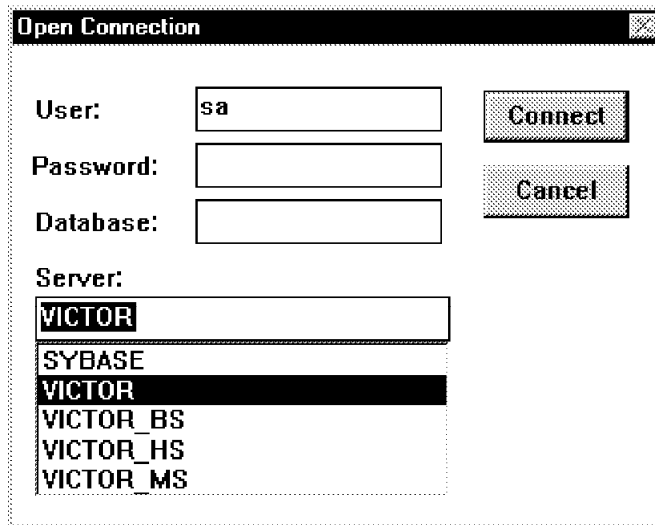


Figure 28. Open Connection Window

Type **sa** in the User field.

Type in your password in the Password field.

Leave the Database field blank.

Type **VICTOR** in the Server field.

Click on **Connect**.

You will now be connected to the master database on VICTOR. If the connection fails, look to see if your SQL.INI file on the client is configured correctly and try to see if you can do an SQLPING.



Figure 29. WISQL32 Window

You will notice the title bar has changed to reflect **VICTOR/master:sa** where:

- VICTOR is the RDBMS server.
- master is the database.
- sa is the user ID of the database user and administrator.

Now that we are connected to the RDBMS server, we can create the database, tables etc. needed for TME 10 Inventory. This is done by running two SQL scripts that are part of the TME 10 Framework and will make the necessary modifications to the RDBMS.

Click on **File, Open** from the menu bar. Change to the source directory of the SQL scripts that are supplied with the TME 10 Framework 3.1 namely:

- tivoli_syb_admin.sql
- tivoli_syb_schema.sql

Note

If you have problems finding the path to the files, type the following command from the drive letter where TME 10 Framework is installed:

```
dir /s *.sql
```

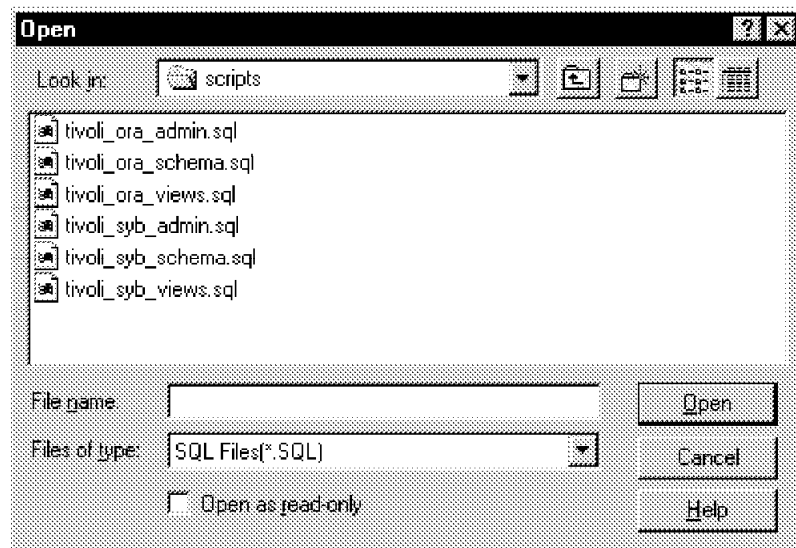


Figure 30. Open Scripts Needed to Be Run

Highlight the first script to be run which is **tivoli_syb_admin.sql**. Click on **Open**.

You will be returned to the WISQL32 window.

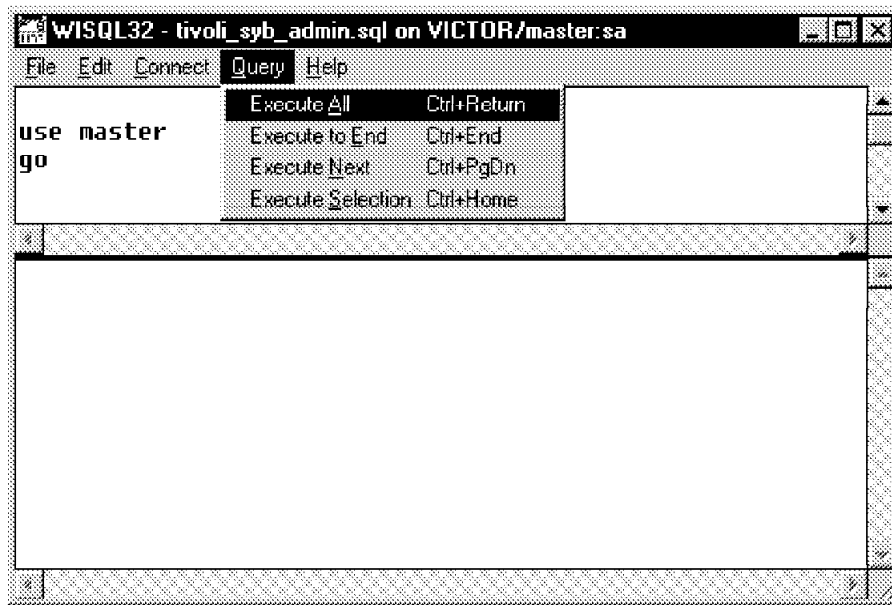


Figure 31. WISQL32 Window

Click on **Query, Execute All** from the pull-down menu. The SQL script will now execute on the RDBMS server.

Note

The name query might be confusing in this context since we are not running a query but creating the database. This option just means running any form of SQL script.

This script creates the tivoli user and the inventory database in the RDBMS on victor. The tivoli user's password is also tivoli.

The default size of this database is 20 MB. As a general rule 500 KB are required for each system whose inventory you want to store so if you have 1000 systems your database will need to be 500 MB in size. If you want to change the size of the database, modify the tivoli_syb_admin.sql script.

When running the SQL scripts that come with TME 10 Framework and TME 10 Inventory, you might see database warnings or errors when running these scripts. These do not necessarily mean that something is wrong, for example, the script that creates the tables, first tries to drop existing tables. If these tables do not exist, this will cause an error message but does not mean any problems.

Now that the tivoli_syb_admin.sql is complete and the tivoli user ID and TME 10 Inventory database has been created on victor, we need to create the database schema.

You must log off the sa user and log on using the tivoli user ID before running the tivoli_syb_schema.sql script. This needs to be done because the tivoli user ID owns the database and the TME 10 Inventory server communicates with the database through this user ID. If you run the tivoli_syb_schema.sql script under the sa ID, the TME 10 Inventory application will not be able to communicate with this database.

Follow the same process described above using the user ID tivoli and password tivoli for the tivoli_syb_schema.sql script.

A new script is included with TME 10 Inventory 3.1 that needs to be run in addition to the tivoli_syb_schema.sql script. If you are upgrading from TME 10 Inventory 3.0 to TME 10 Inventory 3.1, then you must run the syb_sp1_schema.sql BEFORE upgrading to TME 10 Inventory 3.1. If this is a first time installation, you need to run the syb_sp1_schema.sql AFTER the tivoli_syb_schema.sql script.

This SQL script must be run with the user ID tivoli because it updates the database tables created by tivoli_syb_schema.sql script.

If you now go back to the SQL Central tool on the RDBMS server and look at the Logins section, you will see the new tivoli user. This also proves that connectivity between the Sybase server and client is working, because we have created the database, user, etc. from the client foxtrot and can see the effects on the RDBMS server victor.

Figure 32 shows SQL Central on victor.

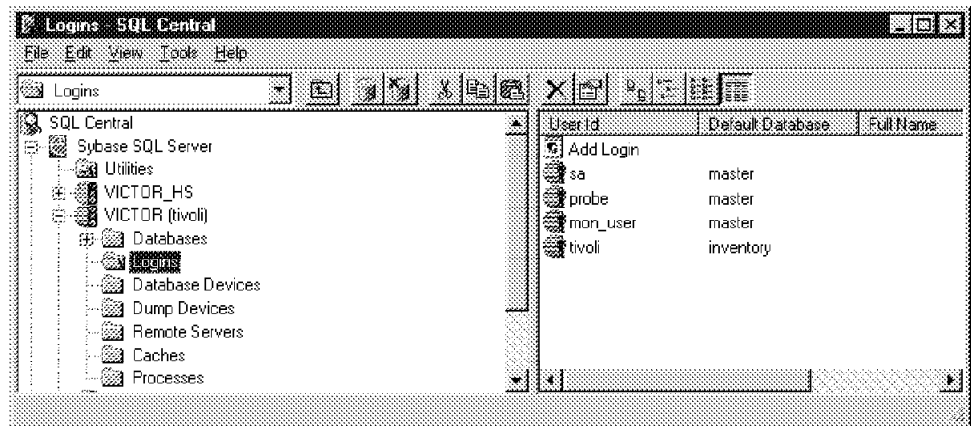


Figure 32. Check Tivoli User ID

Now that all the scripts have been run, we can do another test of SQL connectivity between the RIM host foxtrot and the inventory database on victor. This will be done using the WISQL32 tool on foxtrot again.

Open the WISQL32 interface if its not already open by typing **Wisql32** at a Windows command prompt on foxtrot.

Repeat the steps done previously to connect to the RDBMS server, but this time specify tivoli as the User and Password and inventory as the Database as shown in Figure 33 on page 36.

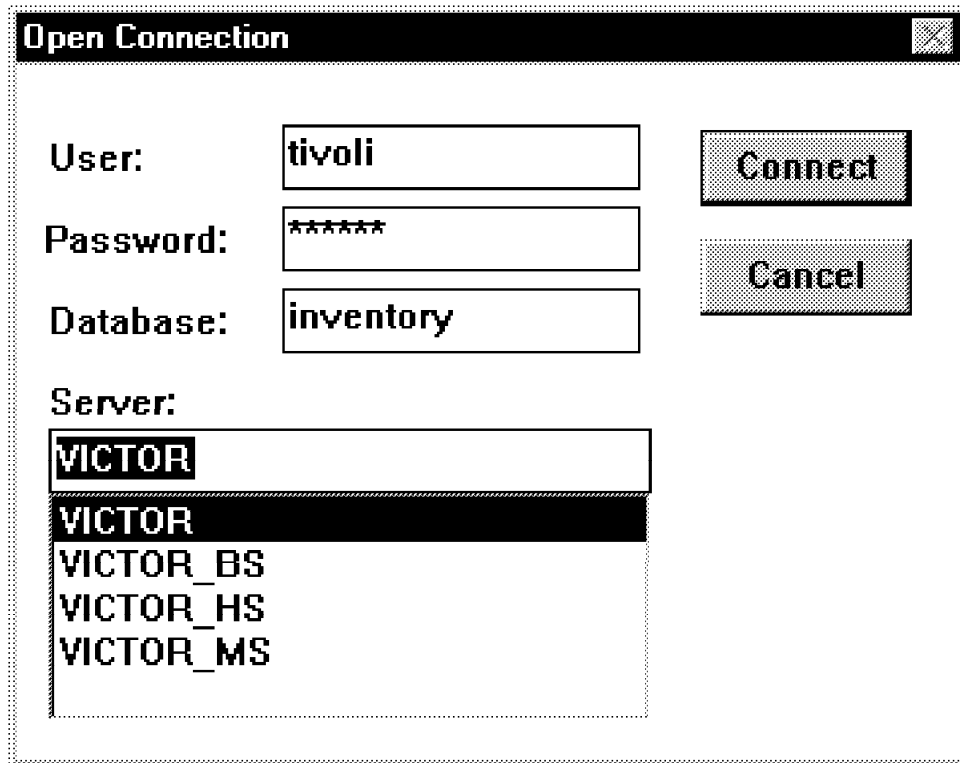


Figure 33. Open Connection to Inventory Using Tivoli User ID

Click on **Connect** to connect to the RDBMS server.

You will notice that the title bar in Figure 34 on page 37 has changed to reflect VICTOR/inventory:tivoli where:

- VICTOR is the RDBMS server.
- inventory is the database.
- tivoli is the user ID.

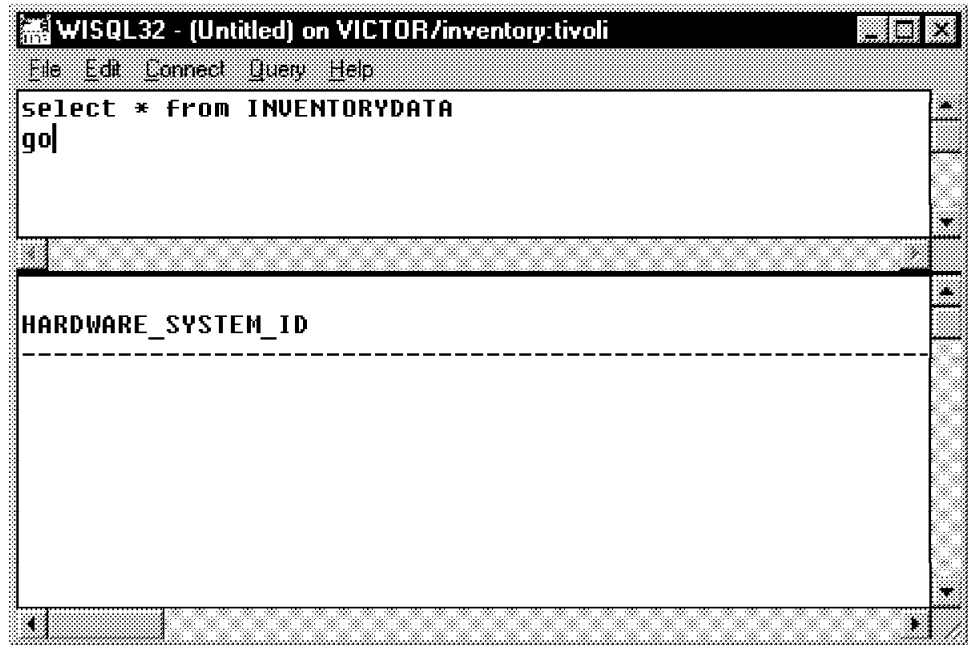


Figure 34. WISQL32 Window

In the query section enter:

```
select * from INVENTORYDATA
go
```

Select **Query, Execute all** from the pull-down menu.

If the table header is displayed, then Sybase, TME 10 Inventory and TCP/IP are configured correctly.

2.6 An Example Using Sybase on Windows NT and TME 10 Inventory

We set up and execute a TME 10 Inventory profile to perform an initial inventory check for the target nodes in our environment. We set up and run the profile from the TME Desktop on romeo. This is just a simple example to confirm that TME 10 Inventory is working correctly in our environment.

Start the TME 10 Desktop.

Open the policy region in which your nodes reside. Create and then open an Inventory profile manager. The Profile Manager window will appear. See Figure 35 on page 38.

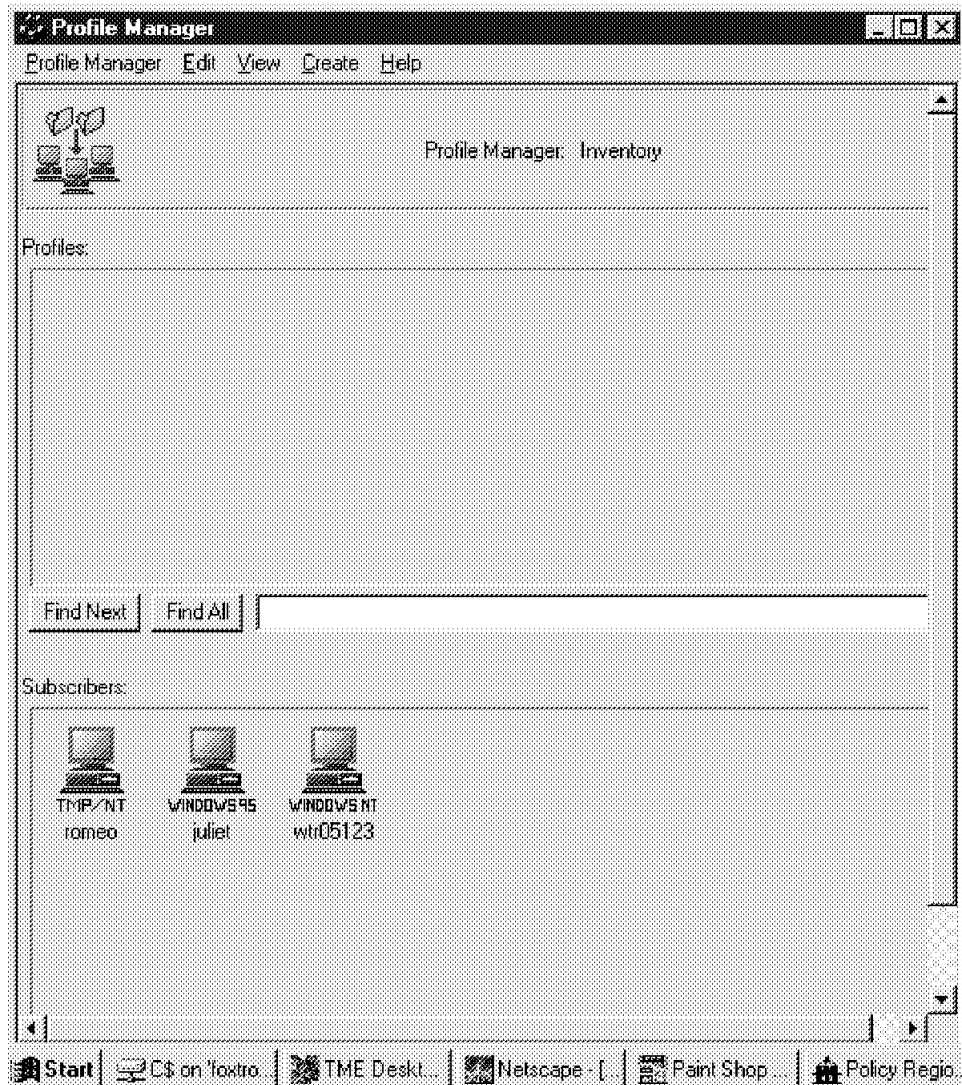


Figure 35. Profile Manager Window

Add subscribers to the Inventory profile manager. In our scenario, we run an inventory scan to romeo, juliet and wtr05123. Once the subscribers have been added, we create a profile. Select **Create** from the menu bar and then **Profile...** from the pull-down menu.

The Create Profile window will appear. See Figure 36 on page 39.

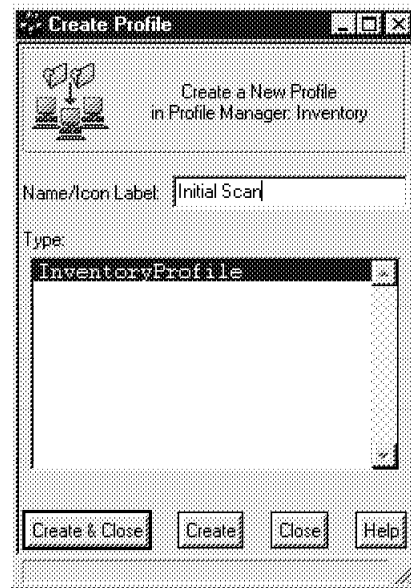


Figure 36. Create Profile Window

Enter Initial Scan in the Name/Icon Label field and click on **Create & Close**. The Initial Scan profile will be added to Profiles as shown in Figure 37 on page 40.

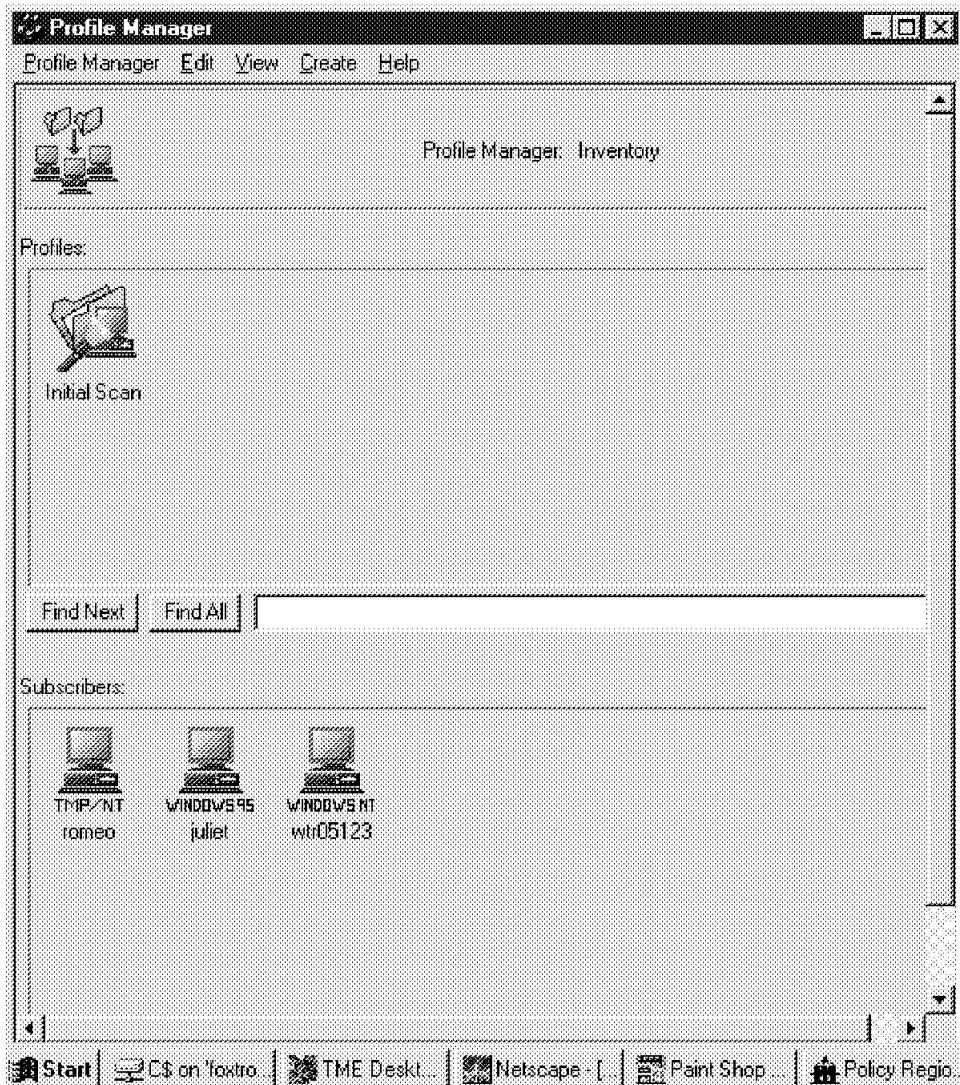


Figure 37. Profile Manager Window

We now open the profile to check the customization options. Double click on **Initial Scan**. The window in Figure 38 on page 41 will appear.

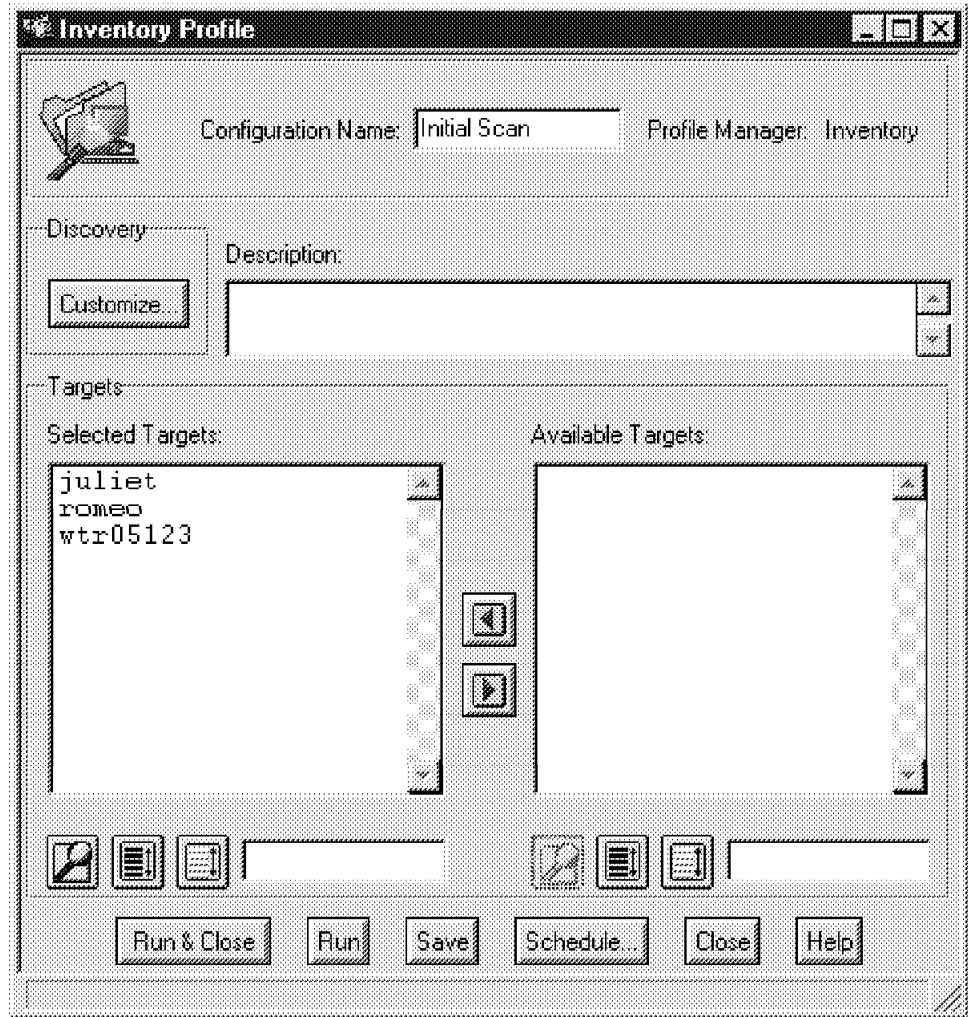


Figure 38. Inventory Profile Window - Initial Scan Settings

Move the targets you want to scan from the Available Targets section to the Selected Targets section. In our scenario we move romeo, juliet and wtr05123. Click on **Customize....**

The window in Figure 39 on page 42 will appear.

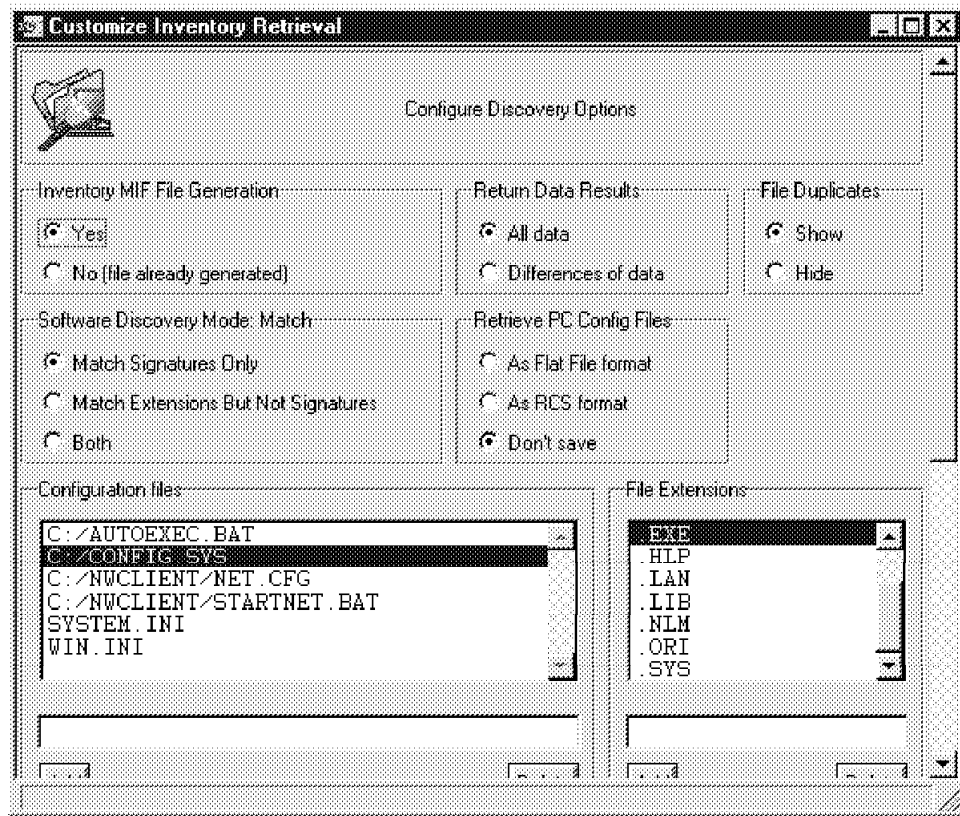


Figure 39. Customization Window (Part 1)

For an explanation on all customization options, refer to the TME 10 Inventory manuals.

For now, we only change the Return Data Results from the default Differences of data to All data as this is the first scan.

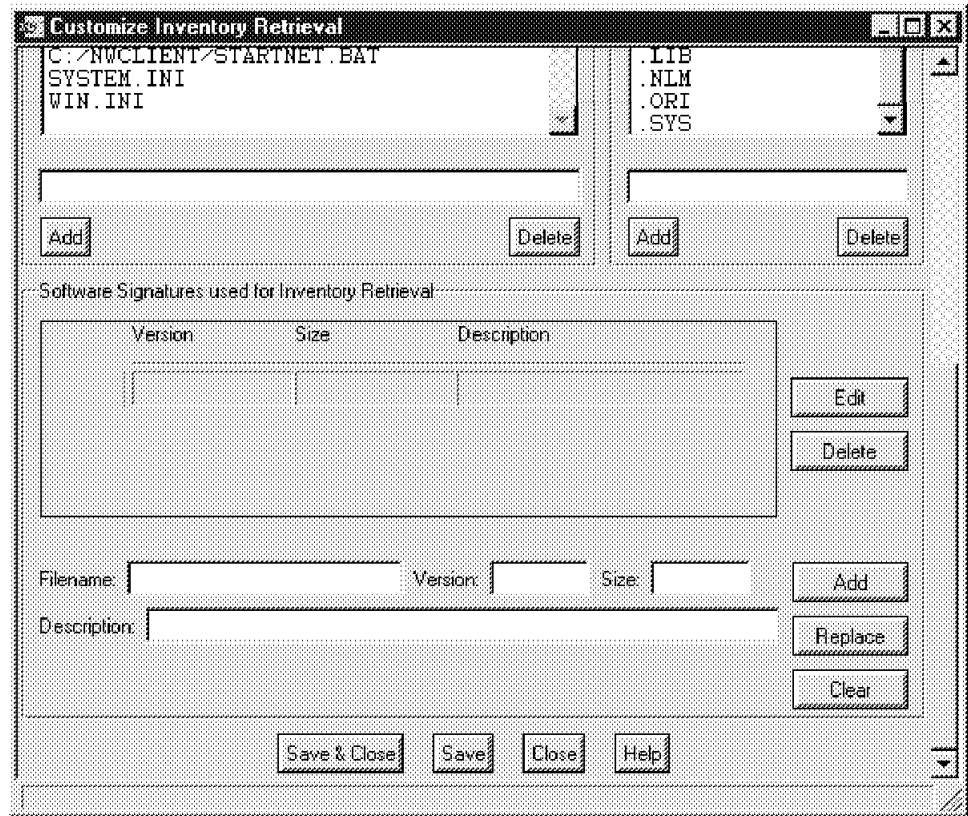


Figure 40. Customization Window (Part 2)

Click on **Save and Close**. This will return you to the Inventory Profile window. We can now run the scan on all the selected targets in our TMR. Click on **Run and Close**.

This scan will run for some time (1-2 minutes in our environment). Although the scanning of the targets is done in parallel, the RIM host waits until all targets have completed or timed out before writing the information to the RDBMS server. This write is done as a serial operation.

Once the scan is complete, you are returned to the Profile Manager window.

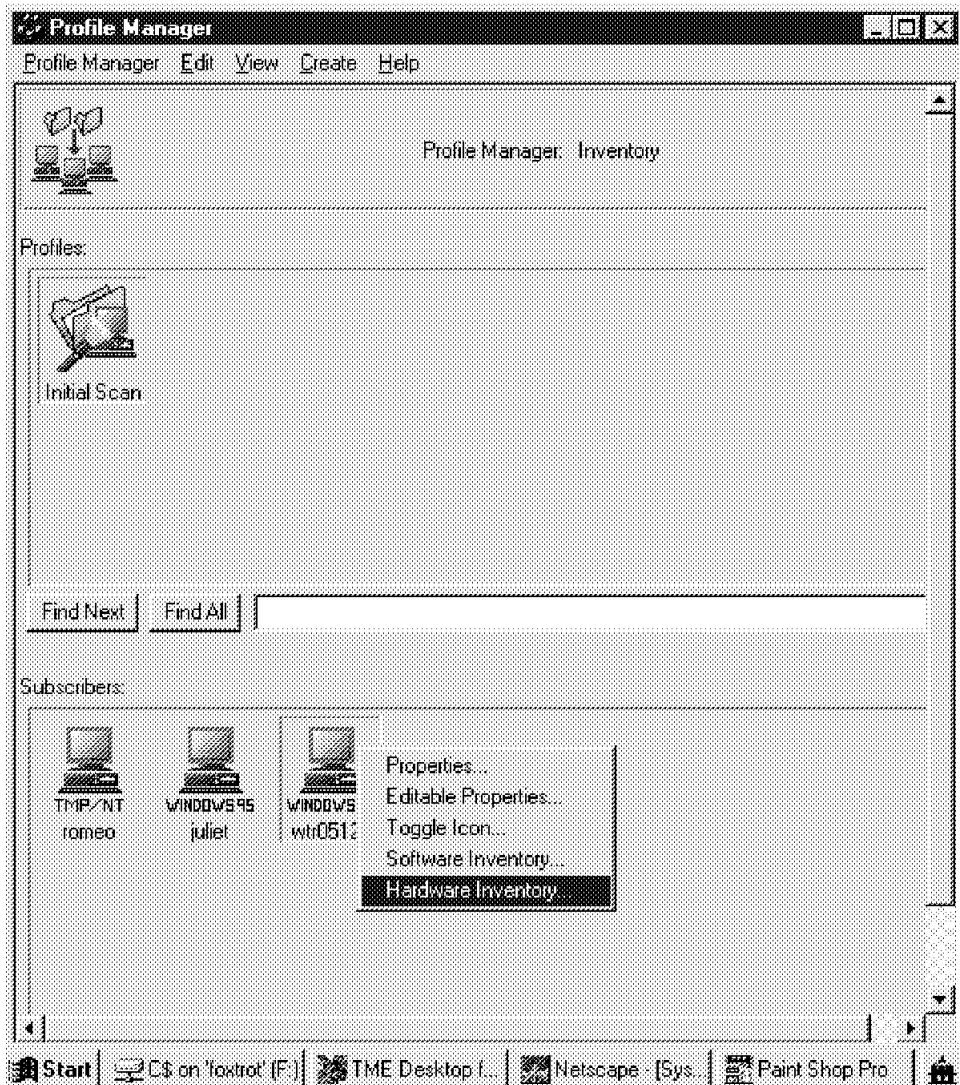


Figure 41. Profile Manager

To view the hardware information of one of the selected targets, highlight the target wtr05123 by clicking on it once with the left mouse button. Click once with the right mouse button to activate the menu and click on **Hardware Inventory...** The hardware inventory for the selected target, wtr05123, will be displayed.

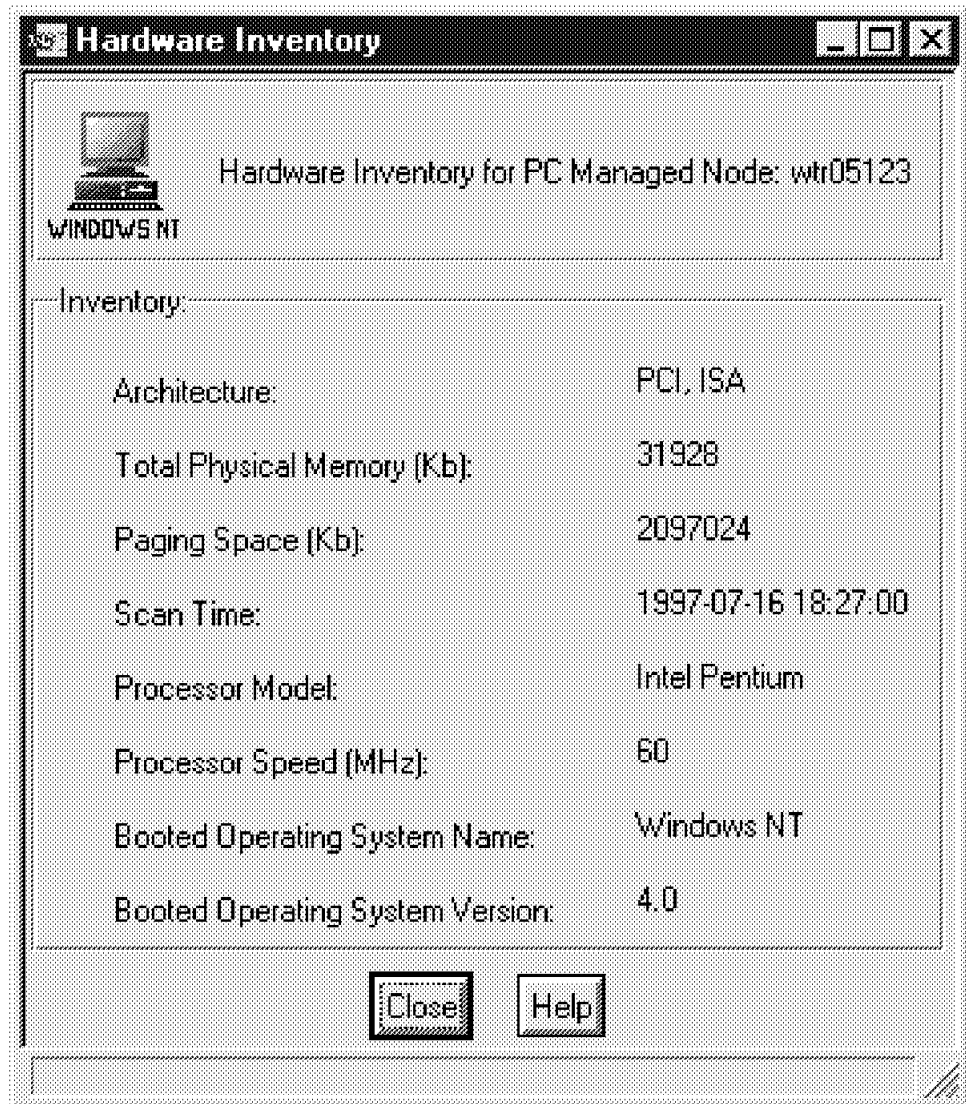


Figure 42. Hardware Inventory for wtr05123

Note

Once this information has been written to the database it can be viewed whether the target is active or inactive. TME 10 Inventory does not recheck or confirm the information with the target before displaying the output. Make sure scans are run at a regular interval in conjunction with your company's change control policies to keep the information up to date.

Chapter 3. Using TME 10 Inventory with Sybase on AIX

In this chapter we use TME 10 Inventory with Sybase for AIX as the database server.

Since we have shown how to use Sybase with TME 10 Inventory in Chapter 2, "Using TME 10 Inventory with Sybase on Windows NT" on page 13, this chapter serves mainly two purposes:

- Showing how to have the RDBMS server on a different platform than the TME 10 Inventory server
- Showing how to modify TME 10 Inventory to use a different RDBMS server

We combine these two goals by re-using the scenario from Chapter 2, "Using TME 10 Inventory with Sybase on Windows NT" on page 13, but moving the Sybase server from the Windows NT system victor to the AIX system rs600015.

The following figure illustrates the environment:

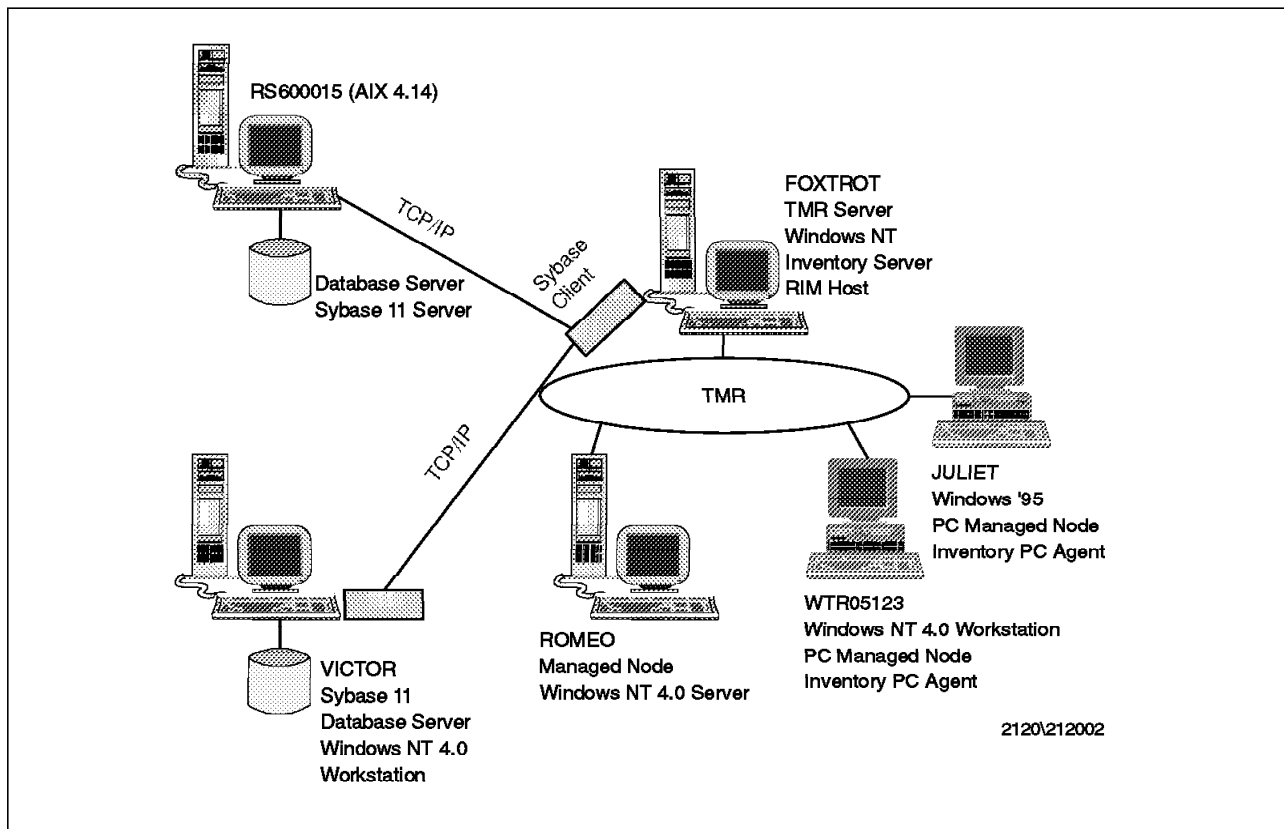


Figure 43. Using TME 10 Inventory with Sybase on AIX

3.1 Installing Sybase for AIX

We only briefly describe the installation steps necessary to install a Sybase RDBMS server on rs600015.

Insert the Sybase installation CD-ROM or have the installation images present in a file system on your machine. In our example, we use the CD-ROM. Assuming we have a CD-ROM file system with the name /cdrom we can mount the CD-ROM by typing:

```
mount /cdrom
```

To start the installation program we type:

```
cd /cdrom  
./sybsetup
```

Note

These steps need to be performed as the AIX root user. Before you install Sybase for AIX you must make sure that Asynchronous I/O on you AIX system is enabled. You can enable Asynchronous I/O typing smitty devices.

The following window will pop up:

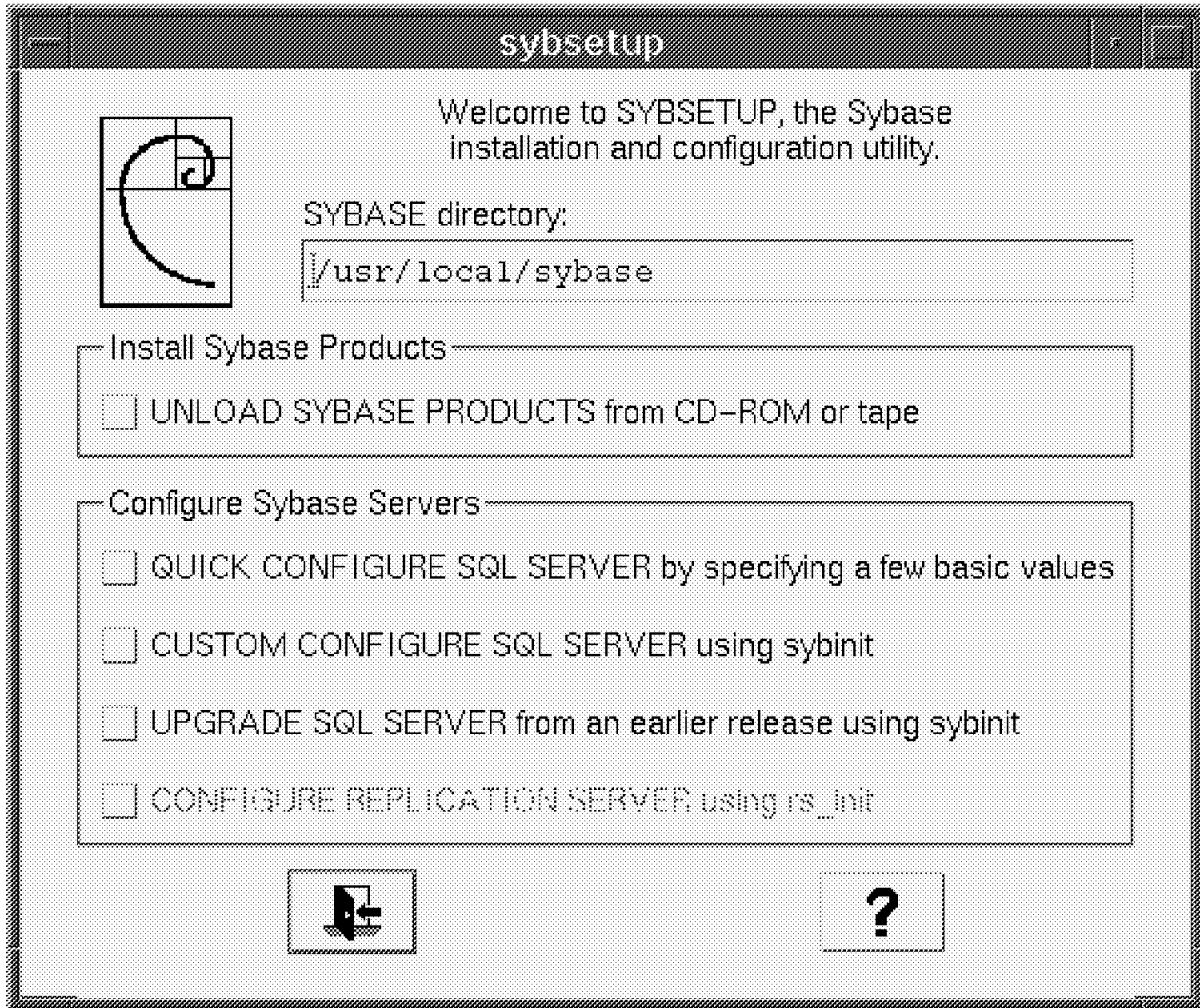


Figure 44. Sybase Installation Program

To install the Sybase RDBMS server, select the check box next to UNLOAD SYBASE PRODUCTS from CD-ROM or tape. Once you have done that you will be asked to enter the directory into which you want to install Sybase.

Leave the default (/usr/local/sybase) or enter a directory of your choice and then click on the check mark symbol.

Note

It is recommended that you create your own file system to install Sybase for AIX (in the example the file system would be /usr/local/sybase).

If you already have installed Sybase on your system, for example, when installing a newer version or reinstalling, you will get a warning saying that the SYBASE environment variable is set to the same directory where you want to install. If you intend to overwrite an existing installation, you can ignore this message. Otherwise, you can specify the SYBASE directory.

You will then be asked for the source of the installation images for Sybase. Usually, you can just leave the defaults here and click the check mark symbol.

Before the installation can begin you will be prompted to enter the license key for Sybase. This can usually be found on the sleeve of the CD-ROM. Enter the key and click on the check mark symbol.

This will bring up the selection list with the products you can install. In our case we just select the first entry in the list (the Sybase RDBMS server) and then click on the check mark symbol. This will start the installation. During the installation (which takes a few minutes), you will see the progress indicator window as shown below.

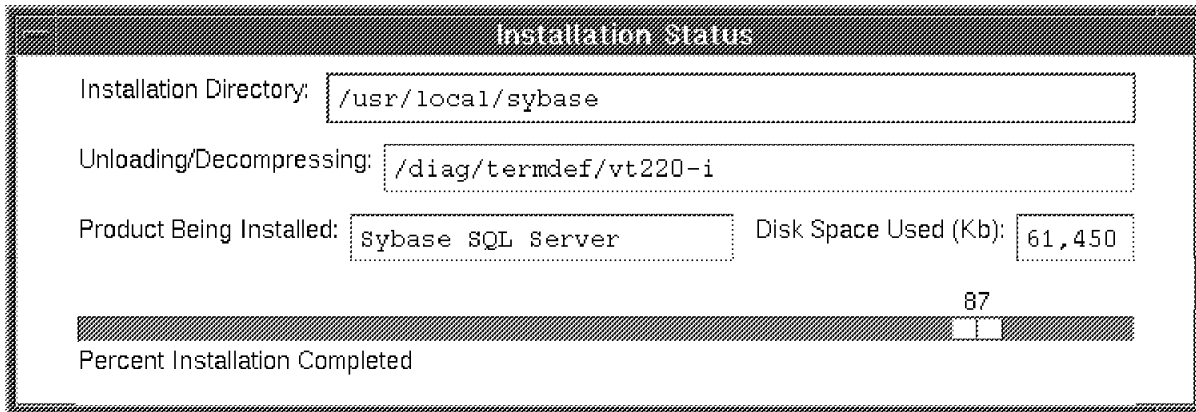


Figure 45. Installation Status Window

3.2 Configuring Sybase for AIX

Once the installation is complete, go back to the installation program home window as shown in Figure 44 on page 49 and this time select the check box next to QUICK CONFIGURE SQL SERVER by specifying a few basic values.

This will pop up the following window:

Quick Configuration

SQL Server Name: SYBASE

TCP Port Number for SQL Server: 4000

SQL Server Error Log: /usr/local/sybase/install/SYBASE.err

Master Device:

Path: /usr/local/sybase/master.dat Size (MB): 100

Sybsemprocs Device:

Path: /usr/local/sybase/sybprocs.dat Size (MB): 21

Backup Server Name: SYB_BACKUP

TCP Port Number for Backup Server: 8001

Backup Server Error Log: /usr/local/sybase/install/SYBASE_BACKUP.err

✓ 🏠 ⬇️ ?

Figure 46. Quick Configuration Window

In Figure 46, you specify the basic configuration for the Sybase RDBMS server. The field SQL Server Name specifies the name of the database server. You can either keep the default (SYBASE) or specify a name of your choice.

In Chapter 2, “Using TME 10 Inventory with Sybase on Windows NT” on page 13, the SQL server name was VICTOR.

In any case, note the name as you will need it later when configuring TME 10 Inventory.

The TCP Port Number for SQL Server field specifies the Sybase port. Again, you can leave the default 4000 or specify another value.

Note

If you look at Chapter 2, “Using TME 10 Inventory with Sybase on Windows NT” on page 13, you will see that the port on the Windows NT version of Sybase defaults to 5000, whereas on the AIX server the default port is 4000.

The most important field for our purposes is the size of the Master Device file. Since we are installing Sybase for AIX on a system where TME 10 Enterprise Console *and* TME 10 Inventory will be storing information in the same RDBMS, we change the value of that field from 21 to 100. When finished we click the check mark symbol. The following window will pop up:

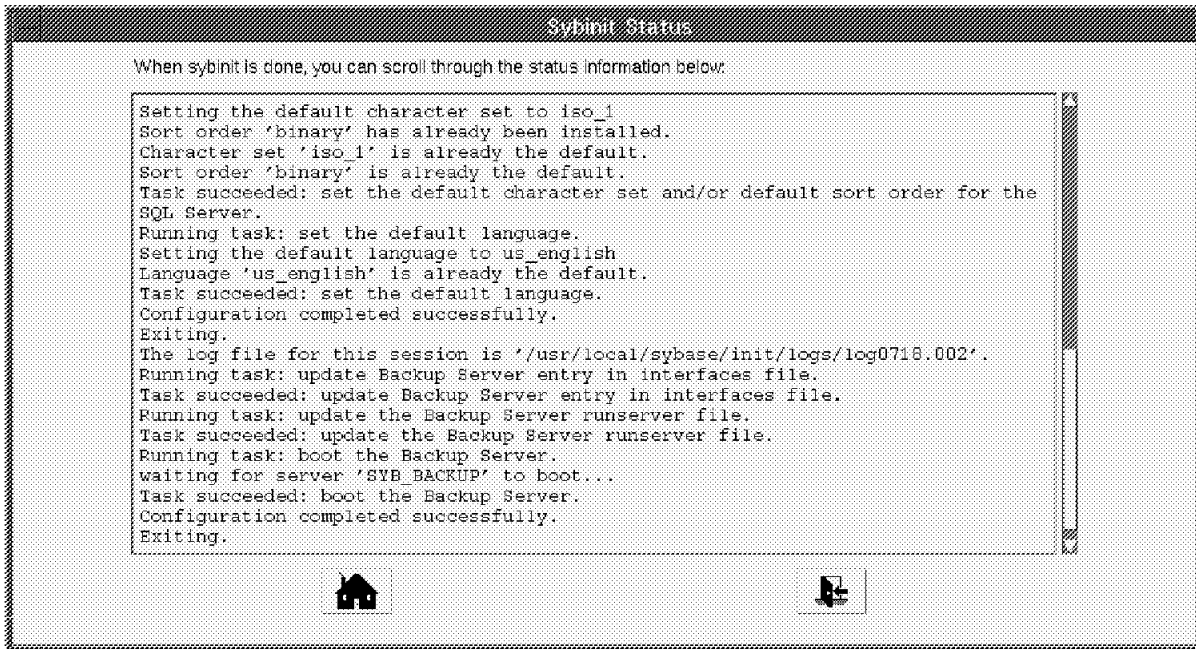


Figure 47. Sybinit Status Window

After the configuration has finished, you can exit the installation program and Sybase will be ready for use.

The `/usr/local/sybase/interfaces` file on rs600015 has the following content after the installation:

```

## SYB_BACKUP on rs600015
##      Services:
##          query  tcp    (8001)
##          master tcp    (8001)

SYB_BACKUP 0 0
          query tcp ether rs600015 8001
          master tcp ether rs600015 8001

## SYBASE on rs600015
##      Services:
##          query  tcp    (4000)
##          master tcp    (4000)

SYBASE 0 0
          query tcp ether rs600015 4000
          master tcp ether rs600015 4000

```

Figure 48. `/usr/local/sybase/interfaces` File

The relevant fields describe a query and master service for both, the server and the backup server. The interfaces file serves the same purpose on Sybase for AIX as the `SQL.INI` file does for Sybase for Windows NT as described in Chapter 2, "Using TME 10 Inventory with Sybase on Windows NT" on page 13.

3.3 Configuring Sybase for AIX for Use with TME 10 Inventory

As we are using the environment already set up in Chapter 2, "Using TME 10 Inventory with Sybase on Windows NT" on page 13 we do not need to install either TME 10 Inventory or a Sybase client.

From a Sybase perspective the only thing we need to do is to point the Sybase client on foxtrot to the new server on rs600015 and create the TME 10 Inventory database on the RDBMS server.

Note

You should notice that by using a new RDBMS server all information stored by TME 10 Inventory in the old RDBMS system will no longer be accessible unless you switch back. Since we have to create the Sybase tables on rs600015, the TME 10 Inventory database will be empty.

To do so, we use the SQLEDT tool as we did in Chapter 2, "Using TME 10 Inventory with Sybase on Windows NT" on page 13.

First, we define a connection to SYBASE through the SQLEDT utility.

Click on **Start, Programs, Sybase SQL Server Professional, SQLEDT** to start the SQLEDT window.

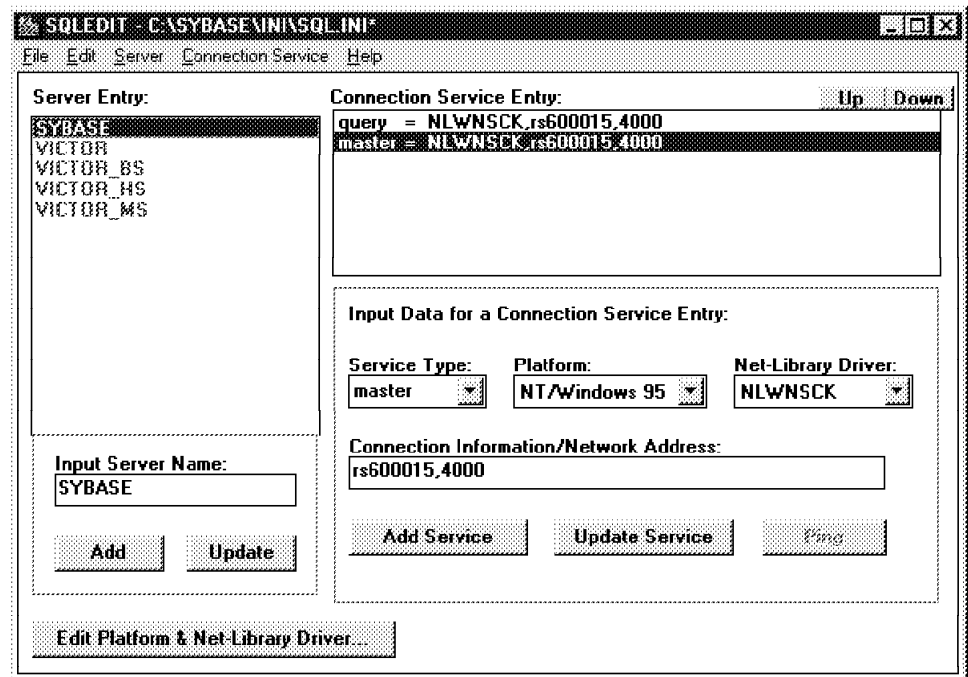


Figure 49. SQLEDT Window

The above window displays the contents of the SQL.INI file which we modify to contain the connection information needed to connect to the SYBASE RDBMS server.

On the left side we add the RDBMS server process, SYBASE, which is the RDBMS server name.

In the Input Server Name field enter SYBASE and click on **Add**. The SYBASE entry will now appear under the Server Entry field. Highlight **SYBASE** to edit the Connection Service Entry information.

In the case of SYBASE, the TCP/IP connection is described in the following way: the connection to the SYBASE RDBMS server is made using the TCP/IP hostname rs600015 and port 4000. We use a WINSOCK connection (NLWNSCK) with rs600015 as the TCP/IP hostname and 4000 as the port for SYBASE.

We add a Connection Service Entry for both query and master on the right side of the window. When you have finished entering all definitions, select **File** from the menu bar and **Save** from the pull-down menu. Then close the SQLEDIT tool.

We now open a connection to SYBASE via WISQL32 on foxtrot to run the necessary scripts to create the TME 10 Inventory database, tivoli userid and database schema. Run **wisql32** from a Windows command prompt.

Click on **Connect, Open Connection**. The following window will appear:

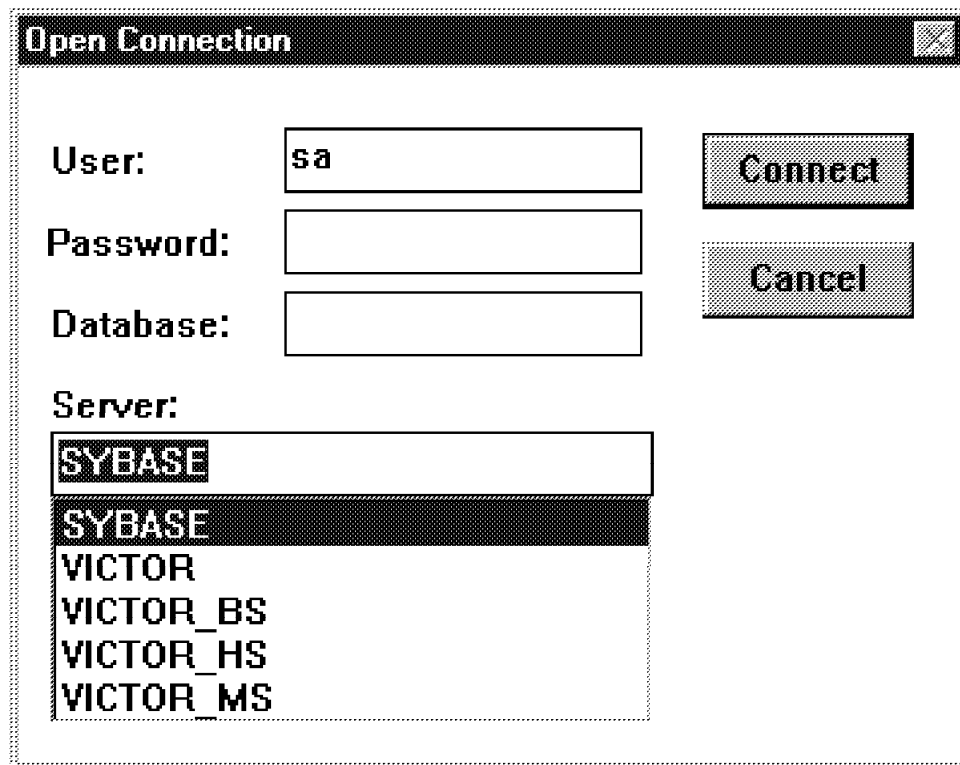


Figure 50. Open Connection Window

Enter **sa** as the userid. Highlight **SYBASE** and click on **Connect**.

Notice that the window title bar has changed to indicate the connection, from (Untitled) on (No Server) to (Untitled) on SYBASE/master/sa.

Select **File** from the menu bar and then **Open** from the pull-down menu. Change to the source directory of the SQL scripts that are supplied with TME 10 Framework 3.1 and look for:

- tivoli_syb_admin.sql
- tivoli_syb_schema.sql

Note

If you have problems finding the path to the files, type the following from the drive letter where TME 10 Framework is installed:

```
dir /s *.sql
```

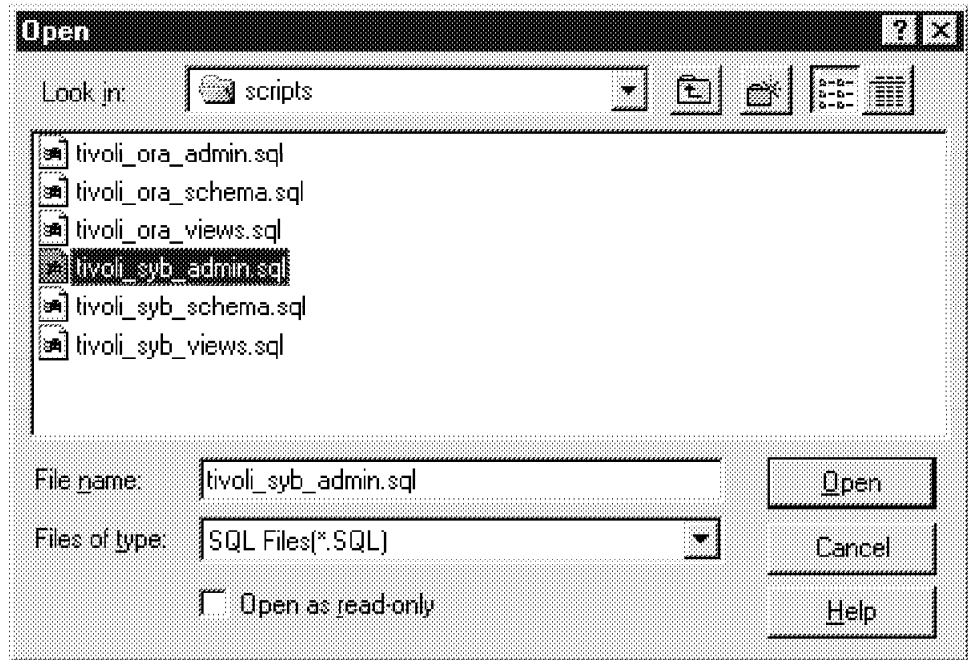


Figure 51. Open Scripts Window

Highlight **tivoli_syb_admin.sql** and click on **Open**.

Click on **Query, Execute All** to run the script.

This script creates the user, tivoli, and the database, inventory, in the RDBMS on rs600015. The password for this new user, tivoli, is also tivoli.

Now that the tivoli_syb_admin.sql is complete and the tivoli user ID and inventory database have been created on rs600015, we need to create the database schema.

Note

You must log off the sa user and log on using the tivoli user ID before running the tivoli_syb_schema.sql script. This needs to be done because the tivoli user ID owns the database and the TME 10 Inventory server communicates with the database through this user ID. If you run the tivoli_syb_schema.sql script under the sa user ID, the TME 10 Inventory application will not be able to communicate with this database.

Follow the same process described above using the tivoli user ID and tivoli password for the tivoli_syb_schema.sql script.

Note

A new script is included with TME 10 Inventory that needs to be run. If you are upgrading from TME 10 Inventory 3.0 to TME 10 Inventory 3.1, then you must run the syb_sp1_schema.sql script *BEFORE* upgrading TME 10 Inventory. If this is a first time installation, you need to run the syb_sp1_schema.sql script *AFTER* the tivoli_syb_schema.sql script.

This SQL script must be run with the tivoli user ID because the script updates the database tables created by the tivoli_syb_schema.sql script.

Now that all the scripts have been run, we can do another test of SQL connectivity between the RIM host foxtrot and the database server on rs600015. This will be done using the WISQL32 tool on foxtrot again.

Open the WISQL32 interface if its not already open by typing **Wisql32** at a Windows command prompt on foxtrot.

Repeat the steps described before to connect to the RDBMS server, but this time specify tivoli as the User and Password and inventory as the Database. Figure 52 will appear after the connection is completed successfully.

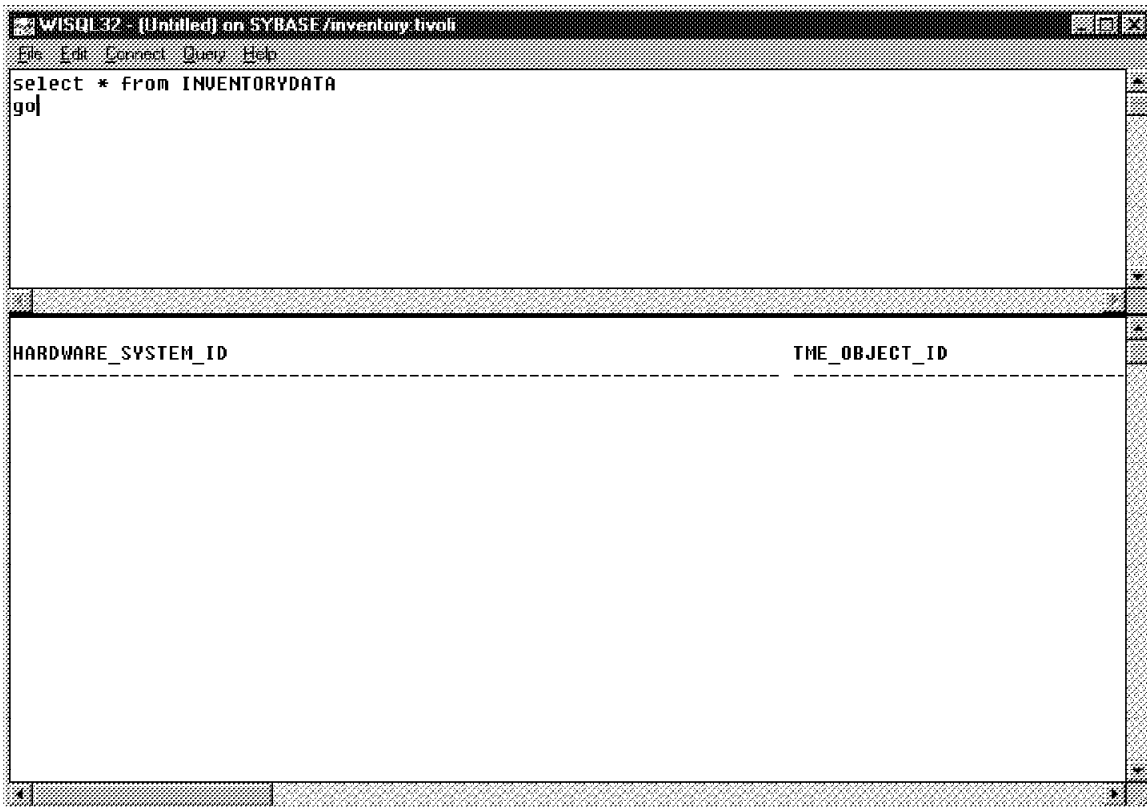


Figure 52. Open Connection to Inventory Using Tivoli User ID

In the query section enter:
`select * from INVENTORYDATA
go`

Select **Query, Execute all** from the pull-down menu. If the table header is displayed, then Sybase, TME 10 Inventory, and TCP/IP are configured correctly.

3.4 Configuring TME 10 Inventory to Work with Sybase for AIX

In this section we change the database information for our RIM object on foxtrot from VICTOR(NT) to SYBASE(AIX) without reinstalling TME 10 Inventory but simply using the *wsetrim* command.

To see the current database information of the RIM object named inventory enter the following from a tivoli command prompt:

```
wgetrim inventory
```

The output is as follows:

```
RIM Host      : foxtrot
RDBMS User   : tivoli
RDBMS Vendor : Sybase
Database ID  : inventory
Database Home : c:\sybase
Server ID    : VICTOR
```

To change the current database information for the RIM object type the following from a tivoli command prompt:

```
wsetrim -u tivoli -d inventory -H c:\sybase -s SYBASE inventory
```

where:

- u** Specifies the name of the database user ID that the RIM object will use.
- d** Specifies the name of the database to which the RIM object will connect.
- H** Specifies the path to the database home directory.

Note

We tried changing this to /usr/local/sybase which is the path of the database on the rs600015 but after it was changed we could no longer connect to the database via TME 10 Inventory. This path, where the SQL commands can be found, is for the SQL client, not the SQL server.

- s** Specifies the server ID for the database.

Note

This is the database name and not the machine hostname

inventory Specifies the label of the RIM object to be created.

Once the command is complete the RIM database information should have changed. To check that the RIM object has the correct database information enter the following command from a tivoli command line:

```
wgetrim inventory
```

The output is as follows:

```
RIM Host      : foxtrot
RDBMS User   : tivoli
RDBMS Vendor : Sybase
Database ID  : inventory
Database Home : c:\sybase
Server ID    : SYBASE
```

Notice that the Server ID field has changed to reflect the new value.

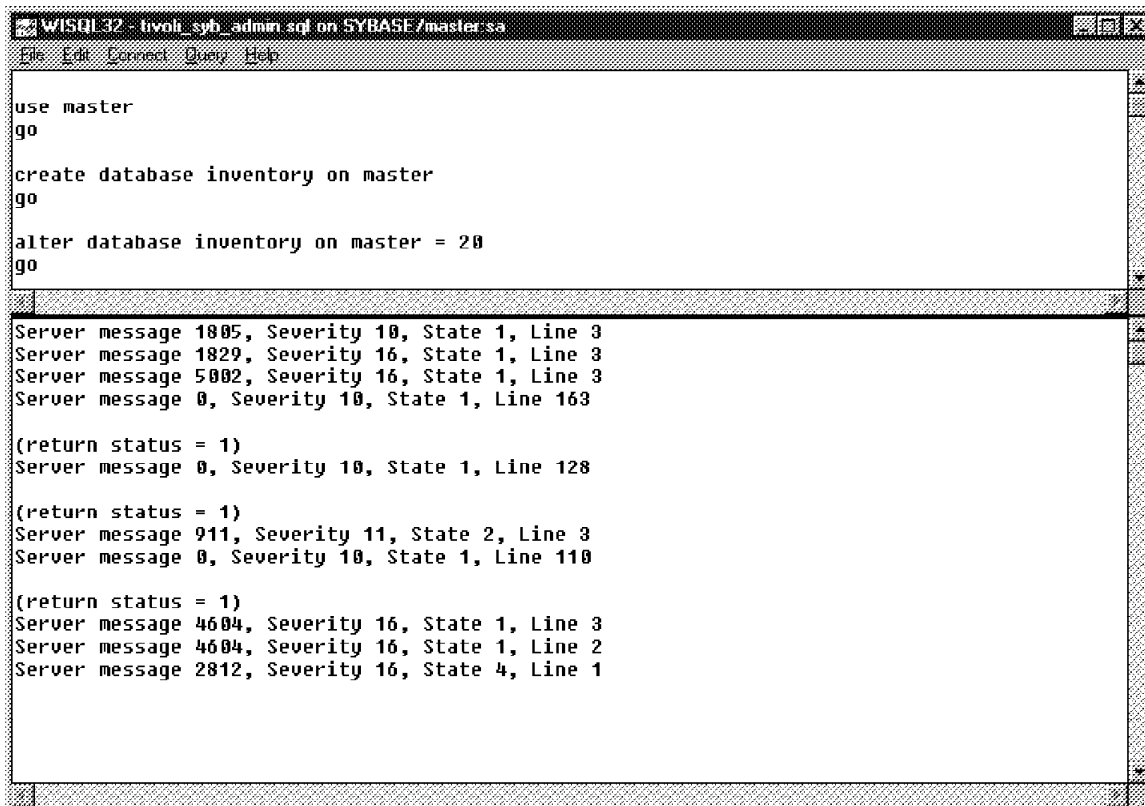
Note

Now that the RIM database information has changed we must remember that the inventory database on rs600015 is empty. A scan initiated by TME 10 Inventory should be run to populate the database and check the RIM object's ability to read and write information to/from the database.

Initiate a scan and check that the TME 10 Desktop information can be accessed via the TME 10 Desktop for Windows and via the WISQL32 tool as described in Chapter 2, "Using TME 10 Inventory with Sybase on Windows NT" on page 13.

3.5 Database Creation Problem

We had the following problem when trying to create the inventory database on rs600015:



```
WISQL32 - tivoli_syb_admin.sql on SYBASE/master sa
File Edit Command Query Help

use master
go

create database inventory on master
go

alter database inventory on master = 20
go

Server message 1805, Severity 10, State 1, Line 3
Server message 1829, Severity 16, State 1, Line 3
Server message 5002, Severity 16, State 1, Line 3
Server message 0, Severity 10, State 1, Line 163

(return status = 1)
Server message 0, Severity 10, State 1, Line 128

(return status = 1)
Server message 911, Severity 11, State 2, Line 3
Server message 0, Severity 10, State 1, Line 110

(return status = 1)
Server message 4604, Severity 16, State 1, Line 3
Server message 4604, Severity 16, State 1, Line 2
Server message 2812, Severity 16, State 4, Line 1
```

Figure 53. WISQL32 Window Connected to SYBASE

We recognized this as a problem because the SQL script run time was very short and there were 3 return code one's indicating 3 failures. The 3 steps performed by the *tivoli_syb_admin.sql* scripts are:

- Creating the inventory database
- Adjusting the size to specified size
- Creating the tivoli userid

We did not have the Sybase error manual to debug the problem so we used SQL Central to assist us in debugging the problem.

Open SQL Central by clicking on **Start, Programs, Sybase SQL Server Professional and SQL Central**. The following window will appear:

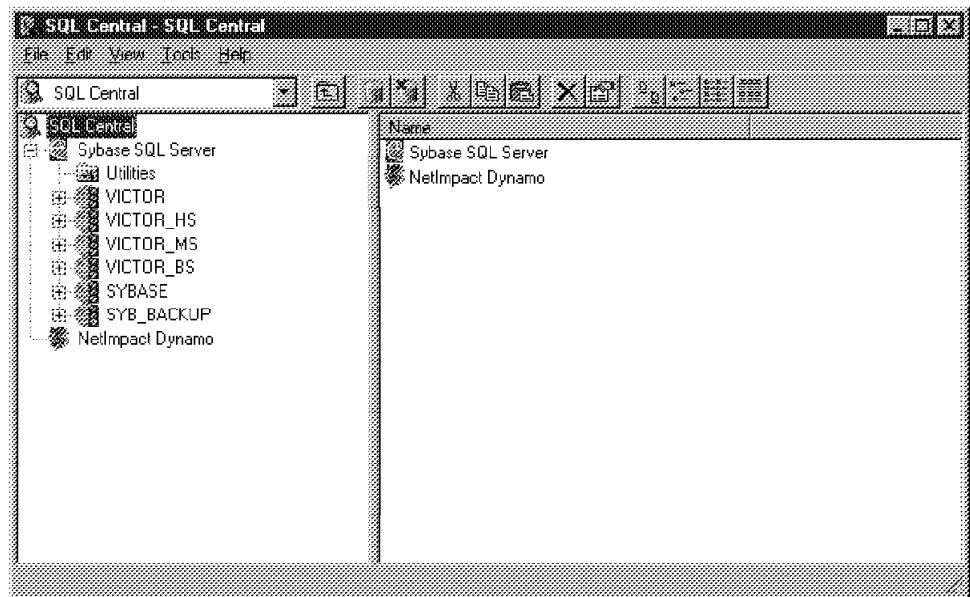


Figure 54. SQL Central Window

Click once with the left mouse button on **SYBASE**. Enter the sa userid at the Connect to Sybase SQL Server window.

Open SYBASE by double-clicking on it with the left mouse button.

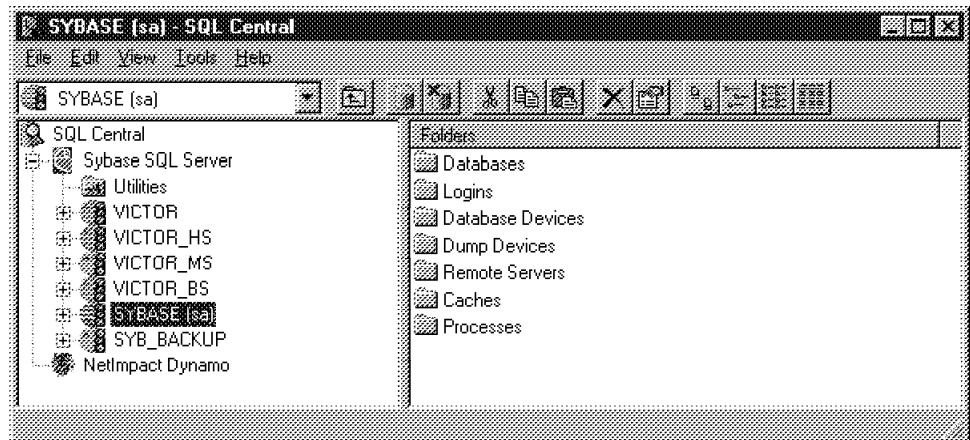


Figure 55. SQL Central Window

Double-click on **Database Devices**. The following window will appear:

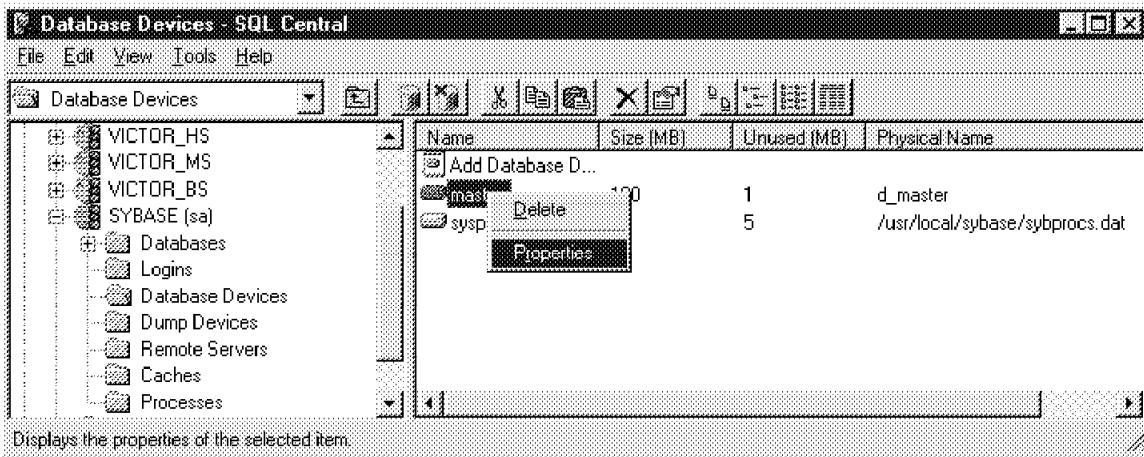


Figure 56. Database Devices Window

Click once with the right mouse button on **master** to activate the pull-down menu.

Click on **Properties**. The master properties window will be displayed.

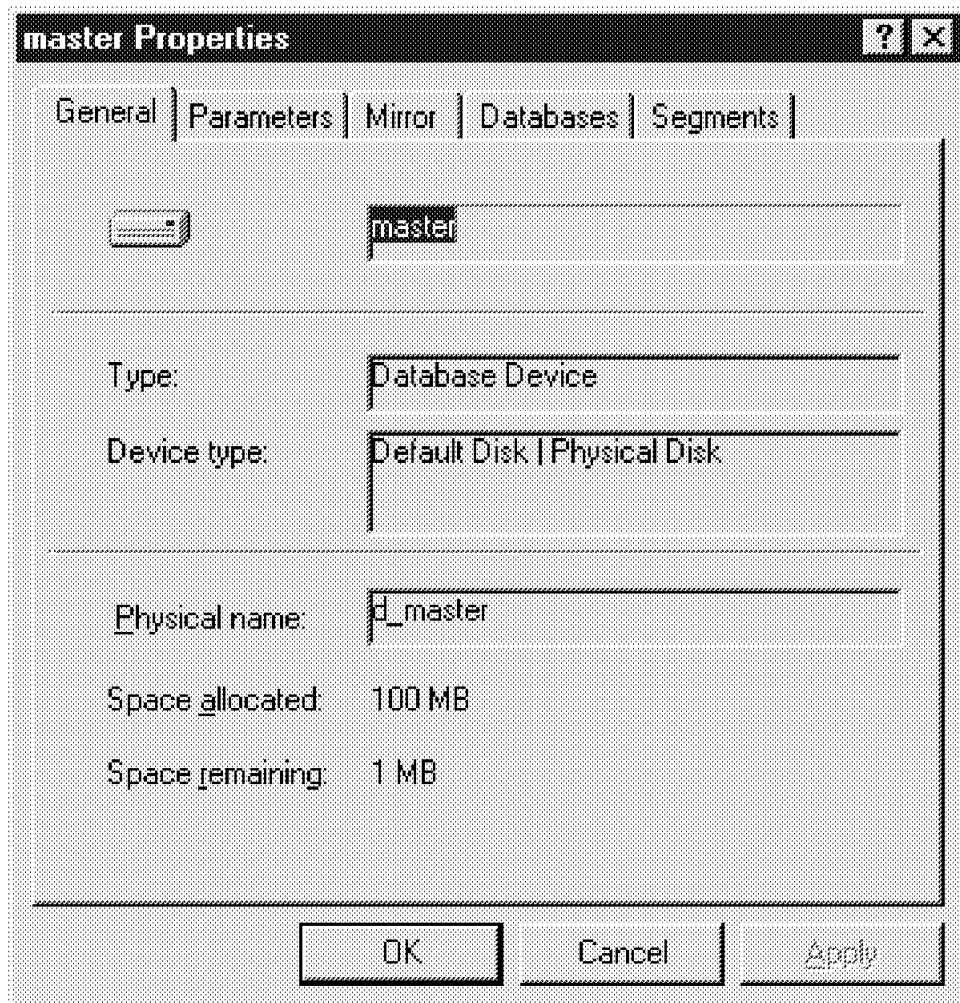


Figure 57. Master Properties Window

Note

Note that the size allocated to master is 100 MB.

Click on the **Databases** tag. The following window will be displayed:

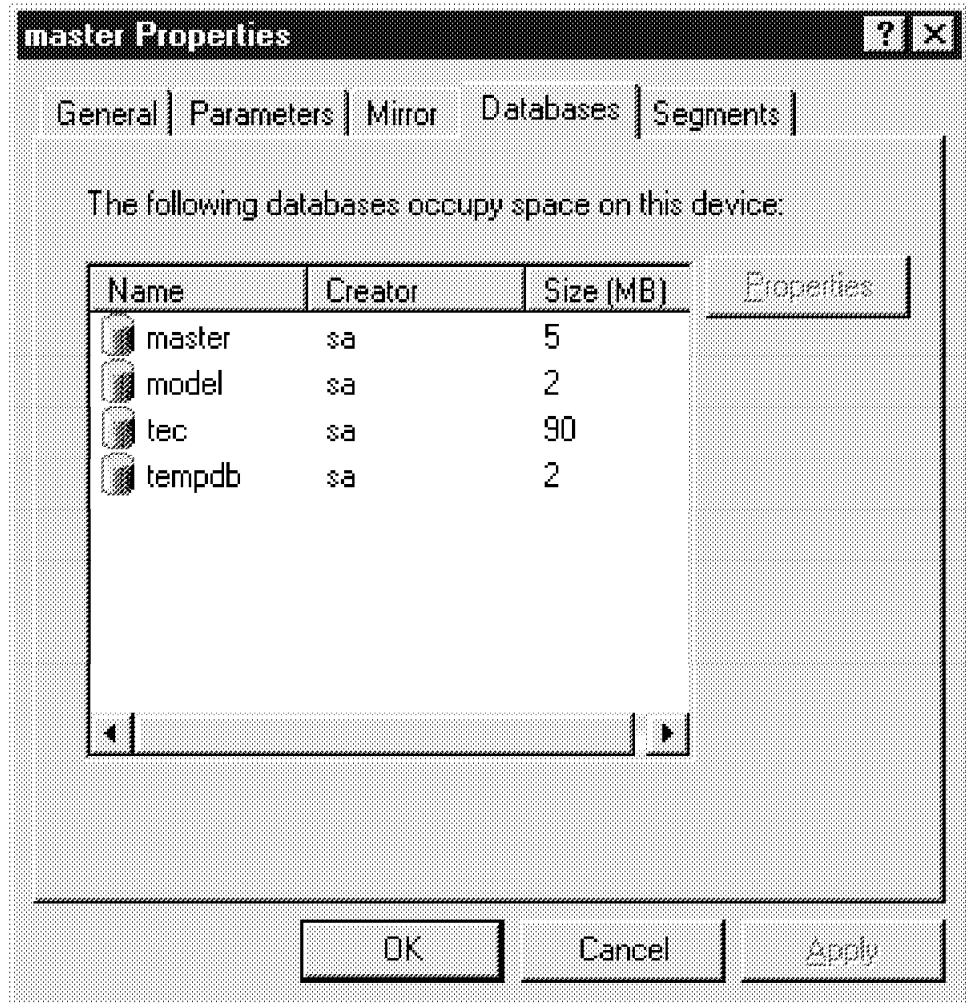


Figure 58. Master Properties Window

You notice that the total size of all the databases is 99 MB. As we only have 100 MB defined and we are trying to define an inventory database of 20 MB (default value in `tivoli_syb_admin.sql`) we know that the SQL script failed because there is not enough space in the master device.

To resolve this problem we chose to delete and re-create the `tec` database with a smaller size.

Note

Deleting the `tec` database deletes all the alerts and messages in the database. When doing an installation like this remember to plan properly when deciding on a size for the master device before running the `tivoli_syb_admin.sql` script.

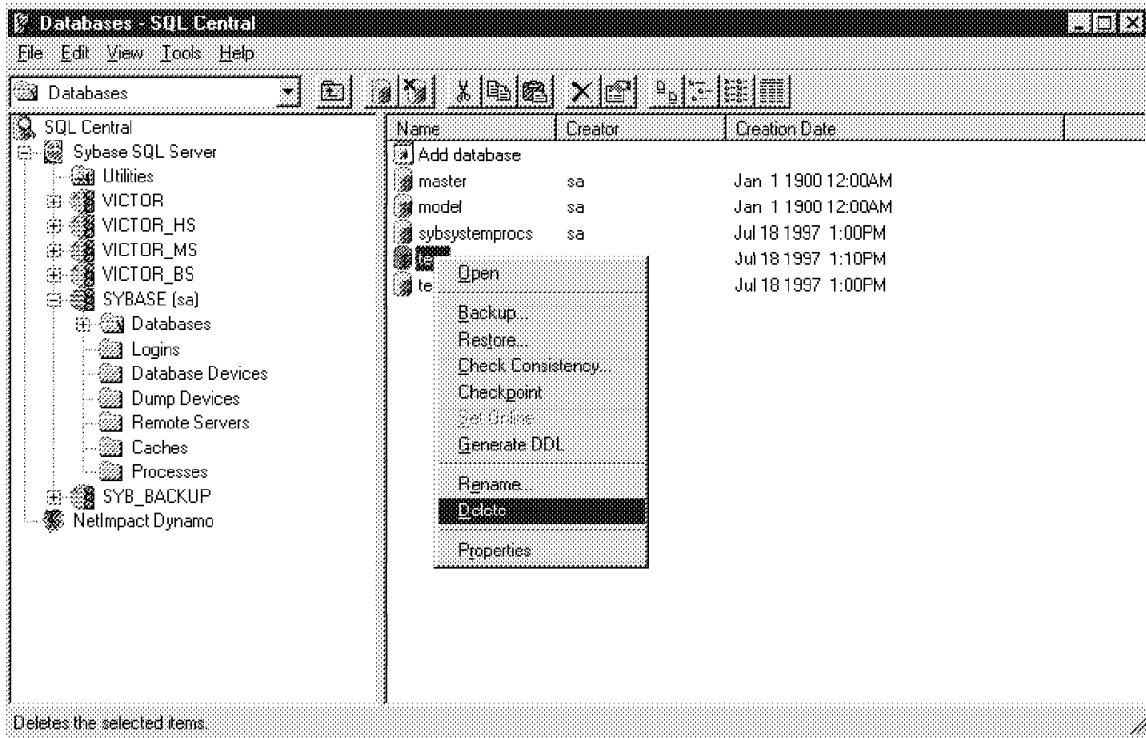


Figure 59. Databases - SQL Central Window

To delete the tec database highlight it with the left mouse button. Click once with the right mouse button to activate the drop down menu and click on **Delete**.

The tec database has now been deleted.

We now re-create the tec database with a size of 50 MB. To create the tec database enter the following command from the AIX command line:

```
cd /usr/local/Tivoli/bin/aix4-r1/TME/TEC/sql
```

Ensure that you are in the above directory before executing the next command:

```
cr_tec_db.sh
```

You will be prompted to enter the configuration details including the size during this install program.

Once the installation is complete you can rerun the *tivoli_syb_admin.sql* script to create the inventory database as described earlier in this chapter.

Chapter 4. Using TME 10 Inventory with Oracle for AIX

In this chapter, we discuss TME 10 Inventory in combination with Oracle for AIX.

We give an overview of the necessary installation and configuration steps in this environment and we show three different examples of how to use TME 10 Inventory, in combination with Oracle for AIX.

4.1 Overview and Objective

The objective of this chapter is to successfully install Oracle for AIX on an AIX machine and then use TME 10 Inventory with Oracle for AIX. We provide three scenarios, showing how to configure an AIX environment in different ways.

In our scenarios, the hardware equipment we used is:

- rs600020** RS/6000 PowerPC 601 (SMP), 128 MB memory, 4.5 GB disk
- rs600024** RS/6000 PowerPC 604 (133 MHz), 128 MB memory, 2 GB disk
- rs600026** RS/6000 PowerPC 604e (166 MHz), 192 MB memory, 2 GB disk

All of these systems are running AIX Version 4.2 operating system and using TCP/IP for the network protocol. The application software used for these scenarios is:

- TME 10 Framework Version 3.1 Revision C
- TME 10 Framework Version 3.1.1 Service Pack 1
- TME 10 Inventory Version 3.1 Revision A
- Oracle Version 7.1

The first managed node installed in the TMR maintains the database that contains information about the TMR and is called the TMR server. TME 10 Inventory refers to two components:

- The TMR RDBMS access host or RIM host, sometimes called the inventory server, is the component that provides the collection point for inventory data for the TMR and the interface to the RDBMS server.

Note

The RIM component, originally a component of the TME 10 Inventory product, is now a component of the TME 10 Framework (Version 3.1 and later).

- The RDBMS server, sometimes called the database server, is the component of the database product that stores the inventory data and interfaces with the TMR RIM host.

4.1.1 Scenario 1: RDBMS Server and RIM Host on the TMR Server

The first scenario has the RDBMS server and RIM host on the TMR server. We installed TME 10 Inventory 3.1 and Oracle for AIX Version 7.1 on rs600020. We installed TME 10 Inventory 3.1 on the other two machines (rs600024 and rs600026), which are the managed nodes. So rs600020 is the TMR server, TME RDBMS access host or RIM host (TME 10 Inventory server) and RDBMS server (Server ID) (see Figure 60).

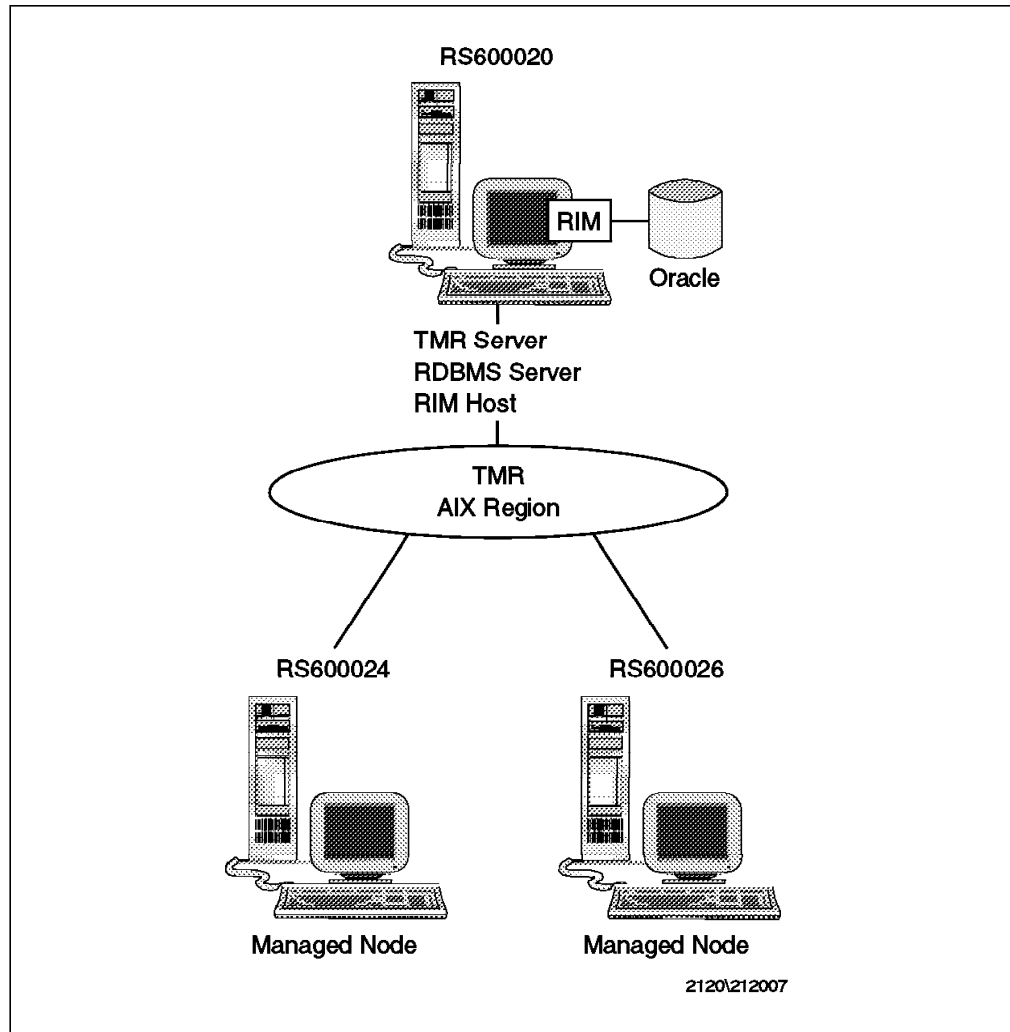


Figure 60. Scenario 1 - RDBMS Server and RIM Host on the TMR Server

4.1.2 Scenario 2: RDBMS Server Located Outside the TMR

The second scenario has the RDBMS server located outside the TMR. We installed TME 10 Inventory 3.1 on the rs600020 machine, and we installed Oracle for AIX on rs600024. On the other machine, rs600026, which is a managed node, we installed TME 10 Inventory 3.1. So rs600020 is the TMR server and the TME RDBMS access host or RIM host (inventory server), rs600024 is the RDBMS server and is outside of the boundary of the TMR region, and rs600026 is the managed node in the TMR region (see Figure 61 on page 65). With this scenario, we show how to have a remote (outside the TMR) RDBMS server for a TME region.

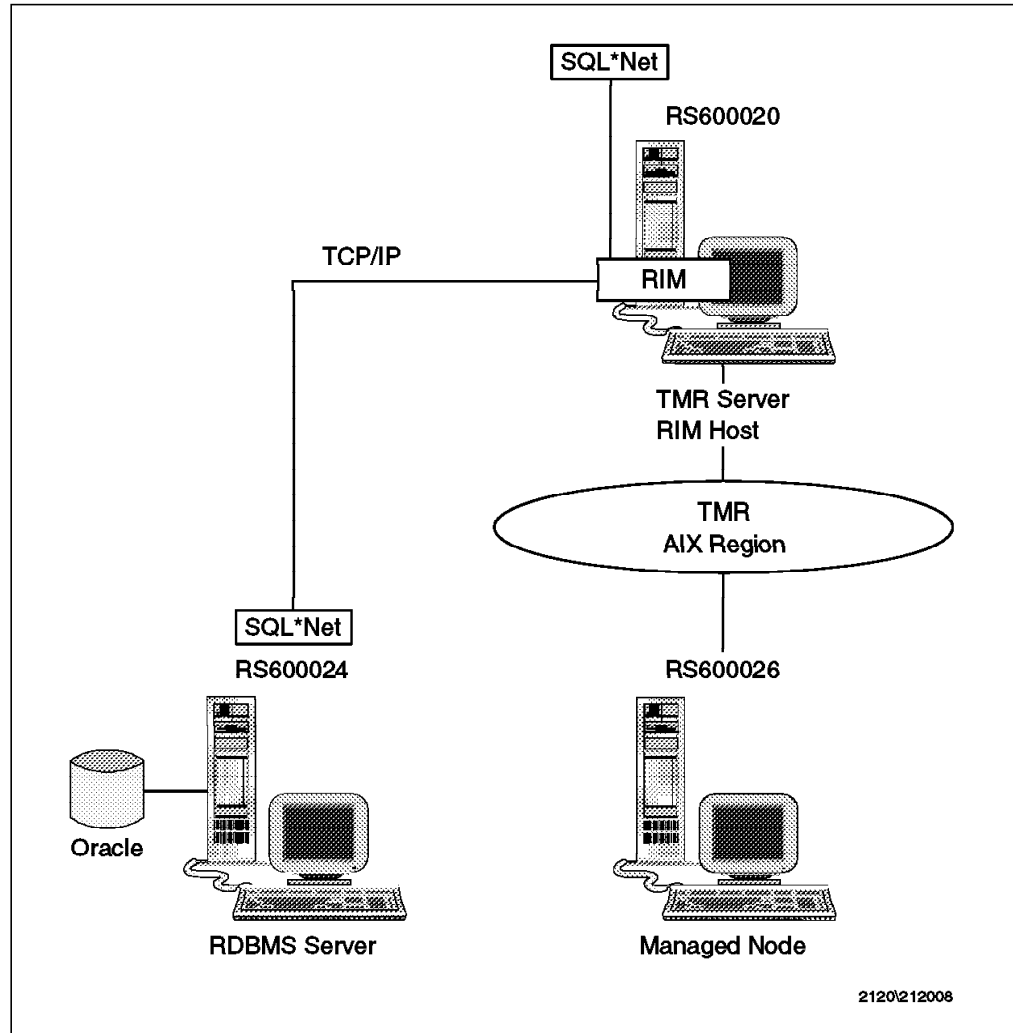


Figure 61. Scenario 2 - RDBMS Server Located Outside the TMR

4.1.3 Scenario 3: RIM Host Not on the TME Server

The third scenario has the TME RDBMS access host on a managed node that is not the TMR server. In this scenario, rs600020 is the TMR server, rs600024 is the RDBMS server and rs600026 is the TME RDBMS access host or RIM host (inventory server) (see Figure 62 on page 66).

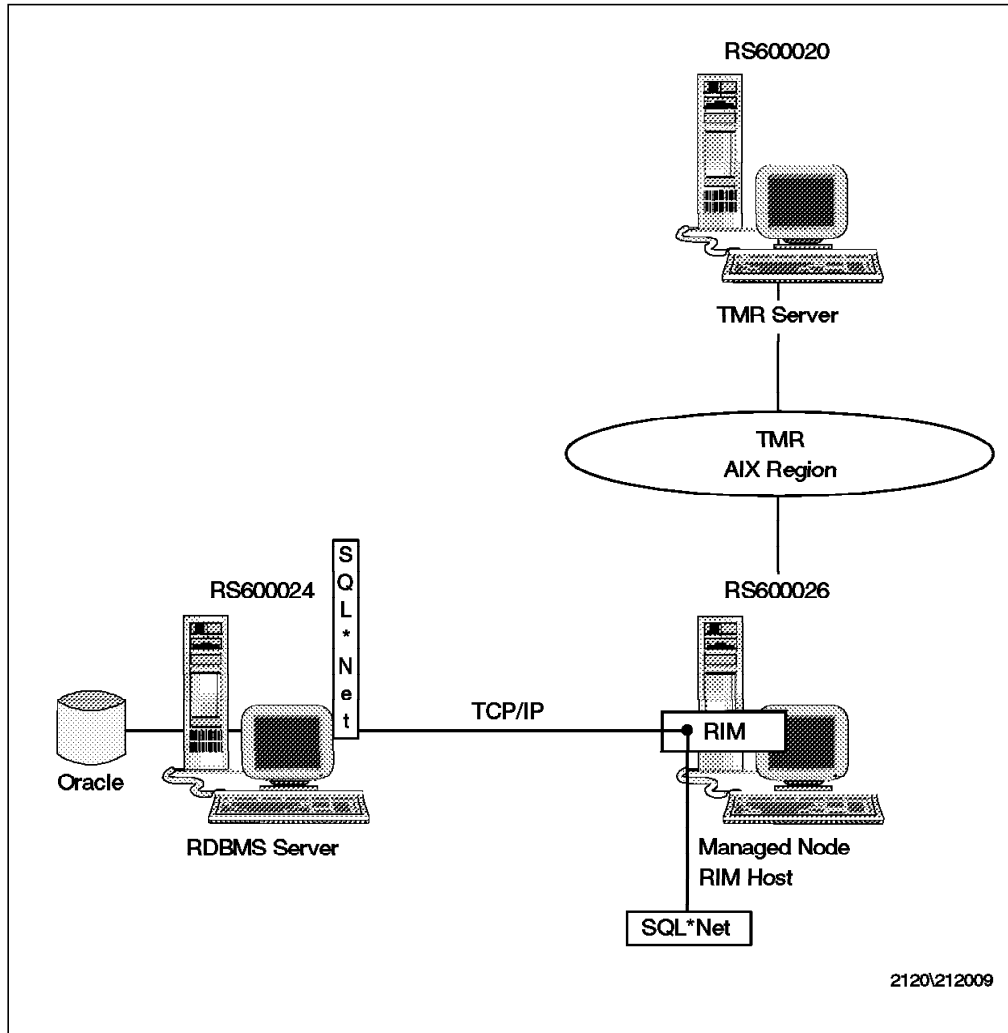


Figure 62. Scenario 3 - RIM Host Not on the TME Server

4.2 Installing the TME Management Framework (TMF)

We have already shown how to install Tivoli Management Framework (TMF) in a UNIX environment in *TME 10 Cookbook for AIX: Systems Management and Network Applications*, SG24-4867.

Note

The Tivoli terminology has changed. The component of the Tivoli Management Environment that was called the Tivoli Management Platform (TMP) in *TME 10 Cookbook for AIX: Systems Management and Network Applications*, SG24-4867, is now called the Tivoli Management Framework (TMF).

Here we show how to do the TME 10 Framework install using Tivoli commands. You need to use the command line interface (CLI) when you do not have access to a GUI, for example, because you are connected over a modem line. Another very useful reason to use the command line is when you want to group commands together inside a shell script.

All Tivoli end-user commands have a common syntax, so that it is easier to think of the action you want to perform. They all have the form `w+action+object`. For example, to create a new profile, use the `wcrtprf` (`w + crt` for create + `prf` for profile) command.

Note

In this chapter we only describe the command flags that we used. For a full explanation of the usage of a command, refer to the *TME 10 Framework Reference Manual Version 3.1*.

4.2.1 Installing TME on an AIX Server

To install TME 10 Framework 3.1 on an AIX machine (in this example, rs600020) from the command line follow these steps:

1. Log in as root.
2. Verify the installation prerequisites as shown in *TME 10 Cookbook for AIX: Systems Management and Network Applications*, SG24-4867.
3. Insert the CD-ROM labelled TME 10 Framework Version 3.1 Revision C, LK2T-6059-02, in the CD-ROM drive.
4. Mount the CD-ROM on rs600020 to install TME Framework binaries:

```
mkdir /cdrom
mount -rv cdrfs /dev/cd0 /cdrom
```

5. Create an installation directory and make it the current directory. Very little disk space is required for this directory. Run these commands on rs600020:

```
mkdir /tiv_inst
cd /tiv_inst
```

6. Start the pre-installation script on rs600020:

```
/cdrom/wpreinst.sh
```

7. Run the command on rs600020:

```
export DOGUI=no
```

You have to set this variable if you want to do the install using the command line.

8. Install TME 10 Framework by running the following command on rs600020:

```
wserver -y -c /cdrom -d LK=XXXXXXXX RN=aixregion CreatePaths=1
```

The output of this command should be:

```
Registering installation information...finished
```

The `wserver` command installs the initial TME 10 server for a Tivoli Management Region (TMR).

The flags mean:

Flag	Purpose
-y	Specifies that the installation should proceed without confirmation.

<i>Table 1 (Page 2 of 2). wserver Flags</i>	
Flag	Purpose
-c	Specifies the complete path to the CD-ROM image.
-d	Specifies that you want to keep the default installation values for all the parameters. If you want to change one of these values, you must set it specifically.
LK	Specifies the license access key that you receive from Tivoli.
RN	Specifies the policy region name specified when the TMR was created.
CreatePaths	Variable set to 1 specifies to create the specific directories, if they do not already exist.

Note

The -y flag is not documented in the manual pages or *TME 10 Framework Reference Manual Version 3.1*, but it worked for us.

9. Unmount the CD-ROM on rs600020:

```
umount /cdrom
```

10. Insert the CD-ROM labelled TME 10 Framework Version 3.1.1 Service Pack 1, LK2T-6108-00, in the CD-ROM drive. Mount the CD-ROM again.

11. Install the patch by running the following command on rs600020:

```
wpatch -c /cdrom -i tmf_31_s -y rs600020
```

The output of this command should be:

```
Finished patch installation
```

The flags mean:

<i>Table 2. wpatch Flags</i>	
Flag	Purpose
-c	Specifies the complete path to the CD-ROM image.
-i	Specifies the patch installation index file to install. A patch installation index file ends with the file extension .IND. You can enter the file name with or without the file extension.
-y	Specifies that the installation should proceed without confirmation.

The rs600020 specifies the managed node on which a TME 10 patch will be installed.

Note

The manual pages state that "You must specify the file name in all capital letters", but the patch file name must be specified exactly like the file name in the CD-ROM image.

4.2.2 Installing AIX Managed Nodes as TME Clients

Once you have installed the TMR server, you can install the managed nodes. We show how you can install the clients remotely, that is, by issuing the commands from the server you have installed previously. We installed the TME Framework 3.1 on the AIX machines rs600024 and rs600026.

Before installing TME Framework on the clients, make sure that your TMR server has trusted host permissions on every client. To give the server trusted host permission, add the TMR server name to the `/.rhosts` file on every client machine. If your TMR server does not have trusted host permissions, when attempting to install the clients you will receive a message similar to the following:

```
The following problems were encountered:  
Host rs600024:  
  Trusted host access from TMR server denied: check host name, /.rhosts  
  The type of this machine could not be classified as a supported  
  architecture.  
Host rs600026:  
  Trusted host access from TMR server denied: check host name, /.rhosts  
  The type of this machine could not be classified as a supported  
  architecture.
```

To install TME 10 Framework 3.1 on the clients, do the following steps on rs600020:

1. Log in as root.
2. Verify the installation prerequisites as shown in *TME 10 Cookbook for AIX: Systems Management and Network Applications*, SG24-4867.
3. Insert the CD-ROM labelled TME 10 Framework Version 3.1 Revision C, LK2T-61059-02, in the CD-ROM drive.
4. Mount the CD-ROM.
5. Run the following command on rs600020:

```
wclient -c /cdrom -d -y -p region_name CreatePaths=1 rs600024 rs600026
```

The `-p` flag specifies the name of the Policy Region defined previously in the installation of the TMR server. The other flags have the same meaning as the ones for the `wserver` command. At the end of the command you have to specify the list of clients you want to install (in this example, rs600024 and rs600026).

Note

The `-y` flag is not documented in the manual pages or *TME 10 Framework Reference Manual Version 3.1*, but it worked for us.

6. Unmount the CD-ROM.
7. Insert the CD-ROM labelled TME 10 Framework Version 3.1.1 Service Pack 1, LK2T-6108-00, in the CD-ROM drive. Mount the CD-ROM again.
8. Install the patch by running the following command on rs600020:

```
wpatch -c /cdrom -i tmf_31_s -y rs600024 rs600026
```

The output of this command should be:

```
Finished patch installation
```

4.3 Installing Oracle for AIX

The relational database management system (RDBMS), in this case, Oracle for AIX, must be installed before installing the TME 10 Inventory application. This is because you are asked to enter database environment variables during TME 10 Inventory installation.

If there is already a supported database system installed, contact your database administrator for the required parameters, as they might be different from those we have chosen.

In the first scenario, we installed the RDBMS server on the same machine (rs600020) where the TMR server was installed.

In this section, we give only an overview of how to install Oracle for AIX and point out some specifics for our environment. For a full explanation, follow the steps described in *TME 10 Cookbook for AIX: Systems Management and Network Applications*, SG24-4867.

1. AIX Base Application Development Toolkit (bos.adt.base) must be installed. To see if it is already installed, from the command prompt, run the following command on rs600020:

```
lslpp -l bos.adt.base
```

The output of this command should be:

```

Fileset                Level State      Description
-----
Path: /usr/lib/objrepos
bos.adt.base           4.2.0.2 COMMITTED  Base Application Development
                                   Toolkit

```

Otherwise, if the output message says that bos.adt.base is not already installed, you have to install it before proceeding.

2. As user root, create dba and oper groups on rs600020:

```
mkgroup '-A' dba
mkgroup '-A' oper
```

3. Create a user ID named oracle on rs600020:

```
mkuser pgrp='dba' groups='dba','staff' home='/home/oracle' oracle
```

4. Create an ORACLE_HOME directory. We decided to use /usr/local/oracle. This directory needs about 310 MB of disk space.

5. Change the owner and group of that directory to oracle and group dba. Change the permission on rs600020 to rwxr-xr-x, as follows:

```
chown oracle.dba /usr/local/oracle
chmod 755 /usr/local/oracle
```

6. Create a bin directory. We chose /usr/lbin. Run the following command on rs600020:

```
mkdir /usr/lbin
```

7. As user oracle, set your environment variables by creating and executing the following shell script named db_setup.sh:

```
#!/bin/ksh
export ORACLE_OWNER=oracle
export ORACLE_HOME=/usr/local/oracle
export PATH=$ORACLE_HOME/bin:/usr/sbin:$PATH
export ORACLE_SID=sid1
```

Figure 63. Shell Script db_setup.sh

We put the script db_setup.sh in the home directory of user oracle (/home/oracle). To run the script, type on rs600020:

```
. db_setup.sh
```

We recommend that you put this command in the .profile file in the home directory of user oracle and the user root.

8. As user root, create a directory as a mount point for the CD-ROM. Create another directory with about 50 MB of disk space to be used by the Oracle installation script as a working directory. We named the directory /oracle_link. Type the following commands on rs600020:

```
mkdir /mount_point
chmod 777 /mount_point
mkdir /oracle_link
chmod 777 /oracle_link
mount -rv cdrfs /dev/cd0 /mount_point
```

9. Provide links from UNIX filenames to the CD-ROM. This process may take up to ten minutes. As user oracle, run these commands on rs600020:

```
cd /mount_point/orainst
./start.sh
```

When prompted for the oracle_link directory, type in the name of your working directory (in this example, /oracle_link).

10. As user root, install post-wait driver kernel extensions on rs600020:

```
cd /oracle_link/orainst
./rootpre.sh
```

11. As user oracle, run the Installer. This may take one hour. Type the following commands on rs600020:

```
cd /oracle_link/orainst
./orainst
```

The commands used with the Oracle Installer are:

<i>Table 3. Oracle Installer Commands</i>	
Commands	Description
LeftCtrl-k	online help
LeftCtrl-x	exit
LeftCtrl-o	operating system prompt

When running the Oracle installation script, you are asked for the following responses:

- a. Enter the value of ORACLE_HOME as /usr/local/oracle.
 - b. Confirm that rootpre.sh has been run.
 - c. Enter the value of ORACLE_OWNER as oracle.
 - d. Enter the value of ORACLE_SID as sid1.
 - e. When asked whether you want to relink all executables, choose **NO**. Otherwise Oracle for AIX will not be installed properly.
 - f. Select the products you want to install. We installed the following products:
 - Oracle Data Query
 - Oracle Easy*SQL
 - ORACLE7 Server (RDBMS)
 - Oracle Unix Installer and Documentation
 - g. You are prompted to select privileged groups and set passwords. The default worked for us.
12. Log in as root, and set your environment by running db_setup.sh.
13. Run the post installation task on rs600020:
- ```
cd $ORACLE_HOME/orainst
sh ./root.sh
```
- If a message that ORACLE\_HOME is not the same as the home directory for this user appears, continue.
14. Edit the file /etc/oratab and change sid1:/usr/local/oracle:N to sid1:/usr/local/oracle:Y. This enables the system to bring up the database at system reboot time.
15. To automatically start the database at system startup, run the following command:
- ```
# mkitab "oracle:2:wait:/bin/su oracle -c $ORACLE_HOME/bin/dbstart"
```

To verify that the installation completed successfully, do these steps:

1. Switch user to oracle on rs600020:

```
su - oracle
```
2. Start the database engine:

```
/usr/local/oracle/bin/dbstart
```

If everything went all right in the installation of Oracle for AIX, the output is:

```
SQL*DBA: Release 7.1.6.2.0 - Production on Wed Jul 23 12:23:06 1997

Copyright (c) Oracle Corporation 1979, 1994. All rights reserved.

Oracle7 Server Release 7.1.6.2.0 - Production Release
With the distributed and parallel query options
PL/SQL Release 2.1.6.2.0 - Production

SQLDBA> Connected.
SQLDBA> ORACLE instance started.
Database mounted.
Database opened.
Total System Global Area      4831032 bytes
      Fixed Size                46596 bytes
      Variable Size            3948852 bytes
Database Buffers              819200 bytes
Redo Buffers                   16384 bytes

SQLDBA>
```

3. To shut down the database, run the command:

```
/usr/local/oracle/bin/dbshut
```

The output should be:

```
SQL*DBA: Release 7.1.6.2.0 - Production on Wed Jul 23 12:22:43 1997

Copyright (c) Oracle Corporation 1979, 1994. All rights reserved.

Oracle7 Server Release 7.1.6.2.0 - Production Release
With the distributed and parallel query options
PL/SQL Release 2.1.6.2.0 - Production

SQLDBA> Connected.
SQLDBA> Database closed.
Database dismounted.
ORACLE instance shut down.
SQLDBA>
SQL*DBA complete.
Database "sid1" shut down.
```

4.4 Installing TME 10 Inventory 3.1

In this section, we install TME 10 Inventory 3.1 using Tivoli commands, instead of using the graphical user interface (GUI).

Before installing TME 10 Inventory 3.1 in the TMR, we backup the database.

Follow these steps to back up the database:

1. Estimate the size of the TME 10 database (the default is to estimate the size of the backup of every managed node). This estimation is not exact, but is very close to the real size of the database image. Run the command on rs600020:

```
wbkupdb -e
```

2. Create a file system of the estimated size (in this example, /backup).
3. Back up the TME 10 database on rs600020, giving as a parameter the name of the file in which you want to save the database image:

```
wbkupdb -d /backup/tme_database.bk
```

The output of this command should be:

```
Backup Complete
```

4.4.1 TME 10 Inventory 3.1 Installation Requirements

To install TME 10 Inventory 3.1 we need the following requirements:

Required Disk Space: You need at least 3.5 MB free on your disk.

Software Requirements:

- AIX 4.1 or 4.2
- Motif V2.1
- TME 10 Framework V3.1
- An RDBMS already installed in the TMR

4.4.2 TME 10 Inventory 3.1 Installation Steps

If all the requirements listed in 4.4.1, "TME 10 Inventory 3.1 Installation Requirements" are met, you are ready to install TME 10 Inventory 3.1. We installed TME 10 Inventory 3.1 on the rs600020 machine.

The installation of TME 10 Inventory 3.1 on an AIX TMR server requires two steps. First you have to install TME 10 Inventory 3.0 and then upgrade to TME 10 Inventory 3.1. While the first step is a product installation, the second one is a patch installation. If you try to install TME 10 Inventory 3.1 directly, you will receive the following message:

```
The following problems were encountered:
Host rs600020:
  Can not find/talk to managed node rs600020
  An instance named "NetWareManagedSite" of resource "rs600020" was
  not found.

Unless you cancel, the following operations will be executed:
Executing queued operation(s)
Registering product installation attributes...registered.
NOTE: The error was due to trying to install 3.1 directly. You must
install first 3.0
NOTE: Check TME 10 Inventory Release Notes Version 3.1 for details

Upgrading to TME 10 Inventory Version 3.1
NOTE: Must be installed as a patch
```

To install TME 10 Inventory 3.0 follow these steps:

1. Insert the CD-ROM disk labelled Tivoli/Inventory 3.0 Revision A, LK2T-6049-00, in the CD-ROM drive.
2. Mount the CD-ROM. We mounted the CD-ROM on the /cdrom directory.
3. Run the command on rs600020:

```
winstall -c /cdrom -i inventor -y RDBMS_Vendor=Oracle
          RDBMS_Host=rs600020 RDBMS_DB_Name=sid1
          RDBMS_DB_Home=/usr/local/oracle
          CreatePaths=1 rs600020
```

Note: Enter all these parameters on a single command line.

The command `winstall` is used to install a TME 10 product (in this example, TME 10 Inventory). The flags have the same meaning as those used with the command `wserver`. You can set the variable `RDBMS_Vendor` to the value Oracle or Sybase and this variable specifies which database you are going to use. The variable `RDBMS_Host` specifies the RIM host that TME 10 Inventory will use to access the RDBMS server. The variable `RDBMS_DB_Name` is set to the database ID used by TME 10 Inventory, while `RDBMS_DB_Home` is set to the directory where you installed the database.

To upgrade TME 10 Inventory 3.0 to TME 10 Inventory 3.1 follow these steps:

1. Insert the CD-ROM disk labelled TME 10 Inventory Version 3.1 Revision A, LK2T-6095-00, in the CD-ROM drive.
2. Mount the CD-ROM. We mounted the CD-ROM on the `/cdrom` directory.
3. Run the following command on rs600020:

```
wpatch -c /cdrom -i inv_sp1 -y rs600020
```

The output of this command should be:

```
Finished patch installation
```

After installing TME 10 Inventory 3.1 on the RIM Host (in this example, rs600020), you can install TME 10 Inventory 3.1 on every managed node you want to scan. We installed TME 10 Inventory 3.1 on rs600024 and rs600026.

Note

You must install TME 10 Inventory on the RIM host first and then on the managed nodes.

To install TME 10 Inventory 3.0 on the managed nodes follow these steps:

1. Insert the CD-ROM disk labelled Tivoli/Inventory 3.0 Revision A, LK2T-6049-00, in the CD-ROM drive.
2. Mount the CD-ROM. We mounted the CD-ROM on the `/cdrom` directory.
3. Run the following command on rs600020:

```
winstall -c /cdrom -s rs600020 -i inventor -y CreatePaths=1  
rs600024 rs600026
```

Note: Enter all these parameters on a single command line.

The flag `-s` specifies the managed node used as the product's installation server.

To upgrade TME 10 Inventory 3.0 to TME 10 Inventory 3.1 follow these steps:

1. Insert the CD-ROM disk labelled TME 10 Inventory Version 3.1 Revision A, LK2T-6095-00, in the CD-ROM drive.
2. Mount the CD-ROM. We mounted the CD-ROM on the `/cdrom` directory.
3. Run the following command on rs600020:

```
wpatch -c /cdrom -i inv_sp1 -y rs600024 rs600026
```

The output of this command should be:

Finished patch installation

4.5 Configuring Oracle for AIX for Use with TME 10 Inventory 3.1

Before you can configure Oracle for AIX for use with TME 10 Inventory 3.1, you must have installed the RDBMS server (in this example, Oracle).

The RDBMS server can be located anywhere on the LAN. In our scenarios, we show you how to have the RDBMS server inside or outside the TMR, for example:

- RDBMS on the TMR server, see 4.1.1, “Scenario 1: RDBMS Server and RIM Host on the TME Server” on page 64.
- RDBMS outside the TMR, see 4.1.2, “Scenario 2: RDBMS Server Located Outside the TMR” on page 64.

To use the inventory application on a managed node, you must install TME 10 Inventory. A managed node is a workstation with TME 10 Framework installed. To use the inventory application on a PC managed node, you must install TME 10 Inventory PC Scanning Program, not TME 10 Inventory. A PC managed node is a workstation with the PC Agent installed, not TME 10 Framework.

Be sure to read the TME 10 Inventory 3.1 Release Notes:

1. Install TME 10 Inventory 3.0 on the TMR server and all managed nodes.
2. Install patch to upgrade TME 10 Inventory 3.0 to TME 10 Inventory 3.1 on the TMR server and all managed nodes.
3. Do not install TME 10 Inventory 3.1 for NT unless the NT managed node does not have TME 10 Inventory 3.0 installed.
4. Install TME 10 Inventory PC Scanning Program (3.1) on all PC managed nodes.

On the RDBMS server (in this example, the same as the TMR server), create the inventory user account and install the configuration repository database by running, as user sys, the `tivoli_ora_admin.sql` script and then the `tivoli_ora_schema.sql` script for Oracle. These scripts are in the `$BINDIR/TAS/RIM/SQL/scripts` directory.

For further details on creating the inventory user account and installing the configuration repository database, refer to *TME 10 Cookbook for AIX: Systems Management and Network Applications*, SG24-4867.

4.5.1 Update Database Tables

At this point, you have created on the RDBMS server, the tables required by TME 10 Inventory.

Note

The TME 10 Inventory 3.1 Release Notes do not document the SQL upgrade scripts directory.

To update these tables to use them with TME 10 Inventory 3.1, follow these steps:

1. As the AIX root user, on rs600020, type:

```
cd $BINDIR/TME/INVENTORY/SCRIPTS
sqlplus
```

The output should look like this:

```
SQL*Plus: Release 3.1.3.7.1 - Production on Thu Jul 24 14:15:26 1997
Copyright (c) Oracle Corporation 1979, 1994. All rights reserved.

Enter user-name: tivoli
Enter password:

Connected to:
Oracle7 Server Release 7.1.6.2.0 - Production Release
With the distributed and parallel query options
PL/SQL Release 2.1.6.2.0 - Production

SQL>
```

When Oracle prompts you to enter the user name, enter `tivoli` and the password. You should be logged on to the Oracle database.

2. Run the SQL script from the command line:

```
SQL>@ora_sp1_patch.sql
```

4.5.2 Erase Database Contents

If for any reason you need to erase the database contents related to Tivoli, do the following:

1. To drop all database tables used by TME 10 Inventory, first make sure that TME 10 Inventory is not running and then from the command line type on rs600020:

```
cd $BINDIR/TAS/RIM/SQL/scripts
sqlplus
```

2. Log in as `tivoli` user.
3. Install the configuration repository database:

```
SQL>@tivoli_ora_schema.sql
```

The output should look like this:

```
Commit complete.
```

4. Run the script to update the database tables:

```
SQL>@$BINDIR/TME/INVENTORY/SCRIPTS/ora_sp1_patch.sql
```

The output should look like this:

```
Commit complete.
```

5. Exit from SQL*PLUS:

```
SQL>quit
```

Note

Some error messages may appear because the script tries to drop a table that doesn't exist.

4.6 An Example Using Oracle for AIX and TME 10 Inventory

So far we have all the elements needed to use Oracle for AIX with TME 10 Inventory. In this section we show you how to make a hardware profile scan of the TMR managed nodes from the command line.

The TMR we are using in this example is composed only of AIX machines, so the hardware profile scanning is the only action we can perform. The example shows all the steps to do the scanning from the command line when the TMR administrator doesn't have a graphics display available.

The components of this scenario are:

- RS600020 is the TMR server with TME 10 Inventory 3.1 and Oracle for AIX Version 7.1 installed.
- RS600024 is a managed node with TME 10 Inventory 3.1.
- RS600026 is a managed node with TME 10 Inventory 3.1.

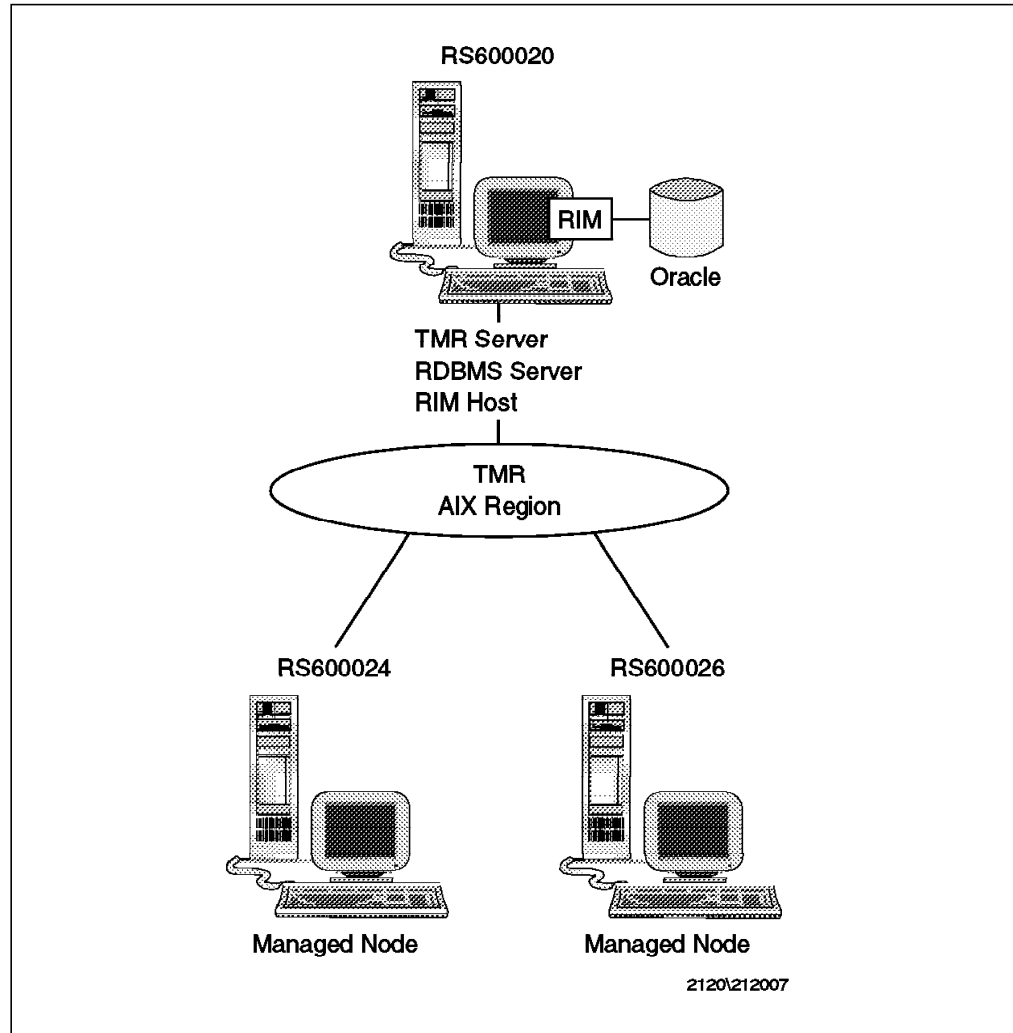


Figure 64. TMR Server, RDBMS Server, and RIM Host on Same Workstation

All the commands are performed on the TMR server, in this example, RS600020.

1. Create a profile manager, named inventory. Log in as root and type the following on rs600020:

```
wcrtprfmgr @aixregion inventory
```

The output should look like this:

```
1693108358.1.574#TMF_CCMS::ProfileManager#    inventory
```

Note

To list resources available in a TMR, check the options for the wlookup command.

2. Subscribe the managed nodes to the profile manager inventory:

```
wsub @inventory @ManagedNode:rs600020 @ManagedNode:rs600024 @ManagedNode:rs600026
```
3. Verify that all managed nodes are included in the profile manager inventory:

```
wgetsub @inventory
```

The output should look like this:

```
rs600020  
rs600024  
rs600026
```

4. Create an inventory profile inside the profile manager inventory. The name for our profile is aixscan.

```
wcrtprf @inventory InventoryProfile aixscan
```

The output should look like this:

```
1693108358.1.575#InventoryProfile#    aixscan
```

5. The `wls` command lists all objects inside a collection. If you want to list all profiles inside the profile manager inventory, type:

```
wls @ProfileManager:inventory
```

The output should look like this:

```
aixscan
```

6. Subscribe the region administrator to the inventory notification bulletin board. First list all available bulletin boards in the region by typing:

```
wlookup -a -r TMF_Notice
```

The output should look like this:

```
AEF      1693108358.1.477#TMF_Notice::Group#  
Inventory 1693108358.1.548#TMF_Notice::Group#  
TME Administration 1693108358.1.97#TMF_Notice::Group#  
TME Authorization 1693108358.1.98#TMF_Notice::Group#  
TME Diagnostics 1693108358.1.103#TMF_Notice::Group#  
TME Scheduler 1693108358.1.159#TMF_Notice::Group#
```

After you know the name of the bulletin board for TME 10 Inventory, on the command line, type:

```
wsetadmin -n Inventory Root_aixregion
```

The `-n` flag indicates that you are adding a notice group and, in this example, our region administrator is `Root_aixregion`.

7. Add TME 10 Inventory 3.1 roles to the region administrator. The roles allow the region administrator to perform the hardware profile scan. According to *TME 10 Inventory User's Guide Version 3.1*, the needed roles are:

- Inventory_view
- Inventory_edit
- Query_view
- Query_execute
- RIM_view
- user
- admin
- senior

- super
- Install_product

To add the roles type:

```
wsetadmin -r
    @PolicyRegion:aixregion,
    Inventory_view:Inventory_edit:Inventory_query:Inventory_scan
    Root_aixregion
```

Note

Type all the command parameters in a single line and be sure to include all the roles you had before, because wsetadmin overrides all prior values.

8. Include the RIM_view role in the region administrator. The region administrator (Root_aixregion) needs the RIM_view role on the whole TMR to view managed nodes hardware profiles. To set this role, type:

```
wsetadmin -r global,
    super:senior:admin:user:install_client:install_product:RIM_view
    Root_aixregion
```

9. Perform the scan process. For this example, we choose to scan all managed nodes. To perform the scanning process, type:

```
wdistrib @InventoryProfile:aixscan @ManagedNode:rs600020 @ManagedNode:rs600026
```

10. The scanning process is now complete. You can review the scanned hardware data for a managed node. To view rs600020 hardware inventory data, type:

```
wviewmn rs600020
```

The output should look like this:

```
Hardware System Id      : 1693108358.1.327#TMF_ManagedNode
                        : :Managed_node
Hardware Architecture   : power
Booted Operating System Name : AIX
Processor Model         : UNKNOWN
Physical Memory (kB)    : 131072
Paging Space (kB)      : 163840
Time of Last Scan       : 1997-07-28 10:34:19
```

4.7 Moving the RDBMS Server Outside the TMR

With this example, we want to show that you can have the RDBMS server and the RIM host on different machines and, also, the RDBMS server can be outside the boundaries of the TMR. In our scenario, we install the RIM host on the TME server (rs600020) and the RDBMS server on the rs600024 (see Figure 65 on page 82).

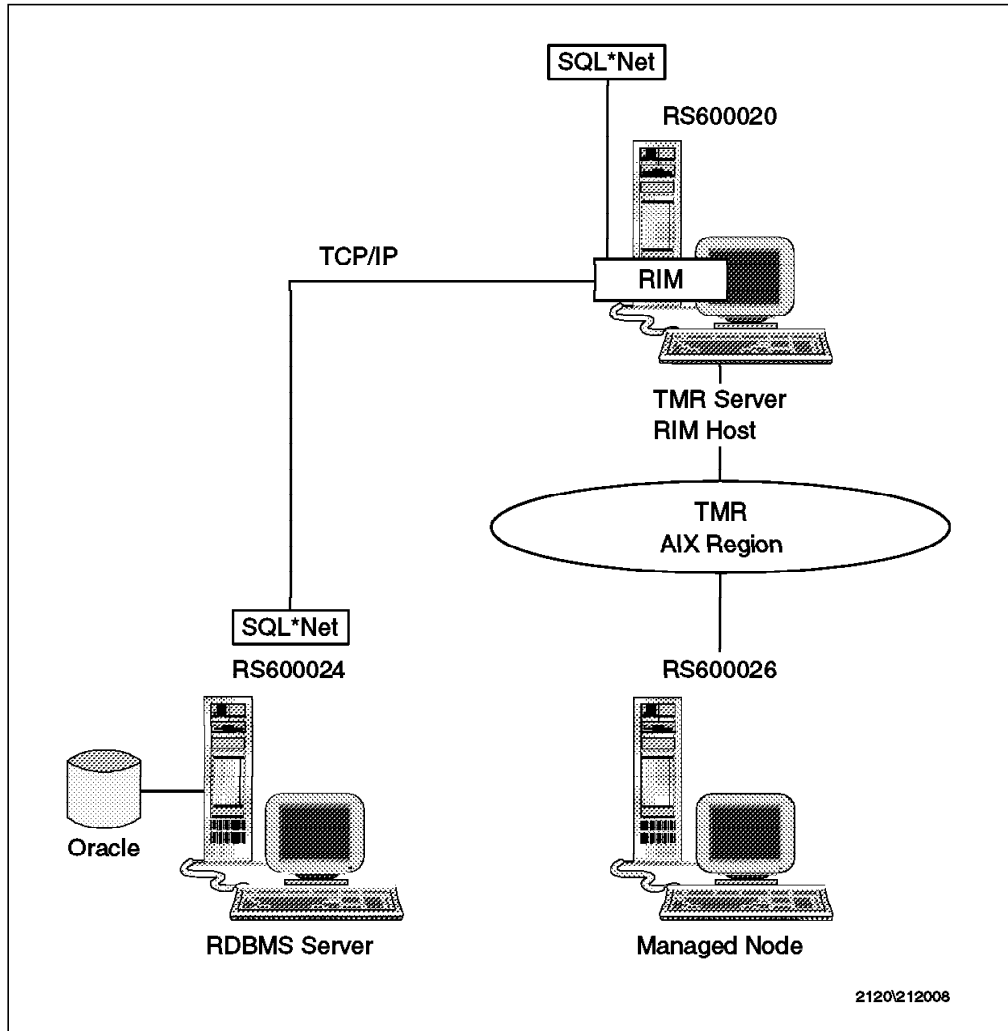


Figure 65. RDBMS Server Outside the TMR

For more details, you can read 4.1.1, “Scenario 1: RDBMS Server and RIM Host on the TME Server” on page 64.

The RIM host and the RDBMS server communicate through SQL*Net. SQL*Net allows applications to access, manipulate and store data in a remote Oracle database. The role of SQL*Net is to establish and maintain a connection between the client machine where the application resides, and the server machine where resides the Oracle database. When a connection is required from the client application, SQL*Net is responsible for checking possible differences in the internal data and character set representation of the two machines. Any differences are resolved by SQL*Net automatically. For details on how SQL*Net works, you can refer to *Understanding SQL*Net Version 2.1* by Oracle Corporation.

Once we installed all the software properly, we could scan the managed node rs600026, from the RIM machine rs600020. Later in 4.7.4, “Testing the Scenario” on page 96, we show you how to test this configuration.

In this section, we show you how to remove the software components of the previous example that you don’t need anymore. We show how to remove Tivoli

and Oracle software. Then we install the software for this scenario on the machines, showing that this configuration works.

4.7.1 Removing a Managed Node from the TMR

To remove the managed node rs600024 from the TMR, you must remove the managed node definition and remove the Tivoli software from the managed node.

4.7.1.1 Removing Managed Node Definition from TMR

In the previous example, we had a TMR, with a policy region called aixregion (see Figure 66).



Figure 66. Policy Region aixregion

Inside this Policy Region, we had the Profile Manager inventory with the inventory profile called aixscan and the three subscribers rs600020, rs600024 and rs600026 (see Figure 67 on page 84).

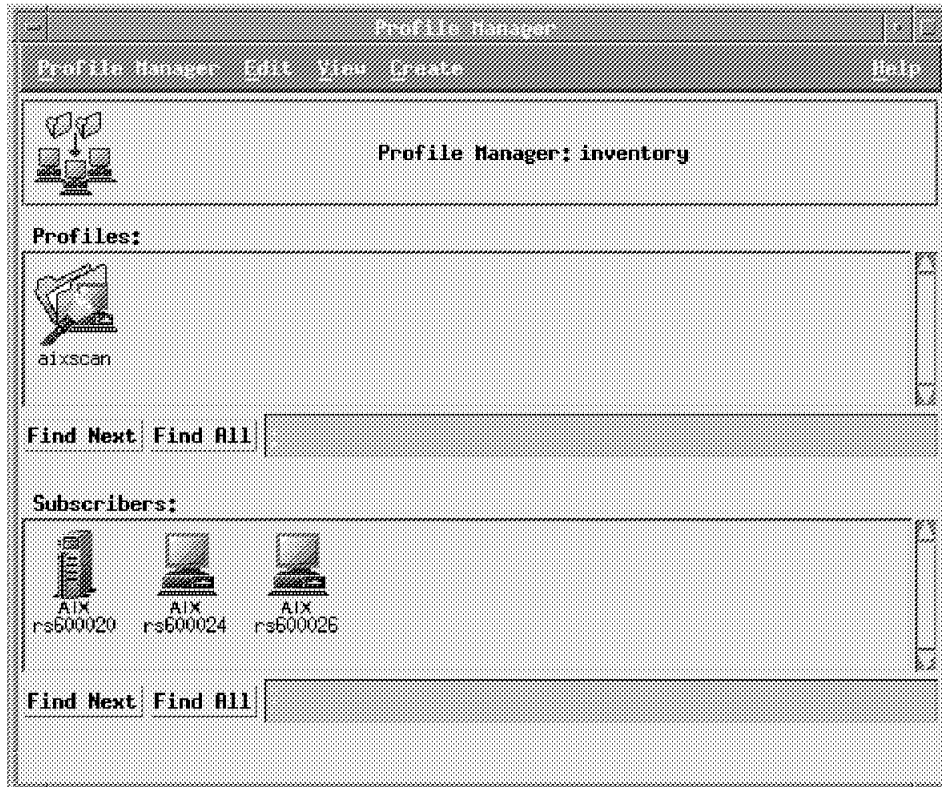


Figure 67. Profile Manager:inventory

To remove the managed node rs600024 from the TMR follow these steps:

1. We strongly suggest that you back up the database before changing the configuration of the TMR. To back up the database, as user root from the machine rs600020 run the command:

```
wbkupdb -d /backup/first_scenario.bk
```

The -d flag specifies the file to which the database will be saved. For more information about the wbkupdb command, see 4.4, "Installing TME 10 Inventory 3.1" on page 73.

Note

You must have the backup role to perform this operation.

The output of this command should be:

```
Backup Complete.
```

2. From the TMR desktop double click on the icon representing the policy region you want to work with (in this example, aixregion).
3. Select the managed node to delete (in this example, rs600024).
4. From the menu bar select **Edit** and then **Delete** (see Figure 68 on page 85).

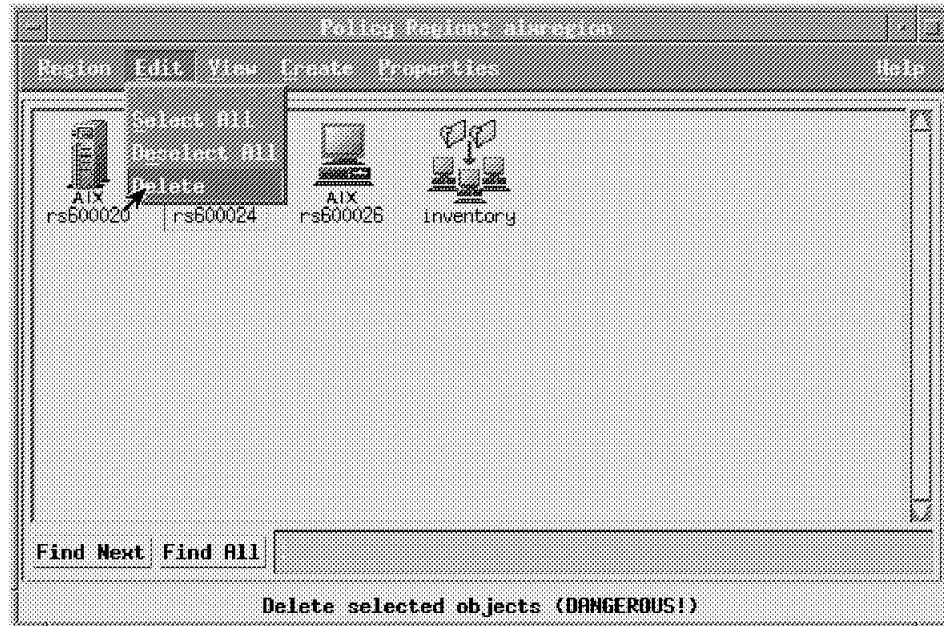


Figure 68. Deleting a Managed Node from the TMR

5. On the next window click **Yes** (see Figure 69). Be careful that the delete command removes the managed node from the TMR database.



Figure 69. Confirm the Deletion of a Managed Node from the TMR

Note that after this operation the managed node rs600024 is removed also from the subscribers of the profile manager inventory.

6. To check if the operation was successful follow these steps:

- a. List all the active TME 10 clients. Run the command:

```
odadmin odlist
```

The output of this command should be:

```

1693108358      1   ct-   94   9.24.104.241  rs600020.itso.ra1.ibm.c,rs600020
                 3   ct-   94   9.24.104.58   rs600026.itso.ra1.ibm.c

```

Note that the managed node rs600024 is no longer in the list.

- b. Check if there are inconsistencies in the database. Run the command:

```
wchkdb
```

If no errors are detected, you removed the managed node successfully.

Otherwise you can try to fix the errors with the command:

```
wchkdb -u
```

The flag `-u` updates the database, and fixes all the errors found in the TMR database.

4.7.1.2 Remove Tivoli Software

If you want to remove the Tivoli software and free the disk space used by the Tivoli software from the rs600024 machine, you can remove the file systems `/var/spool/Tivoli` and `/usr/local/Tivoli`. As user root, from the rs600024 machine, do the following:

1. Unmount the file systems where you installed Tivoli:

```
umount /usr/local/Tivoli /var/spool/Tivoli
```

2. Remove those file systems:

```
rmfs -r '' /usr/local/Tivoli
rmfs -r '' /var/spool/Tivoli
```

3. Erase from the file `.profile` of user root the lines that set up Tivoli variables. The lines you have to erase are the ones in which you run the script `setup_env.sh`. In this example:

```
if [ -f /etc/Tivoli/setup_env.sh ]; then
  . /etc/Tivoli/setup_env.sh
fi
```

4.7.2 Removing Oracle for AIX from TMR Server

To remove Oracle for AIX from the TMR server (rs600020) follow these steps:

1. Log in as user root on rs600020.

2. Remove Oracle for AIX directory:

```
rm -rf /usr/local/oracle
```

3. Remove user oracle and its home directory:

```
rmuser -p 'oracle'
rm -rf /home/oracle
```

4. Remove groups dba and oper:

```
rmgroup 'dba'
rmgroup 'oper'
```

5. Remove the directory for the binaries:

```
rm -rf /usr/lbin
```

6. Remove from file `.profile` of user root the line to run `db_setup.sh` script (we used `db_setup.sh` to set the environment variables for Oracle for AIX).

7. Remove the file `/etc/oratab`:

```
rm -f /etc/oratab
```

8. Erase from the file /etc/inittab the line to start automatically the Oracle database at system reboot. The line is:

```
oracle:2:wait:/bin/su oracle -c /usr/local/oracle/bin/dbstart
```

4.7.3 Installing Software for the Second Scenario

Now we are ready to install all the software needed to configure the environment of this scenario. We install Oracle for AIX Version 7.1, SQL*Net Version 2.1 and Oracle Network Manager Version 1.0 on the rs600024 machine (the new RDBMS server). We must also install SQL*Net on rs600020 to make the connection between rs600020 and rs600024.

Oracle Network Manager is a tool that helps you to create the configuration files needed by SQL*Net to work properly. Oracle Network Manager creates both the files for the RDBMS server and for the RIM host. You can also create these files with a text editor, but Oracle Network Manager provides a GUI to make this job easier and safer.

Note

Oracle Network Manager is a tool that runs on machines with a graphical user interface (Motif). You cannot use Oracle Network Manager with a character-mode terminal. So be sure to have Oracle Network Manager on the machine with the Motif libraries.

4.7.3.1 Installing Oracle for AIX Products on RDBMS Server

To install Oracle for AIX products, follow these steps on the RDBMS server (rs600024):

1. Add, in the .profile file for user oracle, the following line:

```
export LD_LIBRARY_PATH=/usr/lib/X11
```

The LD_LIBRARY_PATH variable is needed by the Oracle Installer to relink the executable. The value of this variable must be the complete path where the Motif libraries (usually /usr/lib/X11) reside.

2. Follow the steps described in 4.3, "Installing Oracle for AIX" on page 70. When you run the Oracle Installer, be careful of the following:
 - a. When you are asked to confirm that you have already run the preinstallation scripts, choose **yes** (see Figure 70 on page 88).

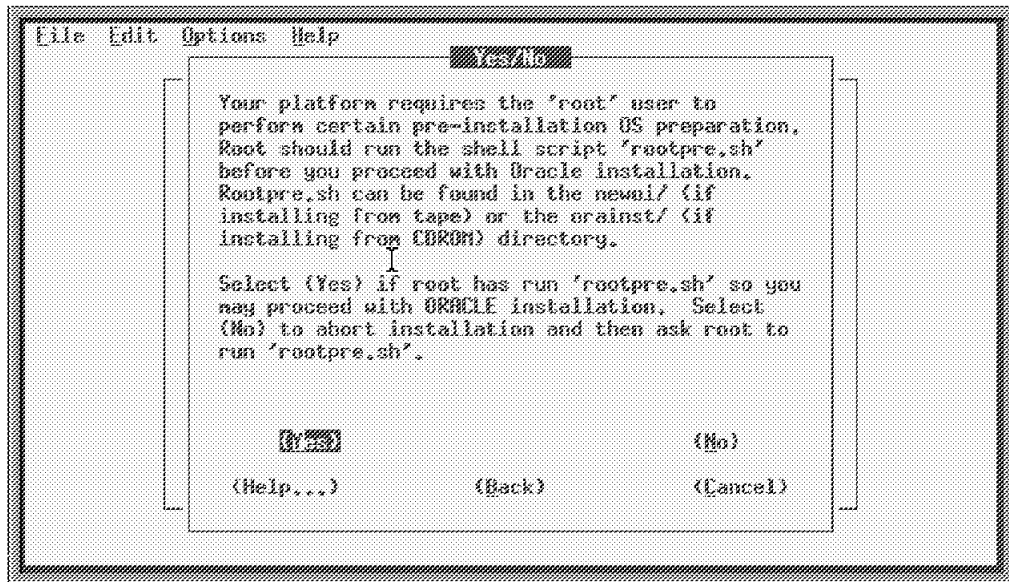


Figure 70. Oracle Installer: Confirm You Have Already Run the Preinstallation Script

- b. Choose the Complete Software/Database Fresh Install type of installation.
- c. In the Relink All Executables? window choose **No** (see Figure 71). Otherwise Oracle for AIX will not be installed properly.

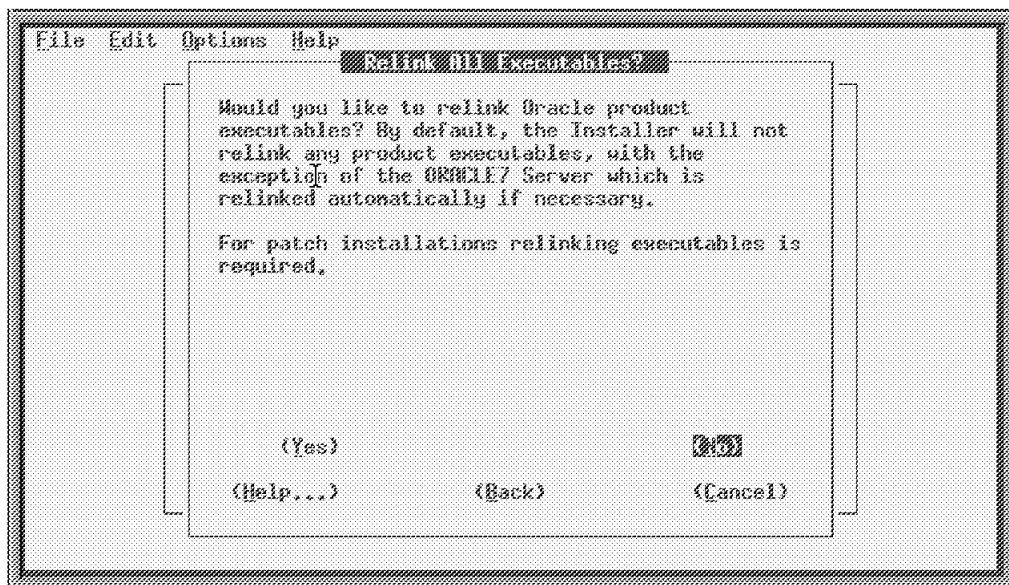


Figure 71. Oracle Installer: Relink All Executables?

- d. Choose the following products to install:
 - Oracle Data Query
 - Oracle Easy*SQL
 - Oracle Network Manager
 - Oracle Unix Installer and Documentation
 - ORACLE7 Server (RDBMS)

- SQL*Net TCP
 - SQL*Net V2
 - TCP/IP Protocol Adapter (V2)
- e. After Oracle Installer asks you to give some passwords, you are asked if you want SQL*Net Listener to start automatically. Choose **Yes** (see Figure 72).

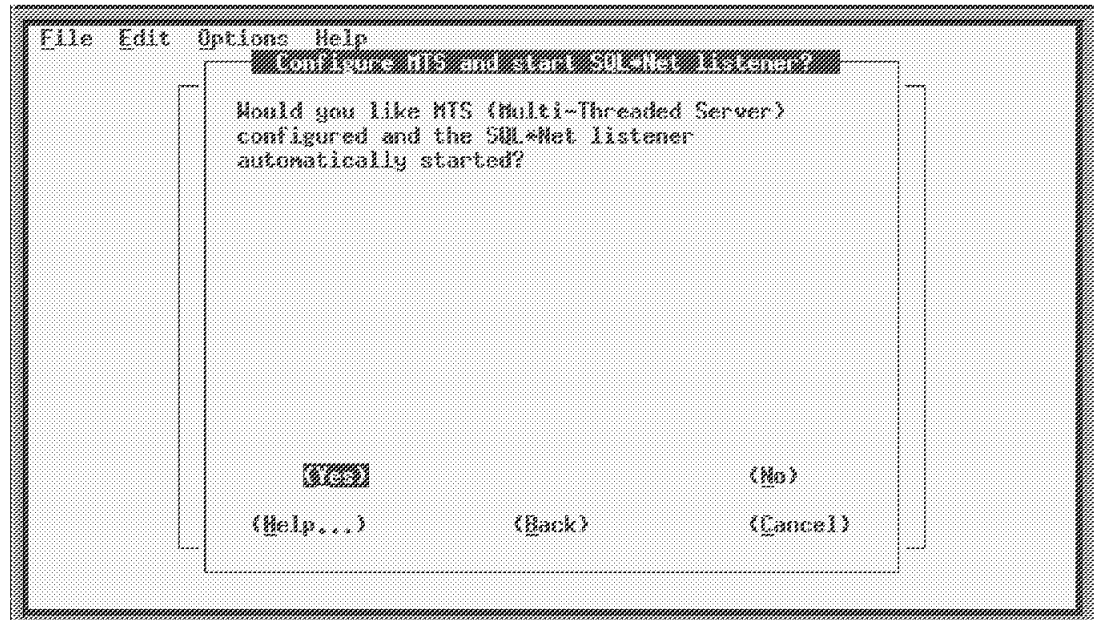


Figure 72. Oracle Installer: SQL*Net Listener

SQL*Net Listener is a protocol-independent process, running on the RDBMS server, that receives all the incoming requests for connection from the client applications. When a connection request is made by a client application, the listener recognizes this request and passes it to a dedicated server process, called the listener process. SQL*Net Listener is an application included in a software layer called Transparent Network Substrate (TNS). TNS is able to mask the different network protocols to Oracle products.

- f. At the end of the installation process, Oracle Installer warns you that SQL*Net Listener was not relinked. This is not an error.
3. As user root run the post-installation scripts:
- ```
./root.sh
```

#### 4.7.3.2 Installing SQL\*Net Version 2.1 on the RIM Host

To establish communication between the RIM host (rs600020) and the RDBMS server, you have to install SQL\*Net Version 2.1 on the RIM host. Do these steps on the RIM host (rs600020):

1. Follow the steps described in 4.3, "Installing Oracle for AIX" on page 70. When you run the Oracle Installer, be careful about the following:
  - a. Choose **Install/Upgrade/Patch Software Only** type of installation.
  - b. Choose not to relink the executables.
  - c. The products to install are:

- Oracle Easy\*SQL
- Oracle Data Query
- SQL\*Net TCP
- SQL\*Net V2

2. After the installation has completed successfully, as user root, run the post-installation script:

```
./root.sh
```

### 4.7.3.3 Configuring SQL\*Net Version 2.1 Using Oracle Network Manager

To configure SQL\*Net, you have to write information in the configuration files. SQL\*Net reads these files to get the information to establish a correct connection between the RDBMS server and the client application. Instead of creating these configuration files with a text editor, we used Oracle Network Manager.

#### Note

For more information about Oracle Network Manager Version 1.0, see *Oracle Network Manager Administration Guide Version 1.0* from Oracle Corporation.

To configure SQL\*Net, do these steps on the RDBMS server (rs600024):

1. Log in as root.
2. Add an entry in the /etc/inittab file to start Oracle Listener automatically:  

```
mkitab "oracle:2:wait:/bin/su - oracle -c lsnrctl start"
```
3. Log in as user oracle.
4. Set the DISPLAY variable properly. Set the DISPLAY variable also if you want to run Oracle Network Manager from the console:  

```
export DISPLAY=.....
```

If you do not set the DISPLAY variable properly, the following message appears:

```
NMC-0010: Resource file cannot be opened
```

5. Run Oracle Network Manager (see Figure 73 on page 91):  

```
netman
```

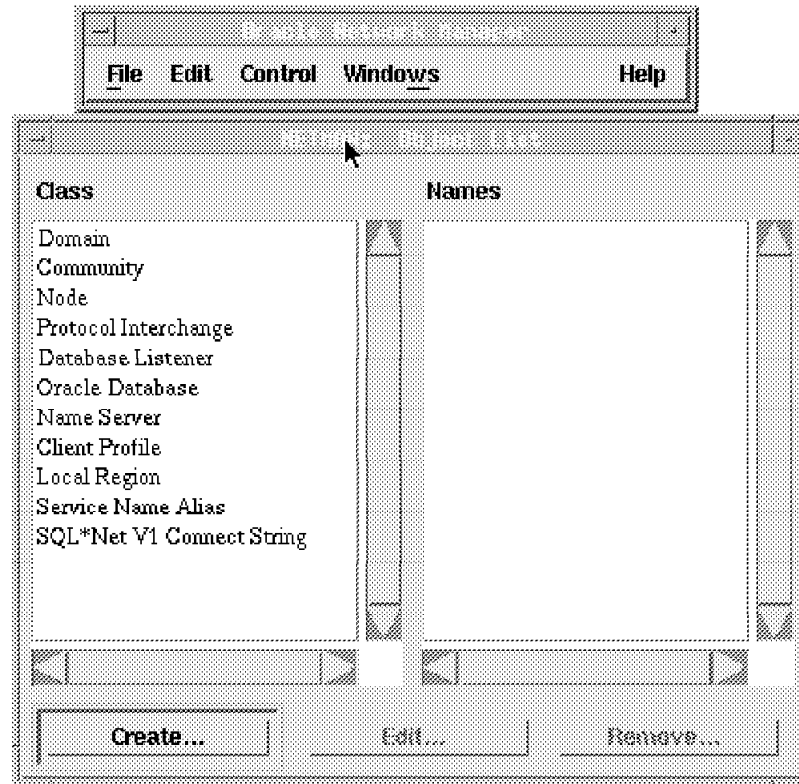


Figure 73. Oracle Network Manager: Main Window and Menu

6. Select **File** from the menu bar and then **New** from the pull-down menu.
7. From the Object List window, double click on **Community**. This action opens the window to set the parameters for the community (see Figure 74 on page 92).

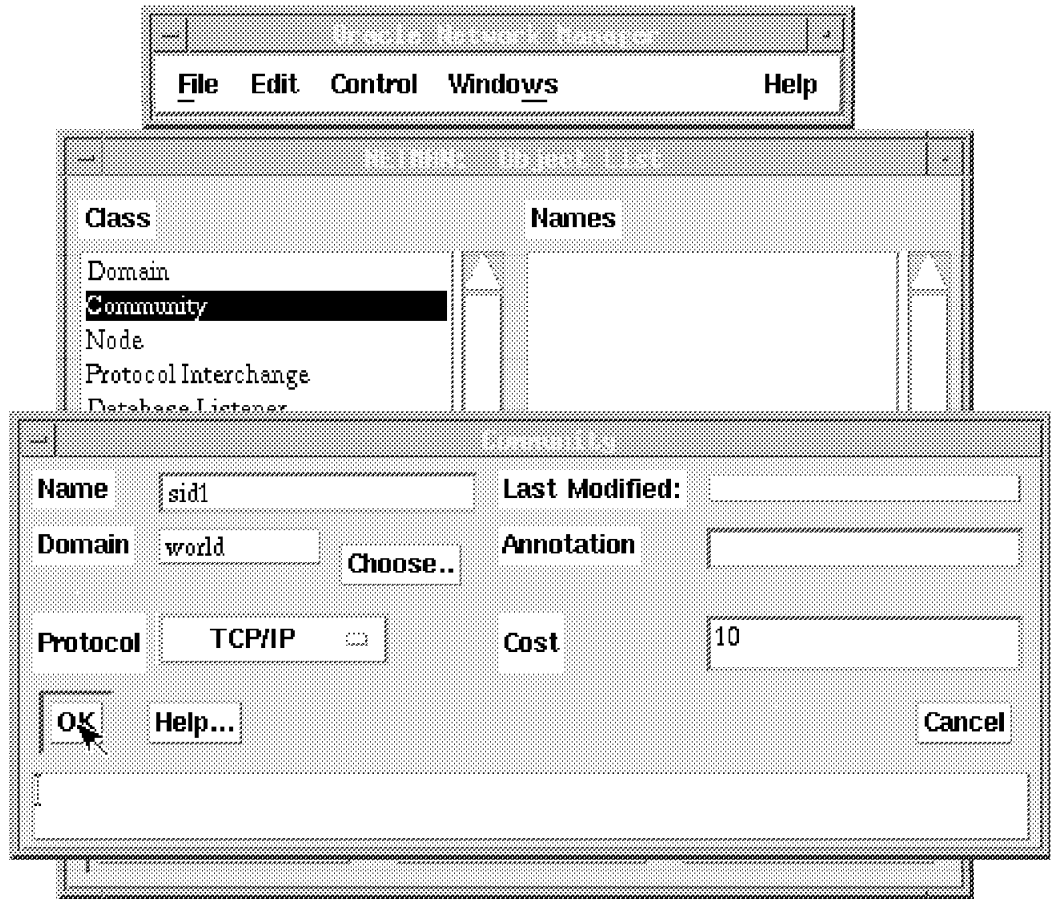


Figure 74. Oracle Network Manager: Community Parameters

**Note**

In SQL\*Net terminology, a *community* is a group of nodes (machines) on which is installed SQL\*Net or TNS-based software, that communicates using the same transport protocol.

In this example, the community consists of the machines rs600020 and rs600024 that communicate using TCP/IP.

Set these parameters:

- Name (name of the community): sid1
- Protocol: TCP/IP

Choose **OK** to exit from the Community window.

8. From the Object List window double click on **Node**. This action opens the window to set the parameters for the nodes (machines) of the community. We need to define only the machine that acts as RDBMS server (rs600024) (see Figure 75 on page 93).



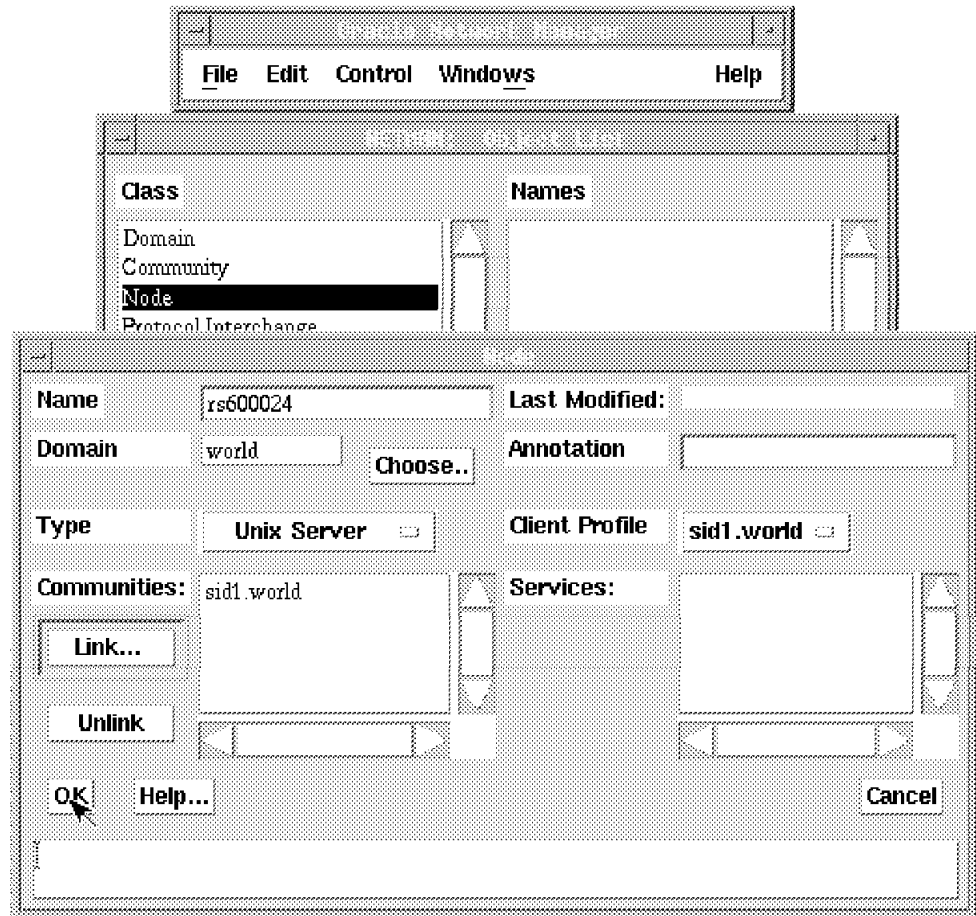


Figure 75. Oracle Network Manager: Node Parameters

Set these parameters:

- Name (name of the RDBMS server): rs600024
- Type: Unix Server
- In the Communities section, choose **Link**. In the window that is opened, select the community name **sid1** and click on **OK** to return to the previous window.

Choose **OK** to exit from the Node window.

9. From the Object List window, double click on **Database Listener**. This action opens the window to set the parameters for the SQL\*Net Listener (see Figure 76 on page 94).

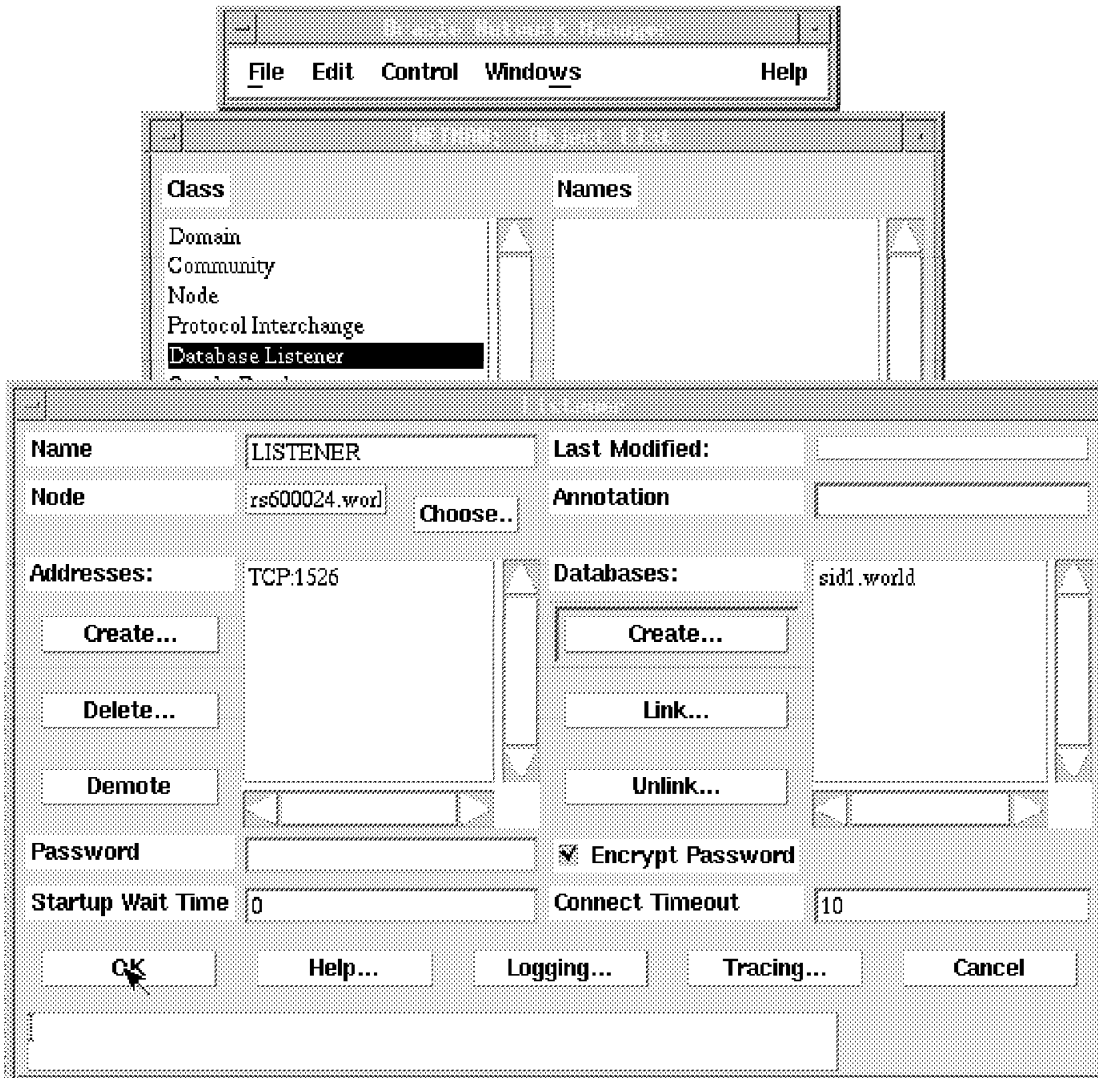


Figure 76. Oracle Network Manager: SQL\*Net Listener Parameters

Do the following steps:

- In the Name field leave the default value (LISTENER).
- In the Node field leave the default name.
- In the Addresses section choose **Create**. In the next window verify that everything is right, and then click on **OK** to return to the Database Listener window. If you use TCP/IP there is nothing to do in the Addresses section.
- In the Database section, choose **Create**. This opens the window to set the parameters for the database for which the Listener process listens (see Figure 77 on page 95).

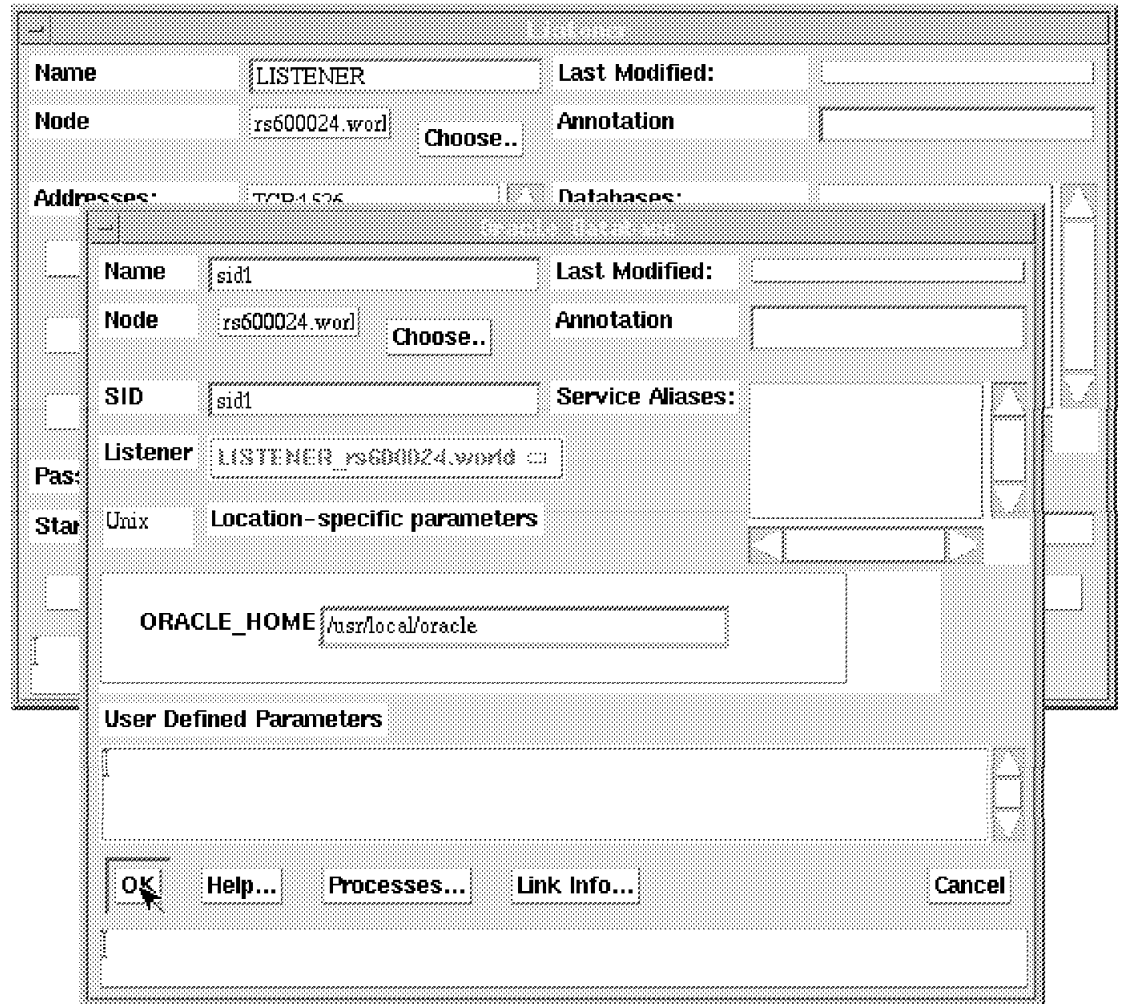


Figure 77. Oracle Network Manager: Database Parameters

Set the following parameters:

- Name (the name of the database): sid1
- Node: use the one already shown.
- SID: sid1
- Location-specific parameters (the value of the variable ORACLE\_HOME): /usr/local/oracle
- Click on **OK** to return to the previous window.

Click on **OK** to return to the Object List window.

- Select **File** from the menu bar and then **Validate** from the pull-down menu. This action allows you to verify that all the information you entered is complete and consistent.
- Select **File** from the menu bar and then **Save** from the pull-down menu. You have to decide in which directory you want to save your configuration for future reuse. We put our configuration in /home/oracle/sid1.net.
- Select **File** from the menu bar and then **Generate** from the pull-down menu. This option creates the configuration files for both the RDBMS

server and the RIM host. Once you create these configuration files, you must distribute them to the RDBMS server and to the RIM host. You must choose the directory in which you want to put the files. We choose /usr/local/oracle.

**Note**

This directory must already exist.

Starting from the directory you chose, Oracle Network Manager creates a subdirectory called sid1 (the name of the community you created). Under sid1, there are two subdirectories, one for the files for the RDBMS server (rs600024\_world) and the other for the files for the RIM host (sid1\_world). Oracle Network Manager creates two files for the RIM host and three for the RDBMS server. The files are:

**tnsnames.ora** For both the RIM host and the RDBMS server. This file contains all the required information to connect the nodes properly. See Figure 230 on page 241.

**sqlnet.ora** For both the RIM host and the RDBMS server. This file contains diagnostic parameters (how to log and trace the connection) for SQL\*Net. See Figure 231 on page 241.

**listener.ora** Only for the RDBMS server. This file contains the information required to start and run a network Listener process.

- To quit from Oracle Network Manager, select **File** from the menu bar and then **Quit** from the pull-down menu.

10. Distribute the file to the RIM host and to the RDBMS server. As user oracle, you have to copy the two files for the RIM host from /usr/local/oracle/sid1/sid1\_world to the directory \$ORACLE\_HOME/network/admin of the RIM host (rs600020). Then, still as user oracle, you have to copy the three files for the RDBMS server from /usr/local/oracle/sid1/rs600024\_world to the directory \$ORACLE\_HOME/network/admin of the RDBMS server (rs600024). Moreover, you must add in the .profile file for user oracle, both in the RIM host and in the RDBMS server, the line:

```
export TNS_ADMIN=/usr/local/oracle/network/admin
```

The TNS\_ADMIN variable is used by SQL\*Net when it needs the configuration files and its value must be the directory where you stored the configuration files.

## 4.7.4 Testing the Scenario

After having installed all the software, you are now able to test if everything works properly.

### 4.7.4.1 Testing the Listener

As user oracle, start the Listener process:

```
lsnrctl start
```

The output of this command should be:

```
LSNRCTL for IBM/AIX RISC System/6000: Version 2.1.6.1.0 -
 Production on 30-JUL-97 16:54:01

Copyright (c) Oracle Corporation 1994. All rights reserved.

Starting /usr/local/oracle/bin/tnslsnr: please wait...

TNSLSNR for IBM/AIX RISC System/6000: Version 2.1.6.1.0 -
 Production on 30-JUL-97 16:54:20
System parameter file is /usr/local/oracle/network/admin/listener.ora
Log messages written to /usr/local/oracle/network/log/listener.log
Listening on: (DESCRIPTION=(CONNECT_TIMEOUT=10)(ADDRESS= (PROTOCOL=IPC)
 (KEY=sid1.world)))
Listening on: (DESCRIPTION=(CONNECT_TIMEOUT=10)(ADDRESS= (PROTOCOL=IPC)
 (KEY=sid1)))
Listening on: (DESCRIPTION=(CONNECT_TIMEOUT=10)(ADDRESS= (COMMUNITY=sid1.world)
 (PROTOCOL=TCP)(Host=rs600024)(Port=1526)))

Connecting to (ADDRESS=(PROTOCOL=IPC)(KEY=sid1.world))
STATUS of the LISTENER

Alias LISTENER
Version TNSLSNR for IBM/AIX RISC System/6000:
 Version 2.1.6.1.0 - Production on 30-JUL-97 16:54:20
Start Date 30-JUL-97 16:54:05
Uptime 0 days 0 hr. 0 min. 0 sec
Trace Level off
Security OFF
Listener Parameter File /usr/local/oracle/network/admin/listener.ora
Listener Log File /usr/local/oracle/network/log/listener.log
Services Summary...
 sid1 has 1 service handlers
The command completed successfully
```

With the command `lsnrctl` you start a tool, called Listener Control Utility, to control the Listener process.

To control the status of the Listener process, run the following command:

```
lsnrctl status
```

The output of this command should be:

```
LSNRCTL for IBM/AIX RISC System/6000: Version 2.1.6.1.0 -
Production on 30-JUL-97 16:59:56
```

Copyright (c) Oracle Corporation 1994. All rights reserved.

```
Connecting to (ADDRESS=(PROTOCOL=IPC)(KEY=sid1.world))
STATUS of the LISTENER
```

```

Alias LISTENER
Version TNSLSNR for IBM/AIX RISC System/6000:
 Version 2.1.6.1.0 - Production
Start Date 30-JUL-97 16:54:05
Uptime 0 days 0 hr. 5 min. 51 sec
Trace Level off
Security OFF
Listener Parameter File /usr/local/oracle/network/admin/listener.ora
Listener Log File /usr/local/oracle/network/log/listener.log
Services Summary...
 sid1 has 2 service handlers
The command completed successfully
```

#### Note

For a detailed explanation of the lsnrctl command, see *Understanding SQL\*Net Version 2.1* from Oracle Corporation.

### 4.7.4.2 Testing the Communication between RIM Host and RDBMS Server

To do this test, we connect to the database sid1, that resides on the RDBMS server (rs600024), from the RIM host (rs600020). So, from rs600020, as user oracle, run the following command:

```
sqlplus sys/oracle@sid1
```

The output of this command should be:

```
SQL*Plus: Release 3.1.3.7.1 - Production on Tue Jul 29 18:45:02 1997
```

Copyright (c) Oracle Corporation 1979, 1994. All rights reserved.

```
Connected to:
Oracle7 Server Release 7.1.6.2.0 - Production Release
With the distributed and parallel query options
PL/SQL Release 2.1.6.2.0 - Production
```

```
SQL>
```

#### Note

If the configuration files are not in the proper directory, you will get error code ORA-12154. You can find information about Oracle errors at the Web site <http://www.oracle.com/support/bulletins/html>.

### 4.7.4.3 Testing Tivoli

We now want to test if Tivoli works properly. First, we configure Oracle for AIX on the RDBMS server (rs600024) to work with the new scenario. To do this, follow the steps in 4.5, “Configuring Oracle for AIX for Use with TME 10 Inventory 3.1” on page 76.

For information about the RIM object, run the following command:

```
wgetrim inventory
```

where inventory is the label defined in the TMR for the RIM object we want information about. The output of this command should be:

```
RIM Host: rs600020
RDBMS User: tivoli
RDBMS Vendor: Oracle
Database ID: sid1
Database Home: /usr/local/oracle
Server ID: sid1
```

Verify that the Server ID parameter is set to the value sid1. If this is not the case, use the following command to change the characteristics of the RIM object:

```
wsetrim -s sid1
```

#### Note

You cannot change the Server ID parameter from the GUI, but only from the command line.

If everything is correct, you can scan the managed nodes in the TMR. To do the scan of a UNIX machine, refer to *TME10 Cookbook for AIX: Systems Management and Networking Applications*, SG24-4867.

---

## 4.8 Changing the RIM Host in the TMR

In this section we show you how to change the RIM host to any location on your TMR. The scenario components are shown in Figure 78 on page 100. For more details you can read 4.1.3, “Scenario 3: RIM Host Not on the TME Server” on page 65.

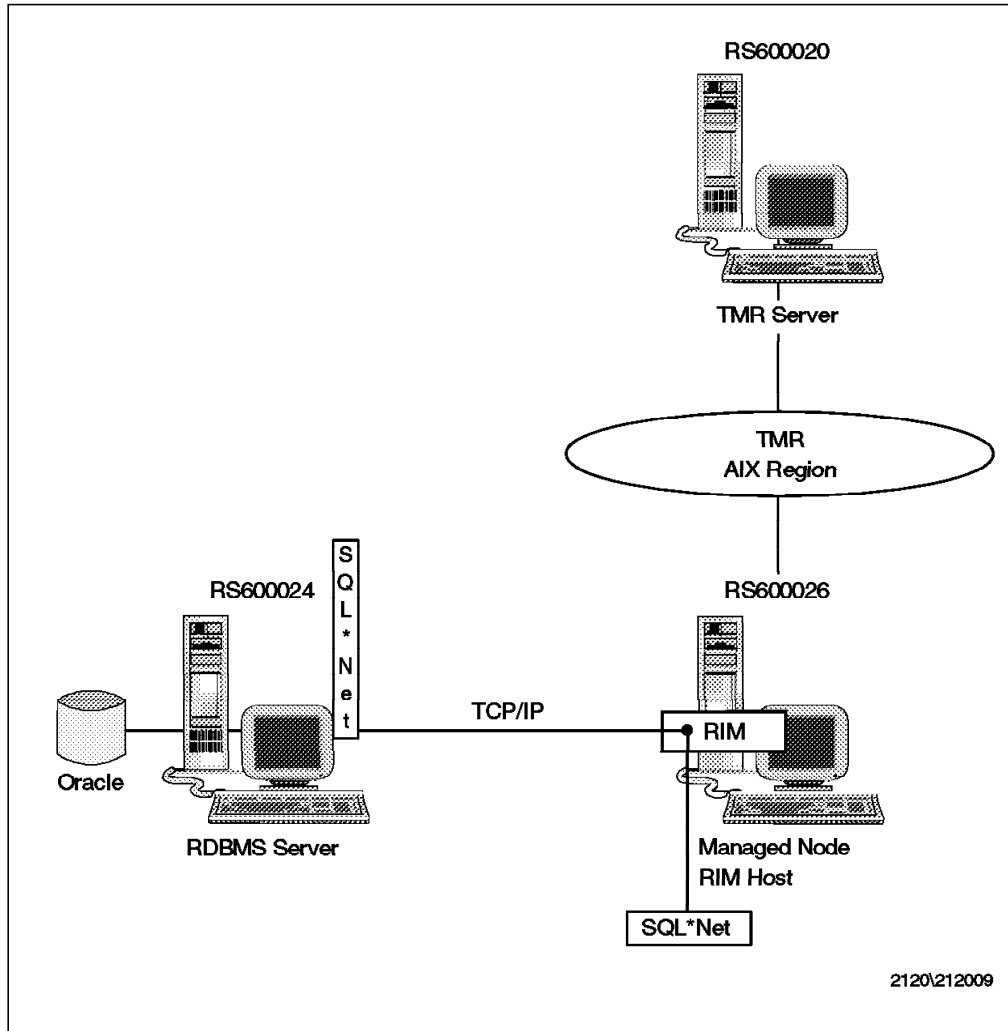


Figure 78. RIM Host on a Managed Node

For this scenario, we have installed:

- Oracle database server on rs600024
- SQL\*Net 2.x and its TCP/IP adapter on rs600024
- TME 10 Framework and TME 10 Inventory 3.1 on rs600026

The first step is to make sure that the managed node where you want to locate your RIM host (in our example, rs600026) has a TCP/IP connection with the RDBMS server (in our example, rs600024). To get the connection working, we need to install SQL\*Net 2.x on rs600026 following steps 1 to 11 in 4.3, “Installing Oracle for AIX” on page 70.

All the configuration needed for SQL\*Net is done on the machine where the database is installed (in our example, rs600024). To get SQL\*Net working on rs600026, we just need to copy the configuration files `tnsnames.ora` and `sqlnet.ora` from rs60024 and place them in the `/usr/local/oracle/network/admin` directory.

If you have further problems configuring SQL\*Net and Oracle, consult the *TME 10 Inventory 3.1 User's Guide*.



To test the SQL\*Net installation, on rs600026, type:

```
sqlplus tivoli/tivoli@sid1
```

The output should look like this:

```
SQL*Plus: Release 3.1.3.7.1 - Production on Mon Aug 4 17:35:13 1997
Copyright (c) Oracle Corporation 1979, 1994. All rights reserved.

Connected to:
Oracle7 Server Release 7.1.6.2.0 - Production Release
With the distributed and parallel query options
PL/SQL Release 2.1.6.2.0 - Production

SQL>
```

The following Perl script can help you to check whether Oracle and SQL\*Net are working properly. You need to create the script on the RIM host machine (in our example, rs600026). If you need more information about Perl, consult <http://www.perl.com> on the World Wide Web.

```

#!/usr/bin/perl

print "\nTesting SQL*NET connection between RS600026 and
 RS600024 as user tivoli ... \n";
print "executing: sqlplus tivoli/tivoli@sid1 \n";
$SQL=sqlplus tivoli/tivoli@sid1 << EOT
quit
EOT;
$ERROR=false;
if ($SQL = /invalid username/)
{
 $ERROR=true;
 print "
=====
 couldn't login as tivoli user, check the steps
 for creating the inventory user account ...
===== \n";
}
elsif (($SQL = /SQL> Disconnected/) && ($ERROR ne true))
{
 print "\nSQL*Net connection seems OK ... \n \n";
}
else
{
 $ERROR=true;
 print "
=====
 Error while connecting to SQL*Net listener
 Verify the configuration and location of tnsnames.ora
 and sqlnet.ora files
===== \n";
}
$TABLES=sqlplus tivoli/tivoli@sid1 << EOT
select * from all_catalog where owner=TIVOLI
quit
EOT;
unless ($ERROR eq true)
{
 print "Verifying that TIVOLI TABLES are included on the database... \n";
 if ($TABLES = /no rows selected/)
 {
 print "
=====
 An error occurred while searching Tivoli tables on the
 database, check the steps for installing the
 configuration repository database \n
===== \n";
 }
 else
 {
 print "\n \n Oracle seems to be working OK ... \n \n";
 }
}
}

```

Figure 79. Verify SQL\*Net Connection between RS600026 and RS600024

When all is working properly, the script output should look like this:

```
Testing SQL*NET connection between RS600026 and
RS600024 as user tivoli ...
executing: sqlplus tivoli/tivoli@sid1

SQL*Net connection seems OK ...
Verifying that TIVOLI TABLES are included on the database...

Oracle seems to be working OK ...
```

When the inventory account is not correctly created, the output should look like this:

```
Testing SQL*NET connection between RS600026 and
RS600024 as user tivoli ...
executing: sqlplus tivoli/tivoli@sid1

=====
 couldn't log in as tivoli user, check the steps
 for creating the inventory user account ...
=====
```

When a communication error occurs the output should look like this:

```
Testing SQL*NET connection between RS600026 and
RS600024 as user tivoli ...
executing: sqlplus tivoli/tivoli@sid1

=====
 Error while connecting to SQL*Net listener
 Verify the configuration and location of tnsnames.ora
 and sqlnet.ora files
=====
```

If the connection between your new RIM host (rs600026) and the RDBMS server (rs600024) is working properly, you can proceed now to configure the new location of the RIM host from Tivoli's point of view.

Although the wsetrim command allows you to set some RDBMS information about a RIM object, you can't change the RIM host label. To change the RIM host label you need to erase the inventory RIM object. On rs600026, type as user root:

```
wregister -ur RIM inventory
```

To create the new RIM object, type:

```
wcrtrim -v Oracle -h rs600026 -u tivoli -H /usr/local/oracle
-d rs600024 -s sid1 inventory
```

The flags mean:

| Flag | Purpose                                                                                                                                                               |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -v   | Sets the RDBMS vendor. You can choose Oracle or Sybase.                                                                                                               |
| -h   | Specifies the RIM host (in our example, rs600026).                                                                                                                    |
| -u   | Specifies the user that Tivoli will use to connect to the database. Use tivoli.                                                                                       |
| -H   | Specifies the database home directory. The value must be the absolute path for Oracle or Sybase on the RDBMS server (in our example, rs600024 and /usr/local/oracle). |
| -d   | Specifies the RDBMS server. You can use the FQDN or the alias name; for example we could have used sid1 instead of rs600024.                                          |
| -s   | Specifies the RDBMS alias name as configured on SQL*Net. The alias name is stored in the tnsnames.ora file.                                                           |

The command prompts you for the tivoli password. After you enter the password, you have to change the password in the script to test SQL\*Net. The syntax for sqlplus is user/password@RDBMS\_alias\_name. Find the line containing tivoli/tivoli@sid1 in the script and change it to tivoli/your\_password@sid1 according to the value you set in the wcrtrim command.

To check the RIM host configuration, type:

```
wgetrim inventory
```

The output should look like this:

```
RIM Host: rs600026
RDBMS User: tivoli
RDBMS Vendor: Oracle
Database ID: rs600024
Database Home: /usr/local/oracle
Server ID: sid1
```

#### Note

Be sure that the Server ID field contains the database alias that you have defined in the SQL\*Net configuration file, tnsnames.ora (in our example, sid1). Don't use the server's IP address or FQDN.

We now display the hardware profile from one managed node:

```
wviewmn rs600020
```

The command output should be:

```
Hardware System Id : 1693108358.1.327#TMF_ManagedNode
 ::Managed_Node
Hardware Architecture : power
Booted Operating System Name : AIX
Booted Operating System Version : 4.2
Processor Model : UNKNOWN
Physical Memory (kB) : 131072
Paging Space (kB) : 163840
Time of Last Scan : 1997-08-04 14:26:16
```



## Chapter 5. Examples Using Asset Management with TME 10 Inventory

In this chapter we show two, rather simple, examples of using asset management in combination with TME 10 Inventory.

In the first example, we use Lotus 1-2-3 to access data stored in the RDBMS system that TME 10 Inventory uses and then show some examples how this data can be used in combination with data stored in the spreadsheet. For the first example, you should be familiar with the basic principles of using a spreadsheet application, such as Lotus 1-2-3.

In the second example, we show a simple license management example with the TME 10 Inventory RDBMS. This example also uses TME 10 Enterprise Console in combination with TME 10 Inventory. For the second example, you should be familiar with basic principles of TME 10 Enterprise Console and RDBMS systems.

### 5.1 Using TME 10 Inventory with Lotus 1-2-3

In this scenario we want to achieve two goals:

- Link Lotus 1-2-3 to the TME 10 Inventory database.
- Show examples for using Lotus 1-2-3 to process TME 10 Inventory data.

The following figure gives an overview:

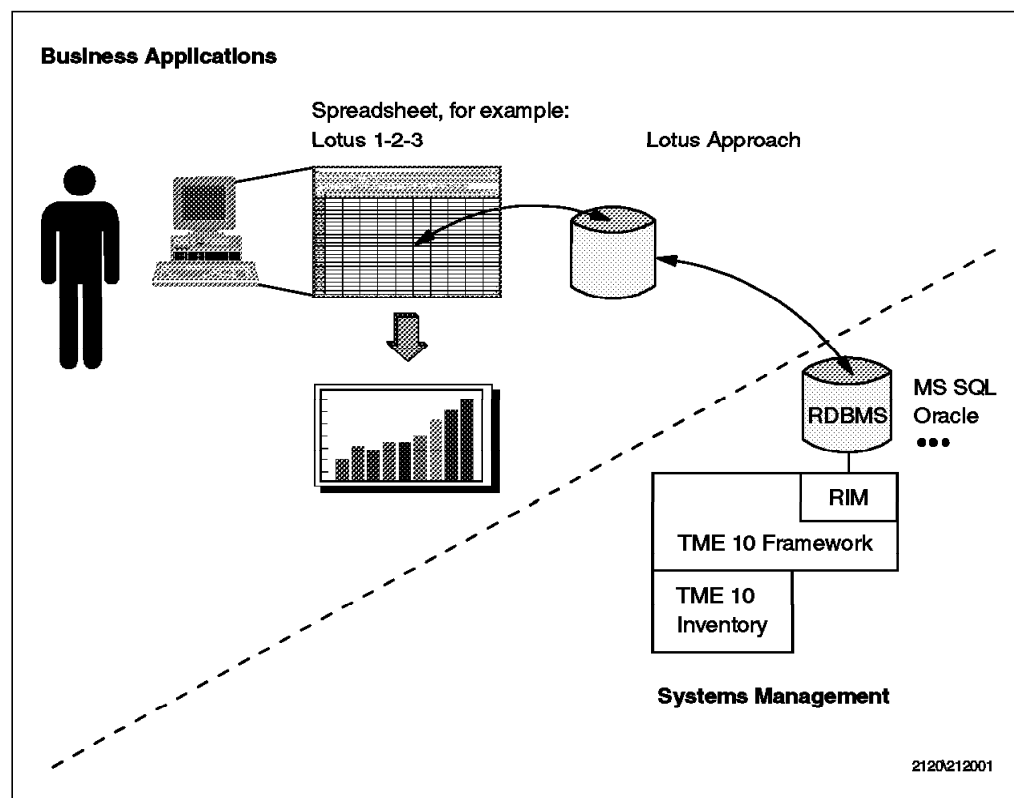


Figure 80. Using Lotus 1-2-3 with TME 10 Inventory

We link Lotus 1-2-3 to the TME 10 Inventory database through Lotus Approach. For that purpose, we will re-use the scenario set up in Chapter 2, "Using TME 10 Inventory with Sybase on Windows NT" on page 13 where victor acts as the RDBMS server. We install Lotus 1-2-3 and Lotus Approach on victor and access the TME 10 Inventory database.

### 5.1.1 Installing Lotus 1-2-3 and Lotus Approach

We install Lotus 1-2-3 and Lotus Approach on victor which acts as the RDBMS server for TME 10 Inventory in the foxtrot TMR (see Chapter 2, "Using TME 10 Inventory with Sybase on Windows NT" on page 13 for details).

To install these products we simply insert the Lotus SmartSuite 97 CD-ROM into the CD-ROM drive of victor (Windows NT 4.0). After a short while the following window will appear:

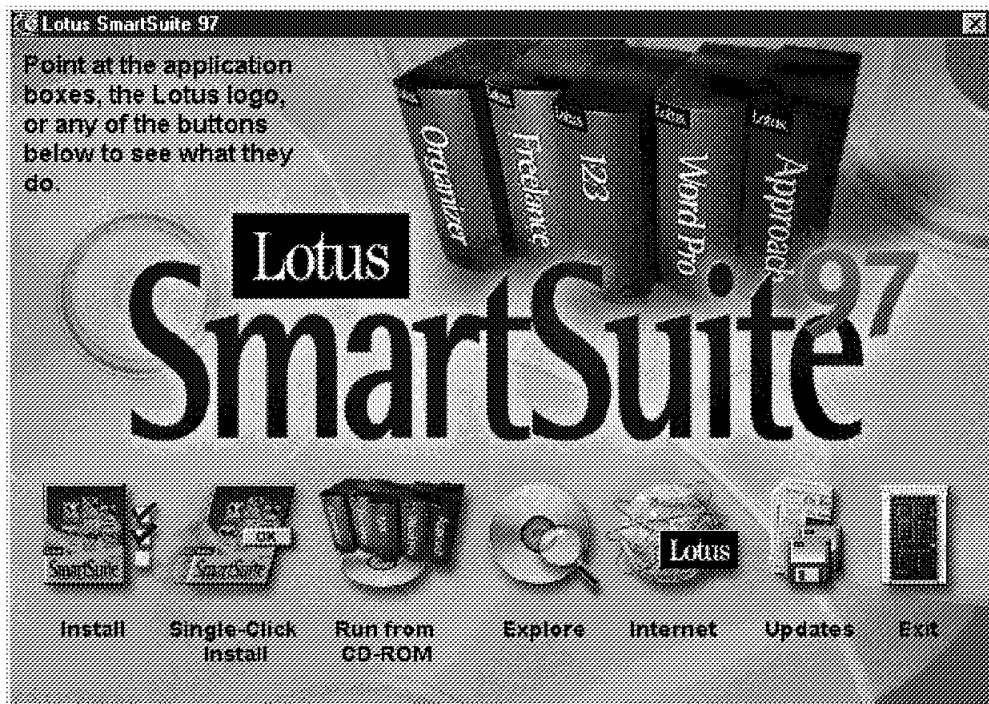


Figure 81. Lotus SmartSuite 97 Window

We click on **Install**. This will start the installation procedure. After a while the Select Lotus SmartSuite Applications window will pop up as shown in the following figure:





Figure 82. Select Lotus SmartSuite Applications

We select the check boxes next to 1-2-3 and Approach and then click the **Next>** button.

During the installation we accept all defaults. After a short period of time Lotus 1-2-3 and Lotus Approach will be installed on victor.

### 5.1.2 Accessing TME 10 Inventory Data from Lotus 1-2-3

In order to access TME 10 Inventory data from victor the only thing needed for Lotus 1-2-3 is access to a Sybase database server or client.

As victor has been installed as a Sybase server in Chapter 2, "Using TME 10 Inventory with Sybase on Windows NT" on page 13 we don't need to take any further steps.

#### Note

If we installed Lotus 1-2-3 and Lotus Approach on a node that has no component of Sybase installed, we at least need to install a Sybase client in addition. Refer to Chapter 2, "Using TME 10 Inventory with Sybase on Windows NT" on page 13 on how to do that.

Access to Sybase from Lotus 1-2-3 is achieved by using external database references. We now show a simple example of how to do that.

In our example we started Lotus 1-2-3 and created a new, empty worksheet as shown in the following figure:

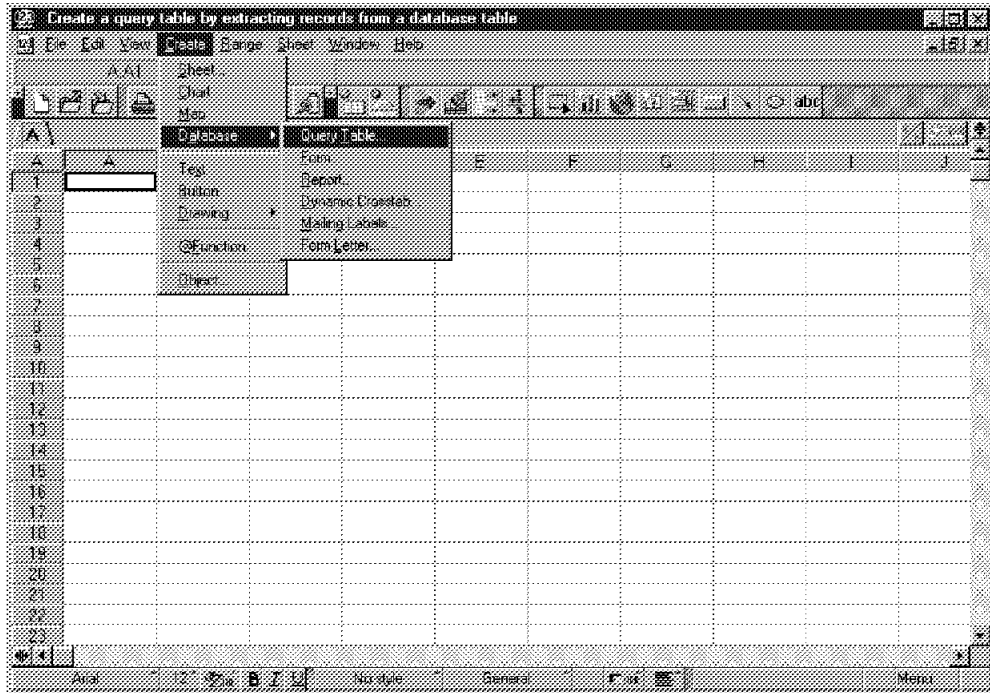


Figure 83. Lotus 1-2-3 Worksheet

To access a Sybase table we select **Create** from the menu bar, then **Database** from the pull-down menu and **Query Table...** from the second level menu.

The following window will pop up:

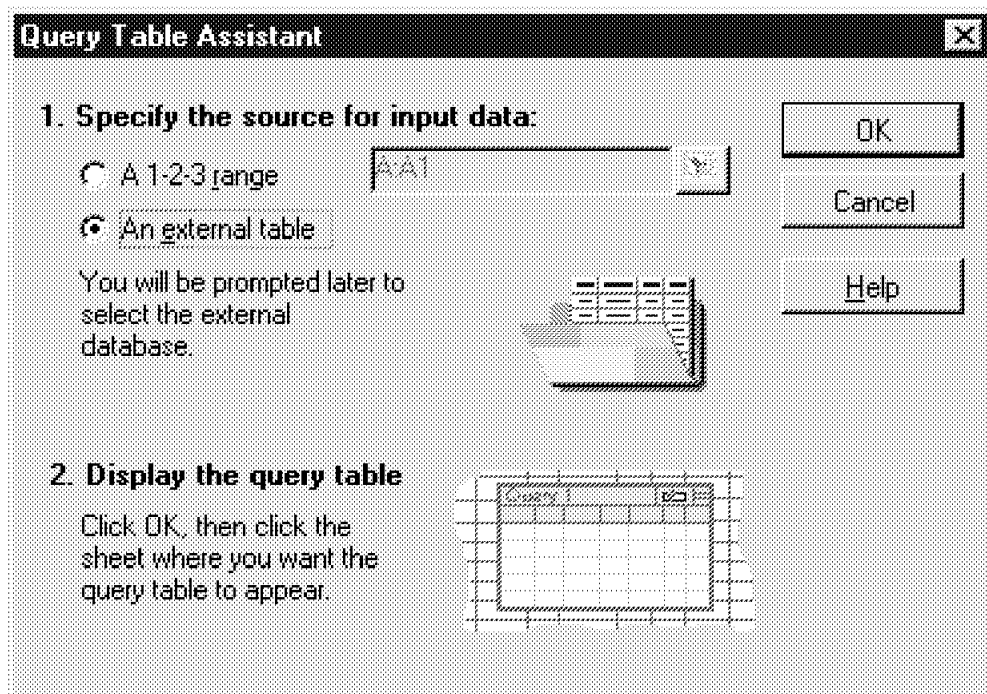


Figure 84. Query Table Assistant Window

We select the **An external table** radio button and then click the **OK** button.

The following window will pop up:

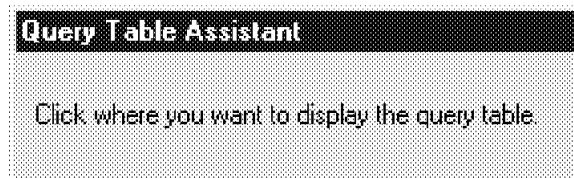


Figure 85. Query Table Positioning Pop-Up

We just click randomly on the worksheet.

The following window will pop up:

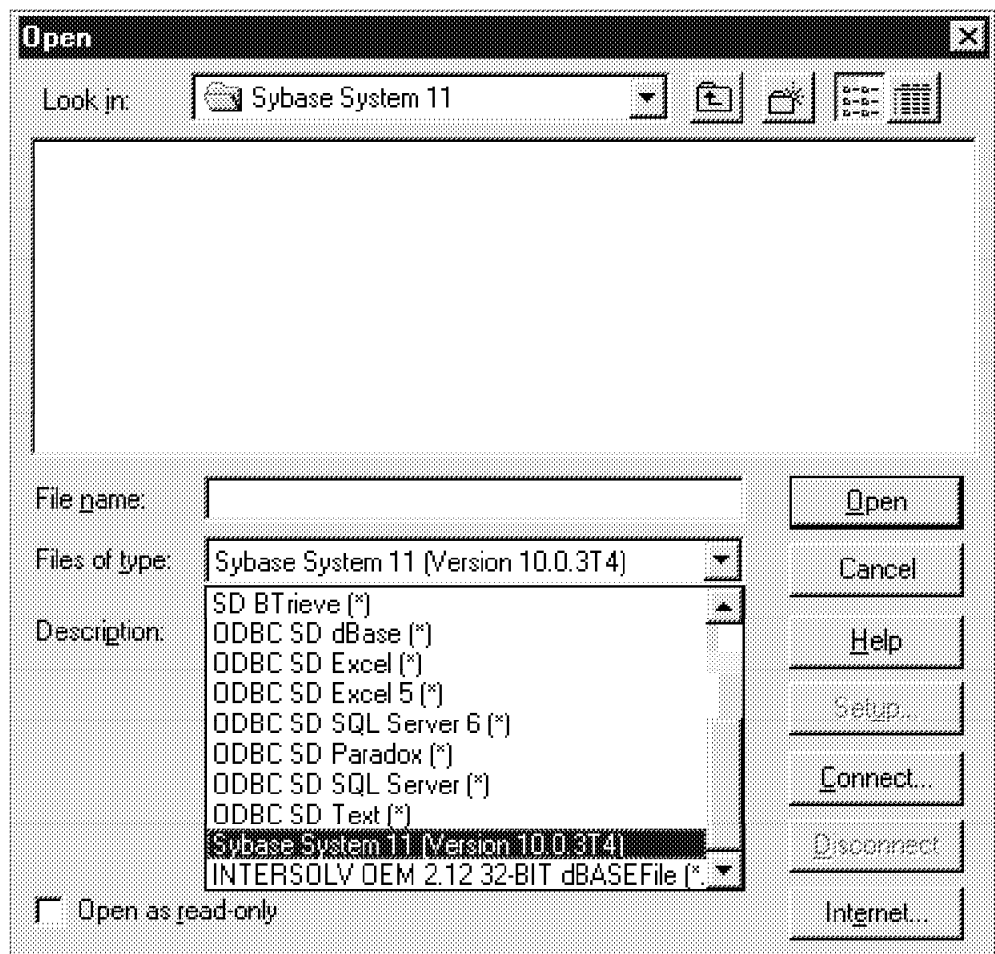


Figure 86. Open Window

In the Files of type list box we select **Sybase System 11 (Version 10.0.3T4)** and then click on the **Open** button.

Lotus 1-2-3 will then try to establish a connection to a Sybase RDBMS server. In order to do so, we need to specify some connection information; hence, the following window will pop up.



Figure 87. Logon to System 10 Window

**Note**

Although the above window mentions Sybase System 10 and we are actually using System 11 we did not experience any problems accessing the database.

We enter VICTOR as the Server Name. This is the database server we installed and configured in Chapter 2, "Using TME 10 Inventory with Sybase on Windows NT" on page 13.

In the example we use tivoli as the Login ID, as we are going to query TME 10 Inventory tables. We also specify the password for the database user and select inventory as the Database to connect to.

Then we click the **OK** button in the Logon to System 10 window, followed by a click on the **Open** button in the Open window.

Lotus 1-2-3 will then try to establish a connection with the RBDMS server.

After a short while, the databases available will be shown in the top section of the window. We double-click on the folders until we get down to the TME 10 Inventory database tables as shown in the following figure:

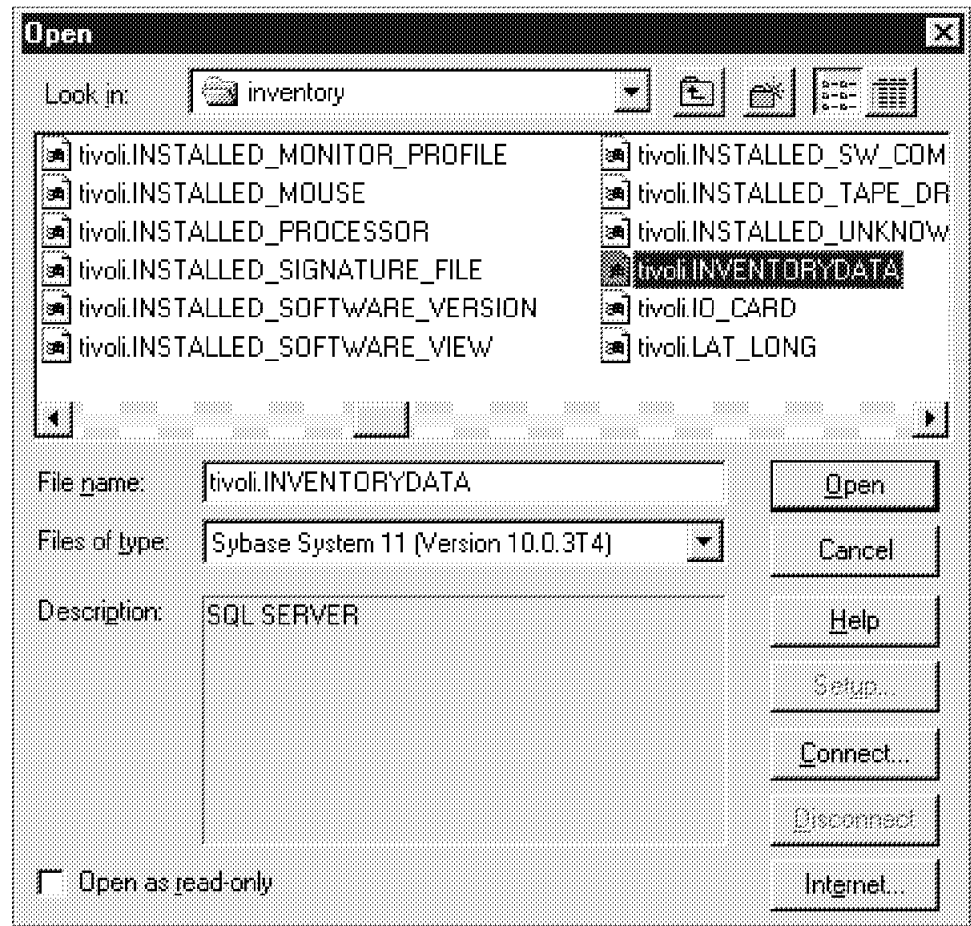


Figure 88. Viewing TME 10 Inventory Tables in Lotus 1-2-3

We scroll right until we find **tivoli.INVENTORYDATA**. This is not actually a table, but one of the views that combines data from multiple tables in the TME 10 Inventory database.

We select **tivoli.INVENTORYDATA** and click the **OK** button and the following window will pop up:

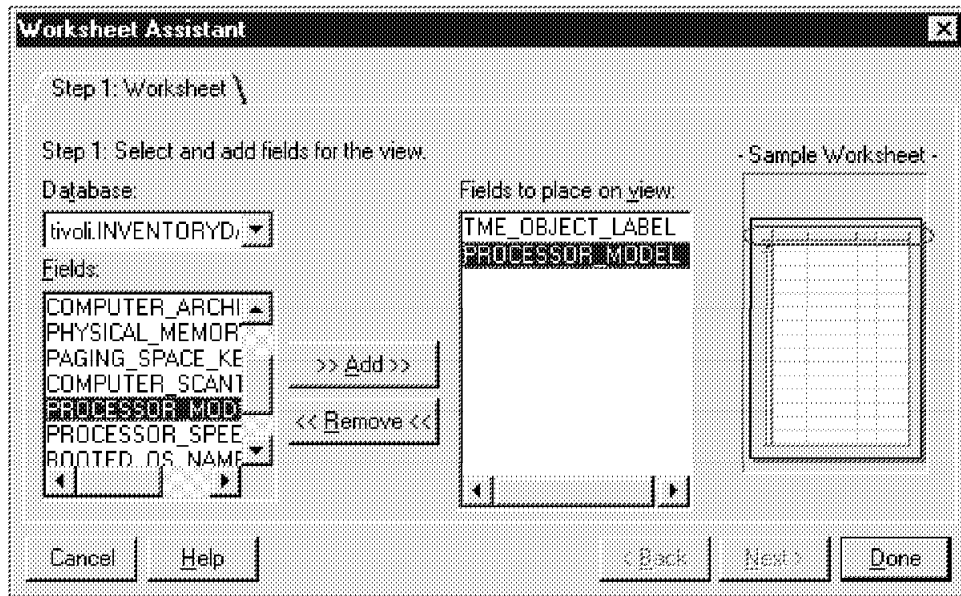


Figure 89. Selecting Columns from INVENTORYDATA View to Bring to Lotus 1-2-3

We now select which columns from the INVENTORYDATA view we want to bring to our Lotus 1-2-3 worksheet. In this example, we select TME\_OBJECT\_LABEL and PROCESSOR\_MODEL, one at a time, and click the **Add** button to copy the field names from the Fields list to the Fields to place on view list of the Worksheet Assistant window.

After having selected the fields we want, we click the **Done** button.

The actual database query will now be executed, and after a while the result will show up on our worksheet in Query Table 1 as shown in the following figure:

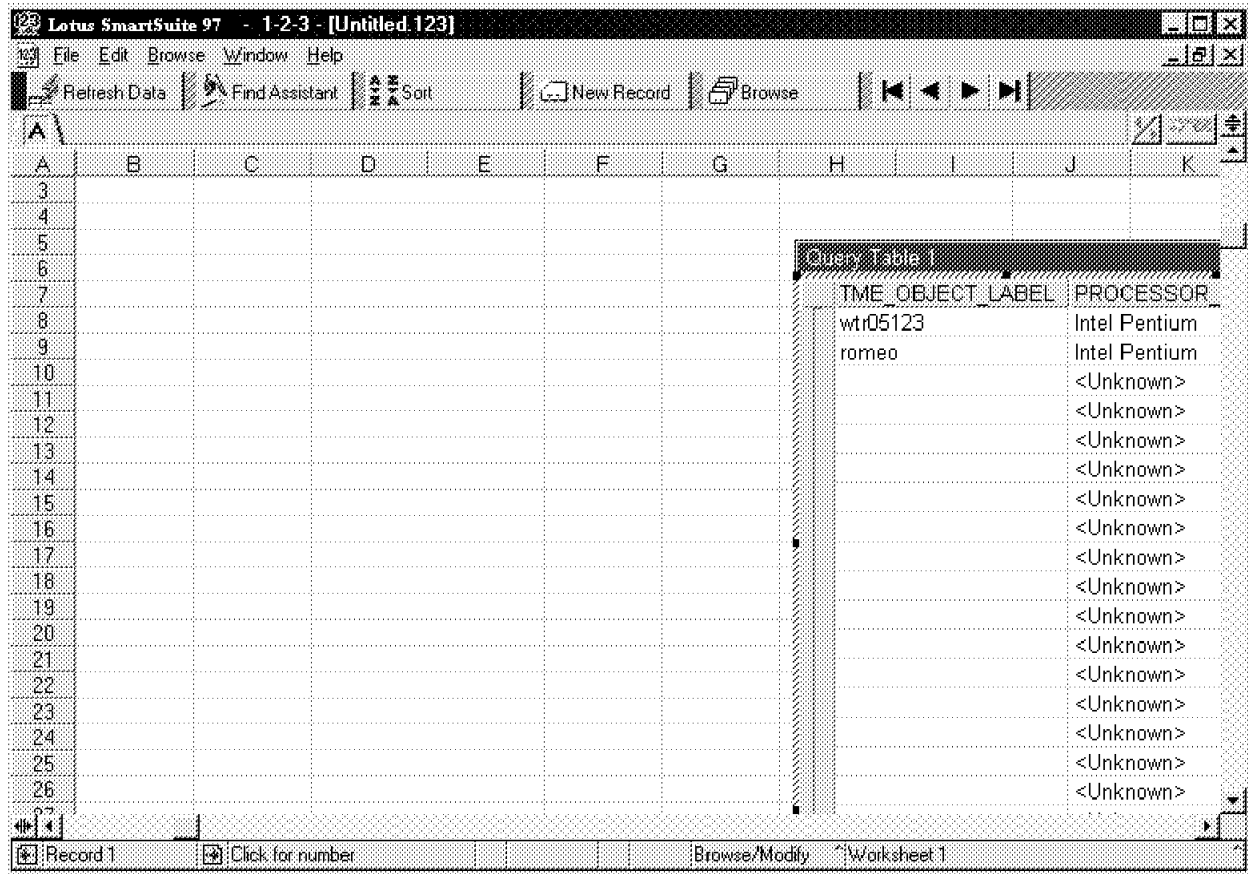


Figure 90. Query Table in a Lotus 1-2-3 Worksheet

Query Table 1 cannot be used directly by Lotus 1-2-3 so we need to copy the contents to a position on the worksheet where we can use it for calculations. We select the whole Query Table 1 (both columns, all rows), and then point anywhere on the Query Table and use the right mouse button to copy it all to the clipboard. We then click on the upper left corner of the area on the worksheet where we want to place the data (in our example we click on B6).

The following window will pop up:

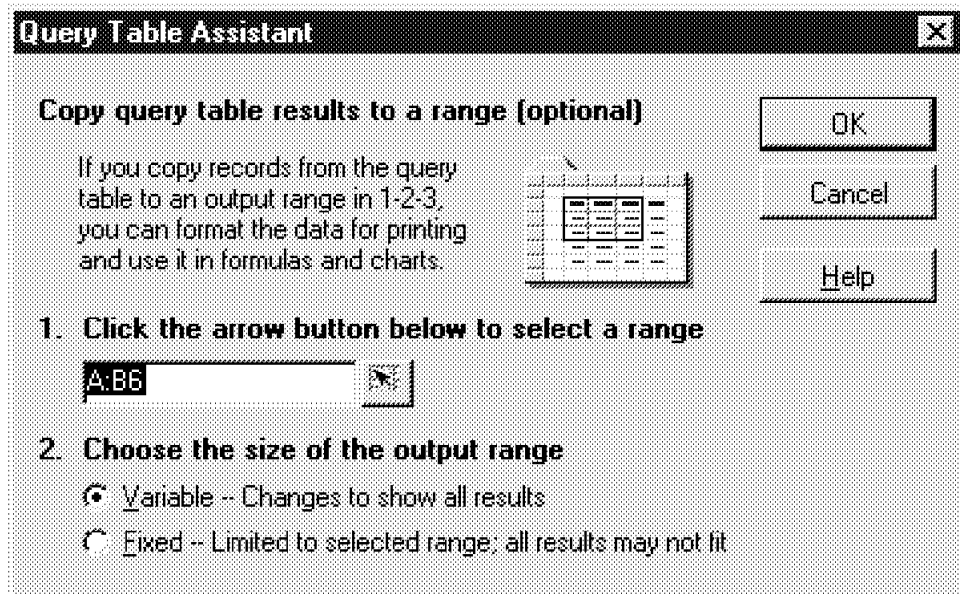


Figure 91. Options when Copying Query Table to a Lotus 1-2-3 Worksheet

We keep the defaults as shown, so the area will be extended to show all columns and rows. We then click **OK**, and our data is ready to be used for calculations.

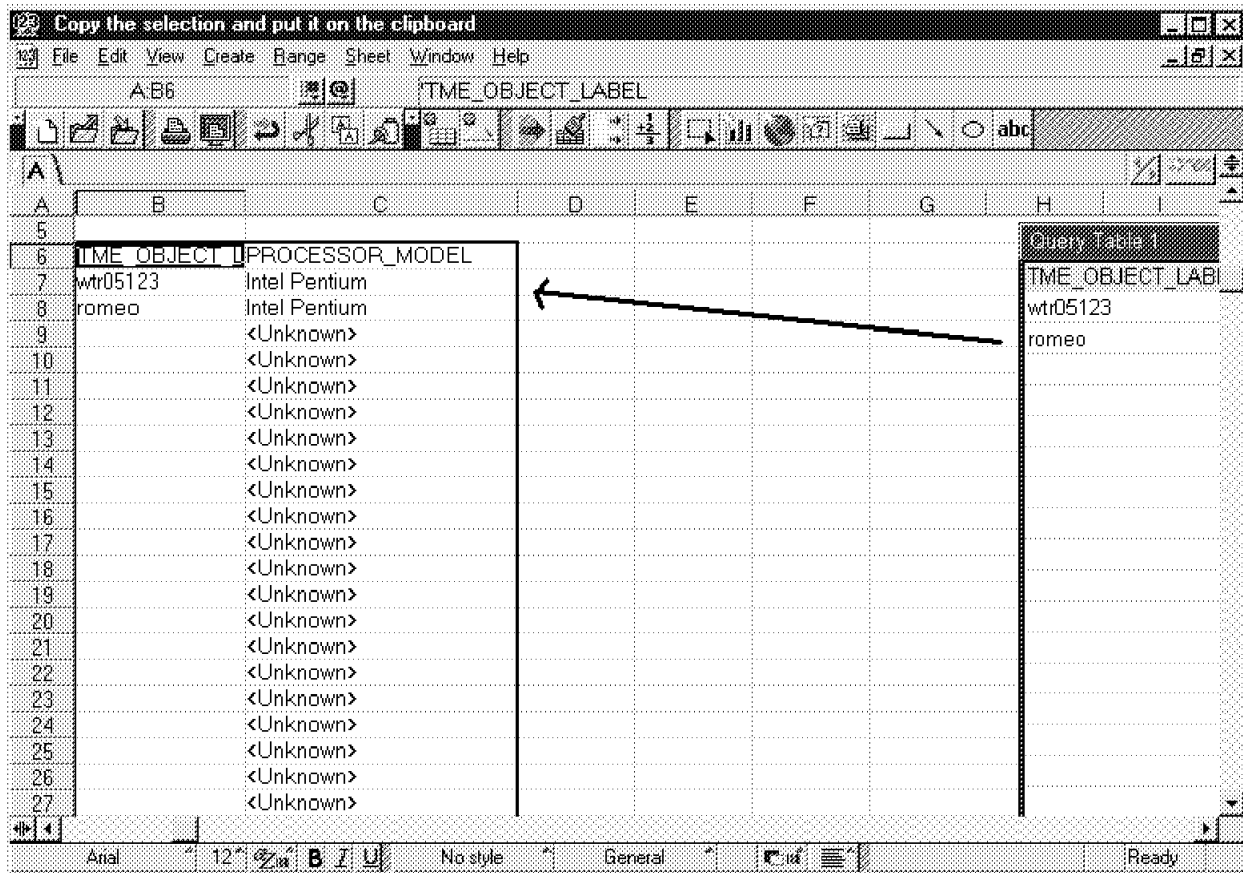


Figure 92. Having Copied Data from Query Table to Lotus 1-2-3 Worksheet



The Query Table 1 can now be deleted. It was only used for transferring the query data from the database to the worksheet.

**Note**

In our example, we deleted the Query Table, but it might be useful to keep it, if we wanted to make a new query to retrieve updated data from the same database table or view.

In that case we would point at the Query Table, use the right mouse button, and select **Refresh Data**. That would result in a new query. The new data would be made available in the Query Table, and we could then copy any part of it into the worksheet as shown in the previous step.

### 5.1.3 An Example Using TME 10 Inventory Data with Lotus 1-2-3

What we want to do in this example is to get an overview of all processors installed in our inventory of scanned computers, and to calculate the total value of the processors. To do this, we need to know all processor types that have been scanned into our TME 10 Inventory database, and we need to assign a value to the different processor types.

So, we repeat the previous steps where we imported data from the INVENTORYDATA view, but this time we want to import data from the PROCESSOR table. All the steps are not shown again, only the step to select what columns to import from the PROCESSOR table:

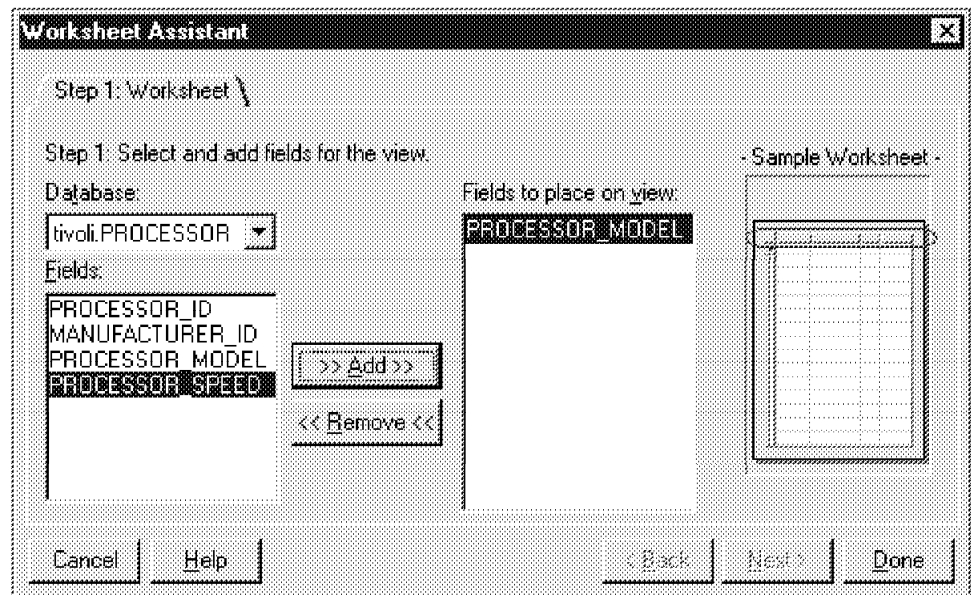


Figure 93. Selecting Columns from the PROCESSOR Table to Bring to Lotus 1-2-3

We only want the PROCESSOR\_MODEL column. After having selected the column, we click the **Done** button and data from the database query will again be placed in a Query Table on the worksheet. As in the previous steps, we then copy the data from Query Table 1 to an area on the worksheet where we can use it for calculations. The result after this second query and copy is shown in the following figure:

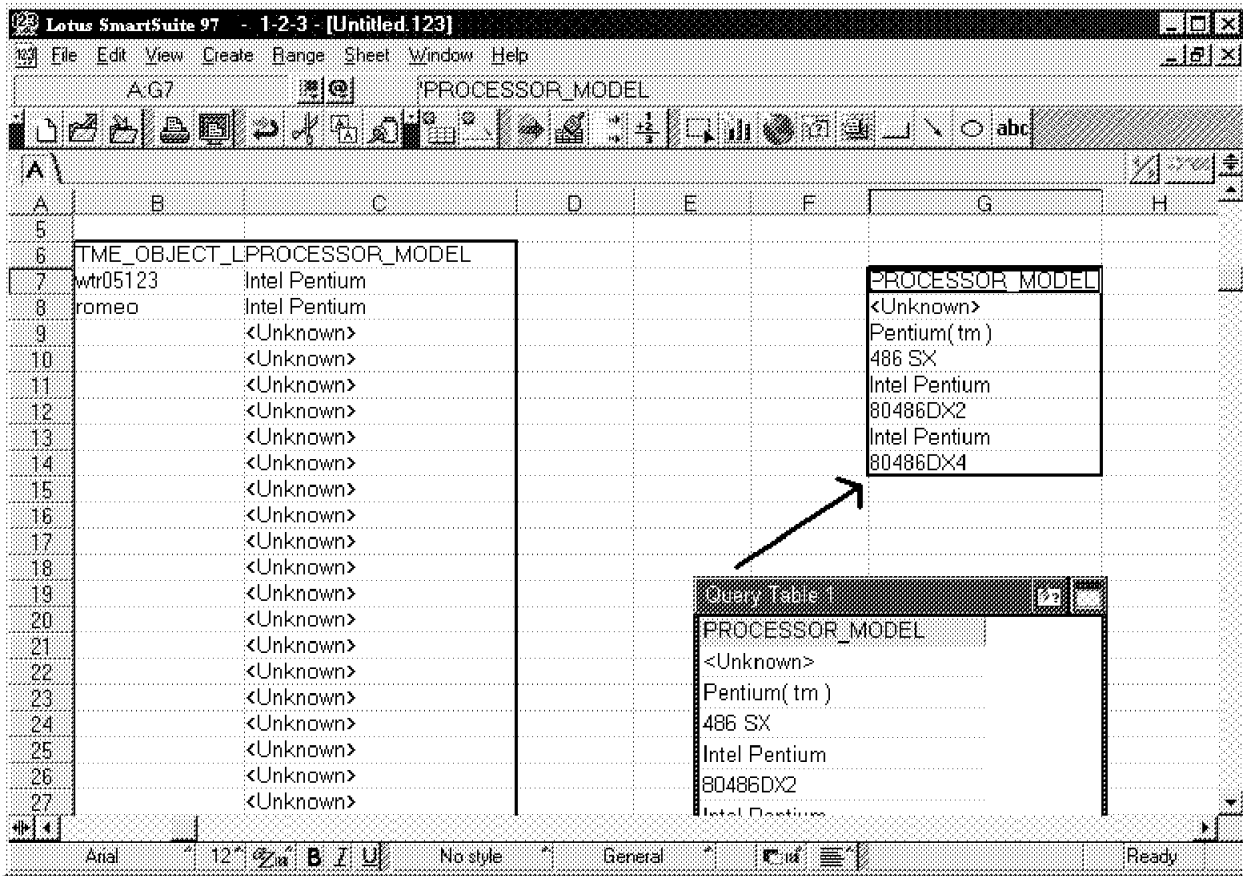


Figure 94. More Data from Query Table to Lotus 1-2-3 Worksheet

We now assign values to the various processor models. We manually enter these values on the worksheet in column H as shown in Figure 95 on page 119.

We then want to assign these processor values to the corresponding machines in the table on the left-hand side. In our example, we used a nested Lotus 1-2-3 function @IF as shown in the white area in the upper part of Figure 95 on page 119. The formula is:

```
@IF(C7=G8,H8,@IF(C7=G9,H9,@IF(C7=G10,H10,@IF(C7=G11,H11,@IF(C7=G12,H12,@IF(C7=G13,H13,@IF(C7=G14,H14,0))))))
```

We define this function for the cell D7, and then copy it to the cells below (D8-D36):

**Note**

Instead of the nested @IF function there might be a better way, for example a function or a macro, to select a value from a table, and assign that value to another cell.

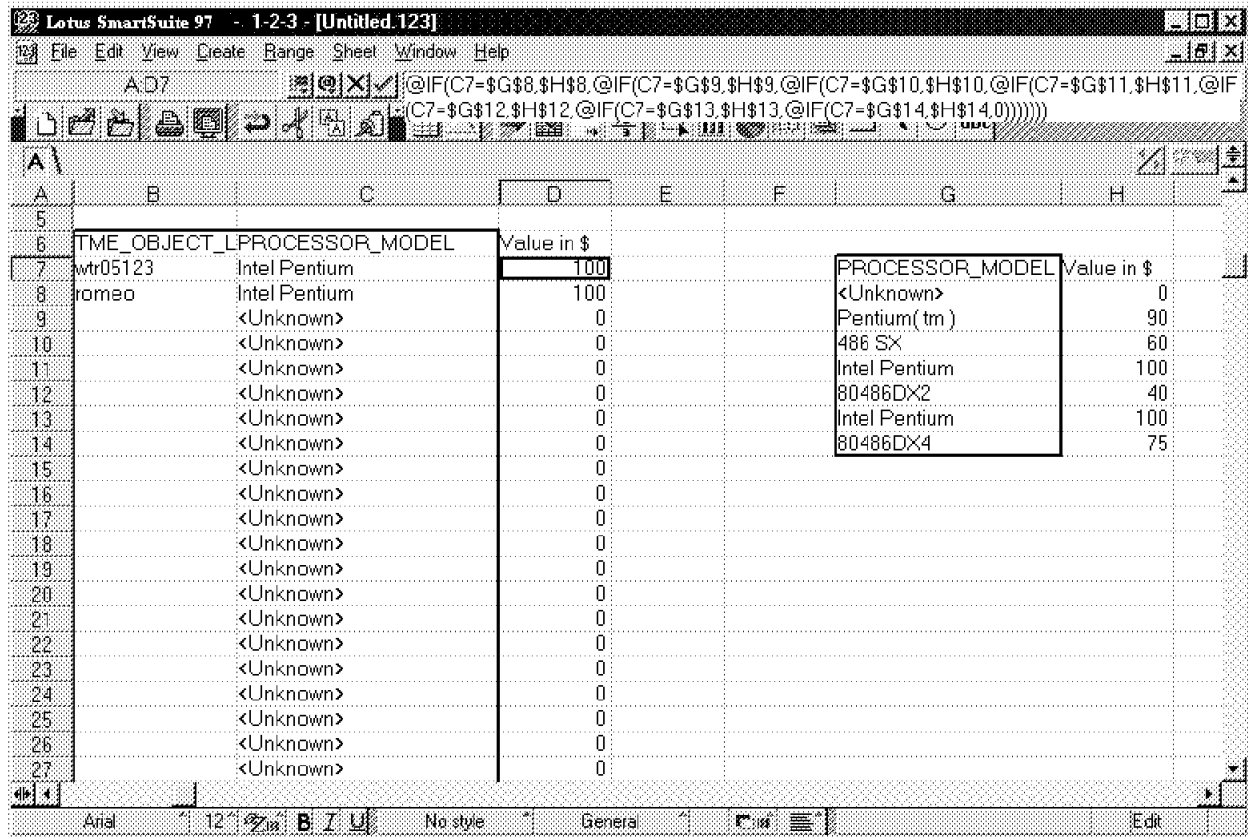


Figure 95. Using Lotus 1-2-3 Function to Assign Asset Value

Of course we can also use the basic @SUM function of Lotus 1-2-3 to show the total value of all processors. This can be seen at the bottom of Figure 96 on page 120 in cell D38.

Another feature of Lotus 1-2-3 is to present data graphically. We select column B "TME\_OBJECT\_ID" (machine name) and column D "Value in \$". To create the graph, we click the graph icon on the tools bar at the top, followed by a click on the position in the worksheet where we want to place the graph as shown in the following figure:

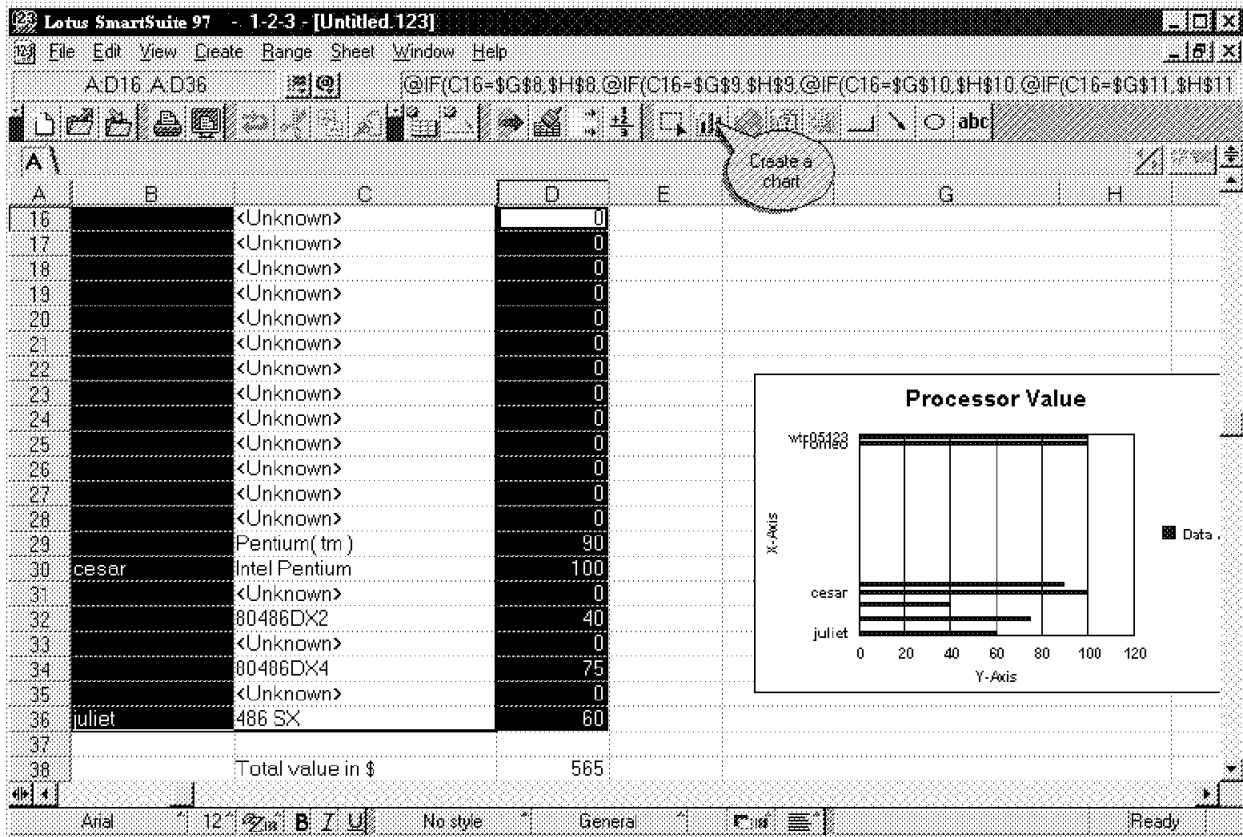


Figure 96. TME 10 Inventory Data Shown Graphically in Lotus 1-2-3 Worksheet

### 5.1.4 Summary

With the previous example we have shown how to use Lotus 1-2-3 and TME 10 Inventory to get a simple overview of the installed processors in the computers in our network.

In a real customer environment similar procedures could be used for example to:

- Update the book value for any kind of equipment scanned by TME 10 Inventory, like for example computer models, hard disks, processors, etc.
- Retrieve information about installed software, so it can be matched against actual licences for the software.
- Create graphical reports in order to get an overview of the assets.

## 5.2 Using TME 10 Inventory with License Management and TME 10 Enterprise Console

In this scenario, we want to show how to use the TME 10 Inventory database contents together with the TME 10 Enterprise Console to generate an alert to the license administrator of unauthorized usage of licensed programs.

The following figure gives an overview:

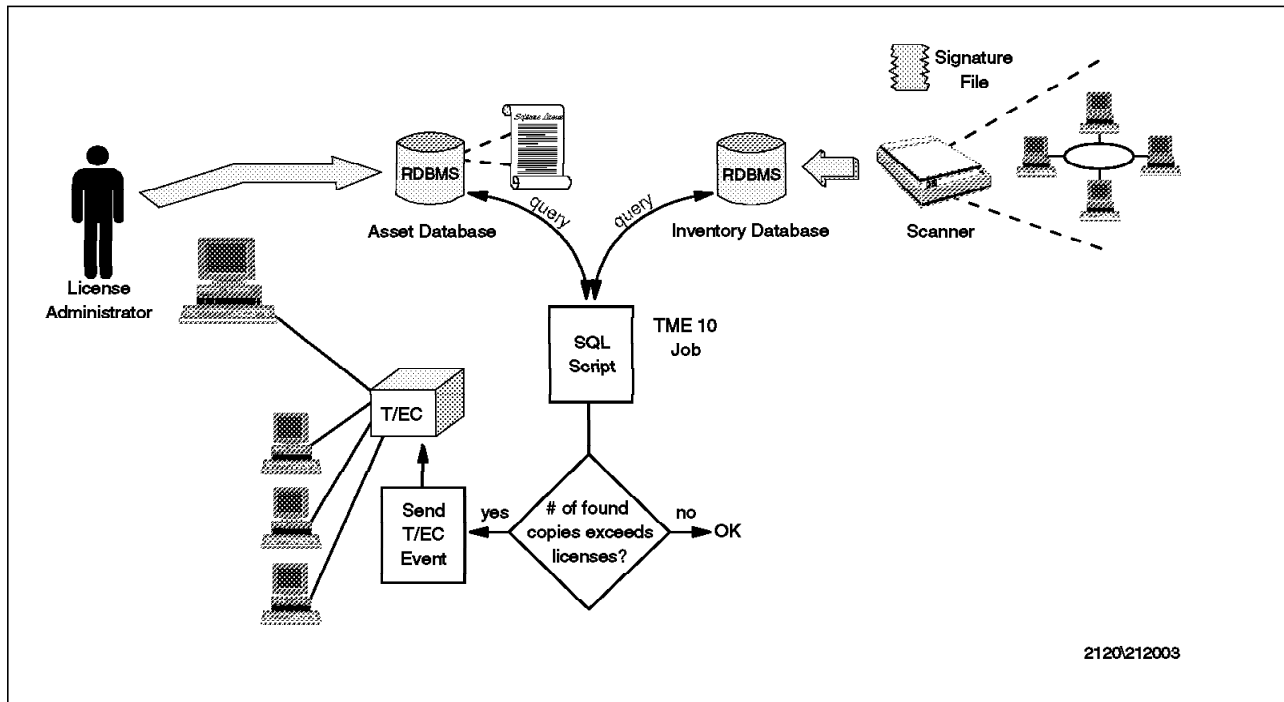


Figure 97. Using TME 10 Enterprise Console with TME 10 Inventory for Checking Licences

Information about available licenses is inserted into the asset database. A simple script will assist the administrator to create and insert information into the asset database. Another script will compare the TME 10 Inventory database and the asset database, and if too many licenses are being used, an event will be sent to the license administrator's T/EC console.

To set up this automatic License Checking Application, the following has to be done:

- Configure signature files in TME 10 Inventory profile.
- Create a license table (or asset database).
- Update the license table.
- Run a script to send a T/EC event if the number of programs found exceeds the number of licenses.
- Configure TME 10 Enterprise Console to recognize the new type of event.
- Schedule a TME 10 job periodically that saves the output from the compare script.

## 5.2.1 Configure Signature Files

First, we add a signature file, so our own program will be discovered by TME 10 Inventory. There are a few things to be aware of to get this to work:

- The file name of the signature file must be entered in capital letters.
- The file name extension must be included in the list of File Extensions.
- The size of the signature file must be exactly its actual length, and entered with at least 3 digits, for example, 078.
- If the size is more than 3 digits, a comma or full stop must not be entered. Just enter for example, 37313 (not 37,313 or 37.313).

The following figure shows the properties of our inventory profile.

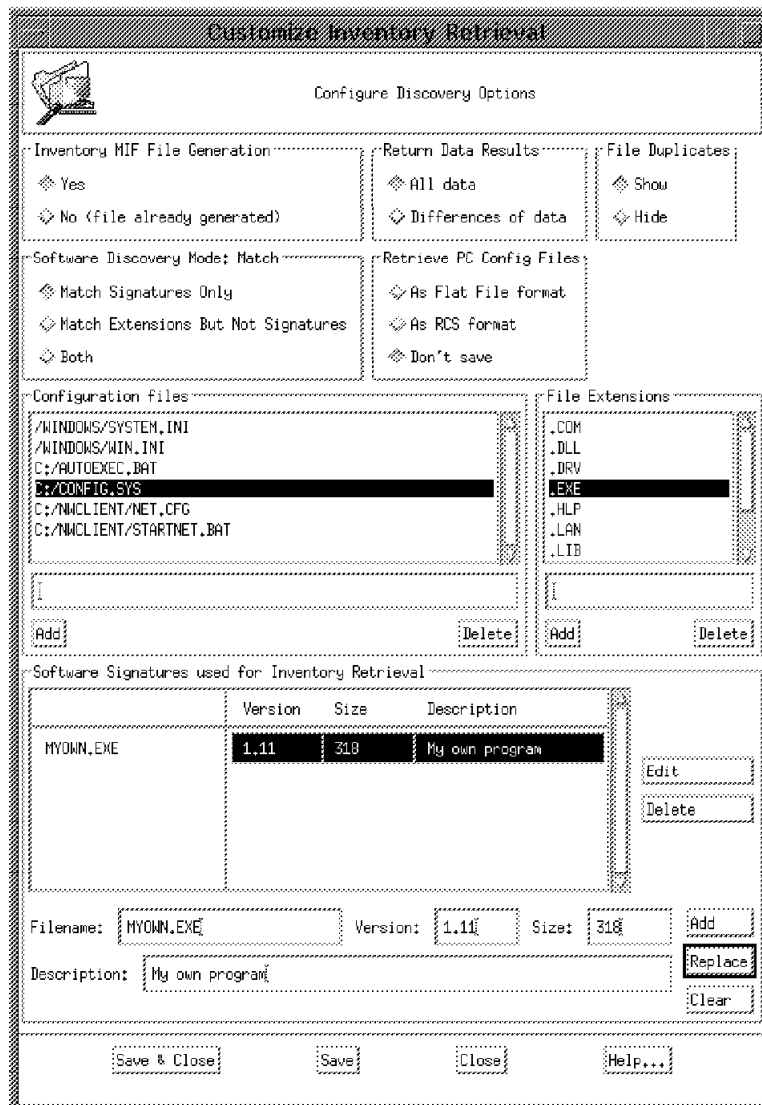


Figure 98. Adding Signature File to Inventory Profile

We enter the filename MYOWN.EXE, the size 318, and a description for our program. Many standard applications are discovered by default, but any new standard application can be added to the profile as we did with our program.

## 5.2.2 Create Asset Database

We do not really create a new database in our example. We create a new table within the existing TME 10 Inventory database and call this new table SW\_LICENSES. We create it by executing the script swlic\_cre.sh, shown in the following figure, that selects two columns of data from the table INSTALLED\_SOFTWARE\_VERSION.

```
#!/bin/ksh

This script creates the new table SW_LICENSES,
retrieves data from table INSTALLED_SOFTWARE_VERSION,
and inserts this data in the table SW_LICENSES

SYB_USER=tivoli
SYB_PASSWD=tivoli
WRKFILE=/usr/local/Tivoli/localtools/wrk1

#
First create the new table SW_LICENSES
#

isql -U $SYB_USER -P $SYB_PASSWD <<EOF

drop table SW_LICENSES
go

create table SW_LICENSES (
 SOFTWARE_ID varchar(64) NOT NULL,
 SOFTWARE_VERSION_ID varchar(32) NOT NULL,
 LICENSES_ALLOWED int NULL,
 LICENSES_USED int NULL
)

go

alter table SW_LICENSES
 ADD PRIMARY KEY (SOFTWARE_ID, SOFTWARE_VERSION_ID)
go

EOF

Next select from the table INSTALLED_SOFTWARE_VERSION
and put one unique line per program in a workfile

isql -U $SYB_USER -P $SYB_PASSWD -o $WRKFILE -w 100 <<EOF

select DISTINCT
 SOFTWARE_ID, SOFTWARE_VERSION_ID
from INSTALLED_SOFTWARE_VERSION
go

EOF
```

Figure 99. Script to Create SW\_LICENSES ( Part 1 of 2 )

```

#
Now insert lines from the workfile into new table SW_LICENSES
#

IFS=\t
while read -r LINE
do
 IFS=
 # But remove header and trailer produced by SYBASE output
 if [[echo $LINE | cut -c 1-12 = " SOFTWARE_ID" || \
 echo $LINE | cut -c 1-12 = " -----" || \
 echo $LINE | cut -c 1-12 = " " || \
 echo $LINE | cut -c 1 = "("]]
 then continue
 else
 PROG=echo $LINE | cut -c 2-65
 VER=echo $LINE | cut -c 67-98

 isql -U $SYB_USER -P $SYB_PASSWD -o /dev/null <<EOF

 insert into SW_LICENSES
 (SOFTWARE_ID, SOFTWARE_VERSION_ID)
 values('$PROG', '$VER')
 go
 commit
 EOF
 fi
 done <$WRKFILE

 rm $WRKFILE
 exit

```

Figure 100. Script to Create SW\_LICENSES ( Part 2 of 2 )

The script creates a new table SW\_LICENSES. It then reads the columns SOFTWARE\_ID and SOFTWARE\_VERSION\_ID from the existing TME 10 Inventory table INSTALLED\_SOFTWARE\_VERSION and writes them to a workfile. Next, these two columns are inserted in the table SW\_LICENSES. This SW\_LICENSES table will then contain one row for each application found by TME 10 Inventory in our managed environment.

Our newly created table SW\_LICENSES has four columns:

| Column name         | Length     | Source                                            |
|---------------------|------------|---------------------------------------------------|
| SOFTWARE_ID         | varchar 64 | Populated from table INSTALLED_SOFTWARE_VERSION   |
| SOFTWARE_VERSION_ID | varchar 32 | Populated from table INSTALLED_SOFTWARE_VERSION   |
| LICENSES_ALLOWED    | int        | Populated by manual input via script swlic_upd.sh |
| LICENSES_USED       | int        | Populated automatically by script swlic_cmp.sh    |



The next figure shows the contents of table SW\_LICENSES after we created it using the script swlic\_cre.sh:

| SOFTWARE_ID                            | SOFTWARE_VERSION_ID | LICENSES_ALLOWED | LICENSES_USED |
|----------------------------------------|---------------------|------------------|---------------|
| Microsoft DOS Command Processor        | 7.0                 | NULL             | NULL          |
| Microsoft DOS XMS Driver               | 3.95                | NULL             | NULL          |
| Microsoft Exchange                     | 1.0                 | NULL             | NULL          |
| Microsoft Exchange EXE                 | 4.00.835.1374       | NULL             | NULL          |
| Microsoft Network Client               | 4.2                 | NULL             | NULL          |
| Microsoft Visual Basic runtime library | 3.0                 | NULL             | NULL          |
| Microsoft Windows 95                   | 4.00.950            | NULL             | NULL          |
| My own program                         | 1.11                | NULL             | NULL          |
| Netscape Navigator application file    | 3.01                | NULL             | NULL          |

Figure 101. Contents of Table SW\_LICENSES

### 5.2.3 Update Licenses in the License Table

After having created the SW\_LICENSES table we must enter values in its column LICENSES\_ALLOWED. To make it easy to do this update, we write a script swlic\_upd.sh shown in the following figure:

```
#!/bin/ksh

This script reads the table SW_LICENSES,
and updates column LICENSES_ALLOWED
with manually entered values.

DSQUERY=SYBASE already set in calling environment
SYBASE=/usr/local/sybase already set in calling environment
PATH=$SYBASE/bin already set in calling environment

SYB_USER=tivoli
SYB_PASSWD=tivoli
WRKFILE=/usr/local/Tivoli/localtools/wrk2

#
Create a temporary file from table SW_LICENSES
#
```

Figure 102. Script to Update SW\_LICENSES ( Part 1 of 2 )

```

isql -U $SYB_USER -P $SYB_PASSWD -o $WRKFILE -w 115<<EOF

select SOFTWARE_ID, SOFTWARE_VERSION_ID, LICENSES_ALLOWED
 from SW_LICENSES
go

EOF

#
Now display programs one by one and ask for input
to update the column LICENSES_ALLOWED

exec 3< $WRKFILE
IFS=\t
while read -u3 LINE
do
 IFS=
 # But remove header trailer and blank lines produced by SYBASE output
 if [[echo $LINE | cut -c 1-12 = " SOFTWARE_ID" || \
 echo $LINE | cut -c 1-12 = " -----" || \
 echo $LINE | cut -c 1-12 = " " || \
 echo $LINE | cut -c 1 = "("]]
 then continue
 else

 PROG=echo $LINE | cut -c 2-65
 VER=echo $LINE | cut -c 67-98
 VE=echo $LINE | cut -c 67-78
 LIC=echo $LINE | cut -c 100-115
 print ""
 print "Program: $PROG \nVersion: $VER"
 print "Is currently registered for $LIC licenses."
 print -n "Hit Enter to keep current value, or enter new value: "
 read INPUT
 if [[$INPUT]]
 then continue
 else

isql -U $SYB_USER -P $SYB_PASSWD -o /dev/null <<EOF

update SW_LICENSES
set LICENSES_ALLOWED=$INPUT
where (SOFTWARE_ID="$PROG" and SOFTWARE_VERSION_ID="$VER")

go
commit
EOF

 fi
 fi
done

rm $WRKFILE
exit

```

Figure 103. Script to Update SW\_LICENSES ( Part 2 of 2 )

We then run the script `swlic_upd.sh` to enter the number of licenses for each of our programs. The following figure shows the terminal session during its execution:

```
swlic_upd.sh

Program: Microsoft DOS Command Processor
Version: 7.0
Is currently registered for NULL licenses.
Hit Enter to keep current value, or enter new value: 1

Program: Microsoft DOS XMS Driver
Version: 3.95
Is currently registered for NULL licenses.
Hit Enter to keep current value, or enter new value: 1

Program: Microsoft Exchange
Version: 1.0
Is currently registered for NULL licenses.
Hit Enter to keep current value, or enter new value: 1

Program: Microsoft Exchange EXE
Version: 4.00.835.1374
Is currently registered for NULL licenses.
Hit Enter to keep current value, or enter new value: 3

Program: Microsoft Network Client
Version: 4.2
Is currently registered for NULL licenses.
Hit Enter to keep current value, or enter new value: 2

Program: Microsoft Visual Basic runtime library
Version: 3.0
Is currently registered for NULL licenses.
Hit Enter to keep current value, or enter new value: 1

Program: Microsoft Windows 95
Version: 4.00.950
Is currently registered for NULL licenses.
Hit Enter to keep current value, or enter new value: 1

Program: My own program
Version: 1.11
Is currently registered for NULL licenses.
Hit Enter to keep current value, or enter new value:

Program: Netscape Navigator application file
Version: 3.01
Is currently registered for NULL licenses.
Hit Enter to keep current value, or enter new value: 2

#
```

Figure 104. Running the Script `swlic_upd.sh`

The following figure shows the contents of table SW\_LICENSES after we updated it using the script swlic\_upd.sh:

| SOFTWARE_ID                            | SOFTWARE_VERSION_ID | LICENSES_ALLOWED | LICENSES_USED |
|----------------------------------------|---------------------|------------------|---------------|
| Microsoft DOS Command Processor        | 7.0                 | 1                | NULL          |
| Microsoft DOS XMS Driver               | 3.95                | 1                | NULL          |
| Microsoft Exchange                     | 1.0                 | 1                | NULL          |
| Microsoft Exchange EXE                 | 4.00.835.1374       | 3                | NULL          |
| Microsoft Network Client               | 4.2                 | 2                | NULL          |
| Microsoft Visual Basic runtime library | 3.0                 | 1                | NULL          |
| Microsoft Windows 95                   | 4.00.950            | 1                | NULL          |
| My own program                         | 1.11                | NULL             | NULL          |
| Netscape Navigator application file    | 3.01                | 2                | NULL          |

Figure 105. Contents on Table SW\_LICENSES, After Update via swlic\_upd.sh

**Note**

The goal of our scenario is to give various suggestions on how to use TME 10 Inventory. To make this example a full blown license management application, we should of course extend the swlic\_upd.sh script to ask if the program is licensed. If it is not, we should not check licenses for it, and therefore remove the row from the SW\_LICENSES table.

### 5.2.4 Compare Licenses to Actual Installed Programs

We now write a script swlic\_cmp.sh to do the following:

- Execute an inventory scan via the Tivoli CLI command wdistrib.
- Populate the column LICENCES\_USED in table SW\_LICENSES.
- Send a T/EC event if LICENCES\_USED is higher than LICENCES\_ALLOWED.

The script is shown on the following two pages:

```
#!/bin/ksh

This script reads the table SW_LICENSES. For each line
it then does a select from table INSTALLED_SOFTWARE_VERSION
and updates the column LICENSES_USED in table SW_LICENSESr.,
If the value exceeds LICENSES_ALLOWED a T/EC event will be sent.

SYB_USER=tivoli
SYB_PASSWD=tivoli
WRKFILE=/usr/local/Tivoli/localtools/wrk3
WRKFILE2=/usr/local/Tivoli/localtools/wrk4
PATH=$PATH:/usr/local/sybase/bin
export PATH
SYBASE=/usr/local/sybase
export SYBASE

#
Run an Inventory scan to update the database

wddistrib foxtrot-region/Inventory

#
Create a temporary file from table SW_LICENSES

isql -U $SYB_USER -P $SYB_PASSWD -o $WRKFILE -w 115<<EOF

select SOFTWARE_ID, SOFTWARE_VERSION_ID, LICENSES_ALLOWED
 from SW_LICENSES
go
EOF

#
Now read the temporary file, and for each line
do a select from table INSTALLED_SOFTWARE_VERSION

exec 3< $WRKFILE
IFS=\t
while read LINE
do
 IFS=
 # But remove header trailer and blank lines produced by SYBASE output
 if [[echo $LINE | cut -c 1-12 = " SOFTWARE_ID" || \
 echo $LINE | cut -c 1-12 = " -----" || \
 echo $LINE | cut -c 1-12 = " " || \
 echo $LINE | cut -c 1 = "("]]
 then continue
 else

 PROG=echo $LINE | cut -c 2-65
 VER=echo $LINE | cut -c 67-98
 LIC=echo $LINE | cut -c 100-115

isql -U $SYB_USER -P $SYB_PASSWD -o $WRKFILE2 <<EOF
```

Figure 106. Script to Check Licenses and Send T/EC Event (Part 1 of 2)

```

select SOFTWARE_ID, SOFTWARE_VERSION_ID
 from INSTALLED_SOFTWARE_VERSION
 where (SOFTWARE_ID="$PROG" and SOFTWARE_VERSION_ID="$VER")
go
EOF
Find out number of rows selected for this program

integer USE

USE=grep "affected)" $WRKFILE2 | cut -f 1 -d ' ' | cut -f 2 -d '('

Update column LICENSES_USED in table SW_LICENSES

isql -U $SYB_USER -P $SYB_PASSWD -o /dev/null <<EOF

update SW_LICENSES
set LICENSES_USED=$USE
where (SOFTWARE_ID="$PROG" and SOFTWARE_VERSION_ID="$VER")

go
commit
EOF

If USED greater than ALLOWED send T/EC event

 if [[$USE != " NULL"]]
 then
 if [[$LIC = " NULL"]]
 then print " Program: $PROG \n Version: $VER \n used without licenses"

 postmsg -S rs600026 -r MINOR -m "Program license missing" \
 program="$PROG", version=$VER, numuse=$USE \
 License_Missing LICENSES

 else
 if (($USE > $LIC))
 then print " Program: $PROG \n Version: $VER \n has $LIC licenses but is used by $USE"
 postmsg -S rs600026 -r MINOR -m "Program licenses exceeded" \
 program="$PROG", version="$VER", \
 numuse="$USE", numlic="$LIC" License_Exceeded LICENSES
 fi
 fi
 fi

done <$WRKFILE

rm $WRKFILE
rm $WRKFILE2
exit

```

Figure 107. Script to Check Licenses and Send T/EC Event (Part 2 of 2)

After having run the script the SW\_LICENSES table looks as shown in the following figure:

| SOFTWARE_ID                            | SOFTWARE_VERSION_ID | LICENSES_ALLOWED | LICENSES_USED |
|----------------------------------------|---------------------|------------------|---------------|
| Microsoft DOS Command Processor        | 7.0                 | 1                | 2 <b>1</b>    |
| Microsoft DOS XMS Driver               | 3.95                | 1                | 1             |
| Microsoft Exchange                     | 1.0                 | 1                | 1             |
| Microsoft Exchange EXE                 | 4.00.835.1374       | 3                | 1             |
| Microsoft Network Client               | 4.2                 | 2                | 1             |
| Microsoft Visual Basic runtime library | 3.0                 | 1                | 1             |
| Microsoft Windows 95                   | 4.00.950            | 1                | 1             |
| My own program                         | 1.11                | NULL             | 1 <b>2</b>    |
| Netscape Navigator application file    | 3.01                | 2                | 3 <b>1</b>    |

Figure 108. Contents on Table SW\_LICENSES, After Update by swlic\_cmp.sh

The lines marked **1** in the figure will result in a message to the standard out file and a T/EC event saying that license is exceeded. The line marked **2** will result in a similar message and T/EC event saying that the product has no license.

### 5.2.5 Create a New Baroc File to Use with TME 10 Enterprise Console

In order to recognize the two new event types sent from the swlic\_cmp.sh, we have to describe them to TME 10 Enterprise Console. We describe our two new event classes License\_Missing and License\_Exceeded in a file, and call that file lic.baroc. The file can be stored in any directory, and will later be imported into a rule base in T/EC. The contents of our file lic.baroc are shown in the next figure.

```
TEC_CLASS :
 License_Missing ISA EVENT
 DEFINES {
 program: STRING, dup_detect= no ;
 version: STRING, dup_detect= no ;
 numuse: STRING, dup_detect= no ;
 };
END

TEC_CLASS :
 License_Exceeded ISA EVENT
 DEFINES {
 program: STRING, dup_detect= no ;
 version: STRING, dup_detect= no ;
 numlic: STRING, dup_detect= no ;
 numuse: STRING, dup_detect= no ;
 };
END
```

Figure 109. BAROC File to Describe License Events

## 5.2.6 Create a New Rule Base

We create a new rule base into which we want to import our baroc file. On the Tivoli desktop we click the right mouse button on the **Event Server** icon, and select **Rule Bases...** from the pull-down menu as shown in following figure:



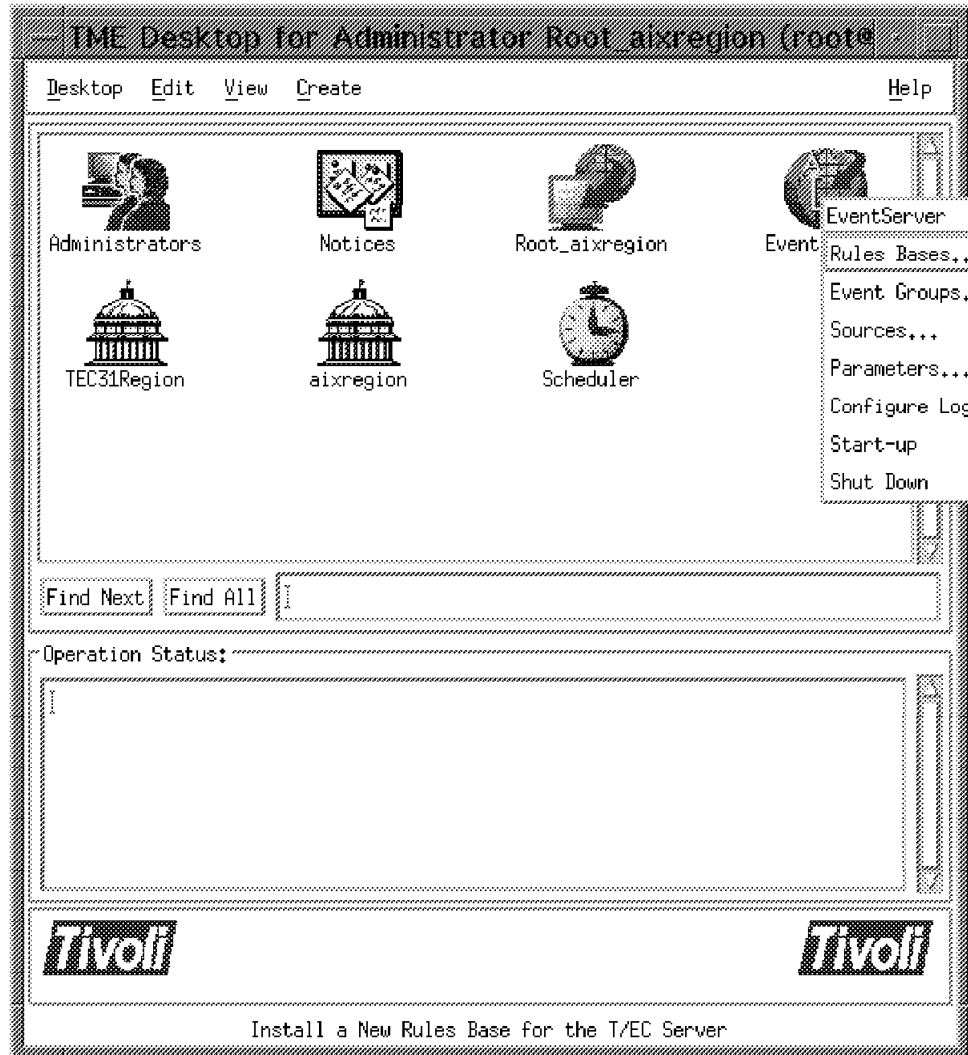


Figure 110. Accessing Rule Base Definitions

The Event Server Rule Bases window is displayed and contains our current rule bases as shown in the next figure.

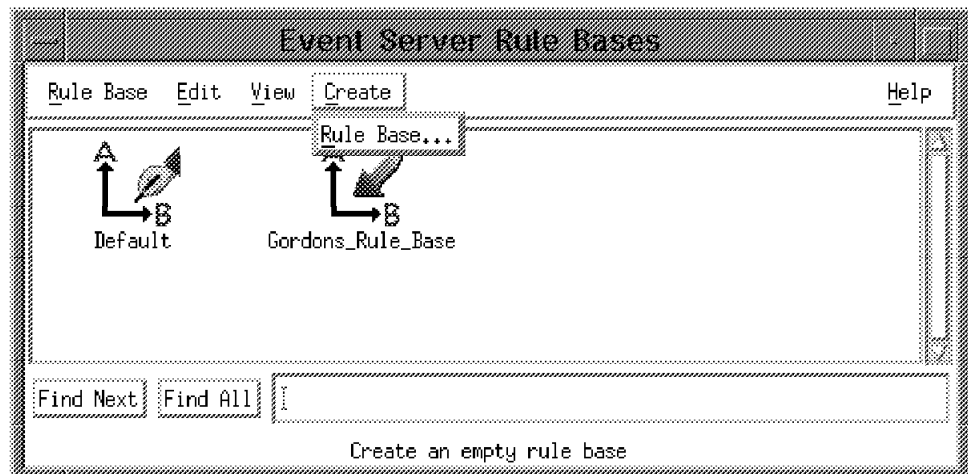


Figure 111. Creating a New Rule Base

In the Event Server Rule Bases window, we select **Rule Base...** from the Create pull-down, and get a window where we can enter the properties for the new rule base. This is shown in the next figure:

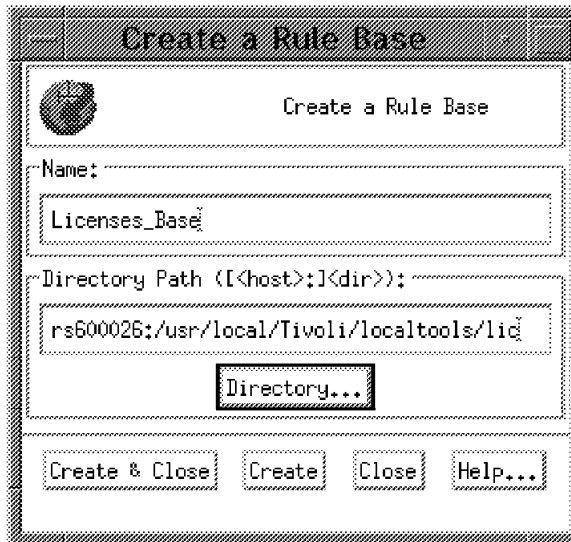


Figure 112. Rule Base Properties

We give our new rule base the name Licences\_Base. We also enter a directory path, where we want the rule base to be created. Finally, we click **Create & Close**. Our new rule base will now be created, and an icon that represents it will be created in the Event Server Rule Bases window as shown in the next figure.

## 5.2.7 Copy Default Rule Base to Licences\_Base Rule Base

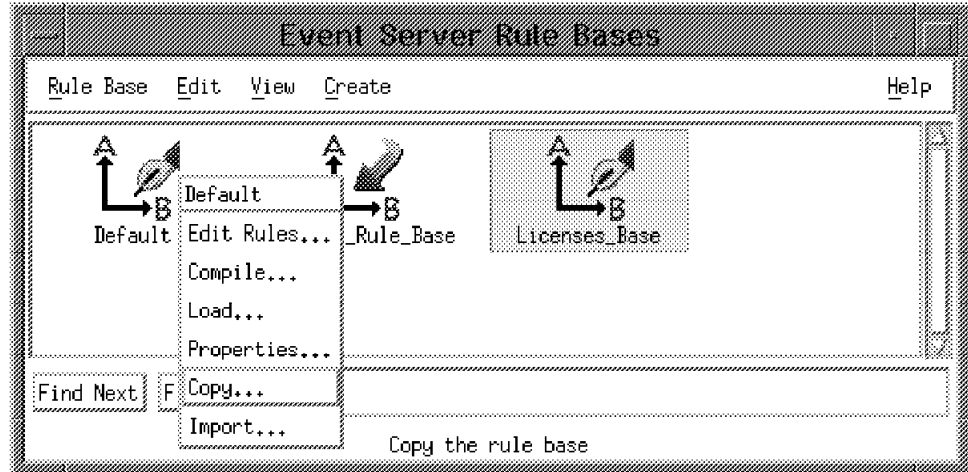


Figure 113. Copy Default Rule Base to Licences\_Base Rule Base

We want to keep all current rules and classes, and only add the new classes defined in our baroc file. We will therefore copy the Default rule base to our Licences\_Base rule base, and then add our own definitions. The rule base copy is always a copy from the source rule base to the destination rule base. We click the right mouse button on the Default rule base icon and select **Copy...** We will get a window to select our copy destination as shown in the next figure:

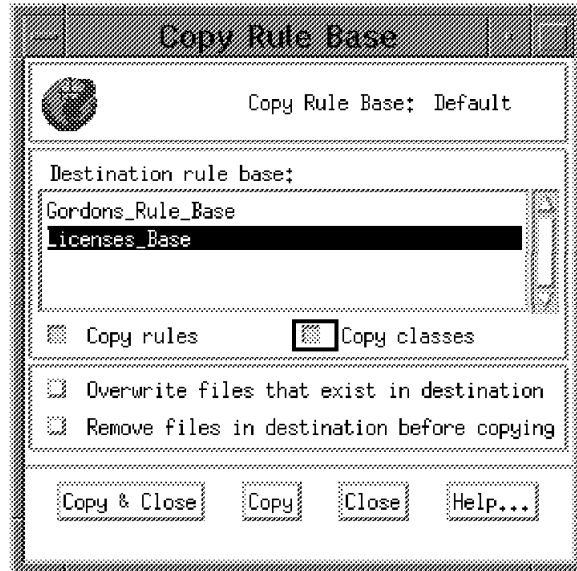


Figure 114. Copy Rule Base

From the list we select Licences\_Base as the destination. We click on both **Copy rules** and **Copy classes**. Then we click **Copy & Close** and the copy will be performed.

## 5.2.8 Import Baroc File to Licences\_Base Rule Base

The rules and classes are now the same in our new Licences\_Base rule base as in the Default rule base. Now we want to add our own classes to the Licences\_Base. We do that by selecting **Import** from the pull-down of the Licences\_Base icon as shown in the next figure:

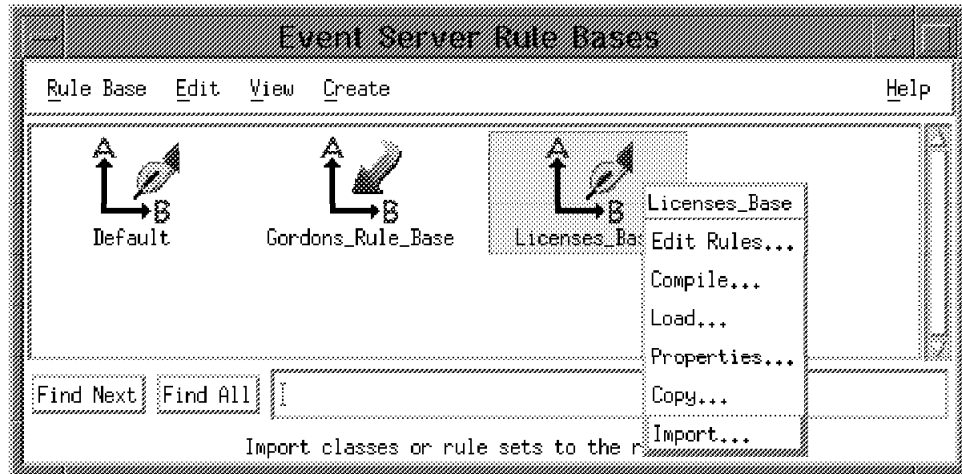


Figure 115. Select Import to Rule Base

The Import Into Rule Base window will pop up, as shown in the next figure:

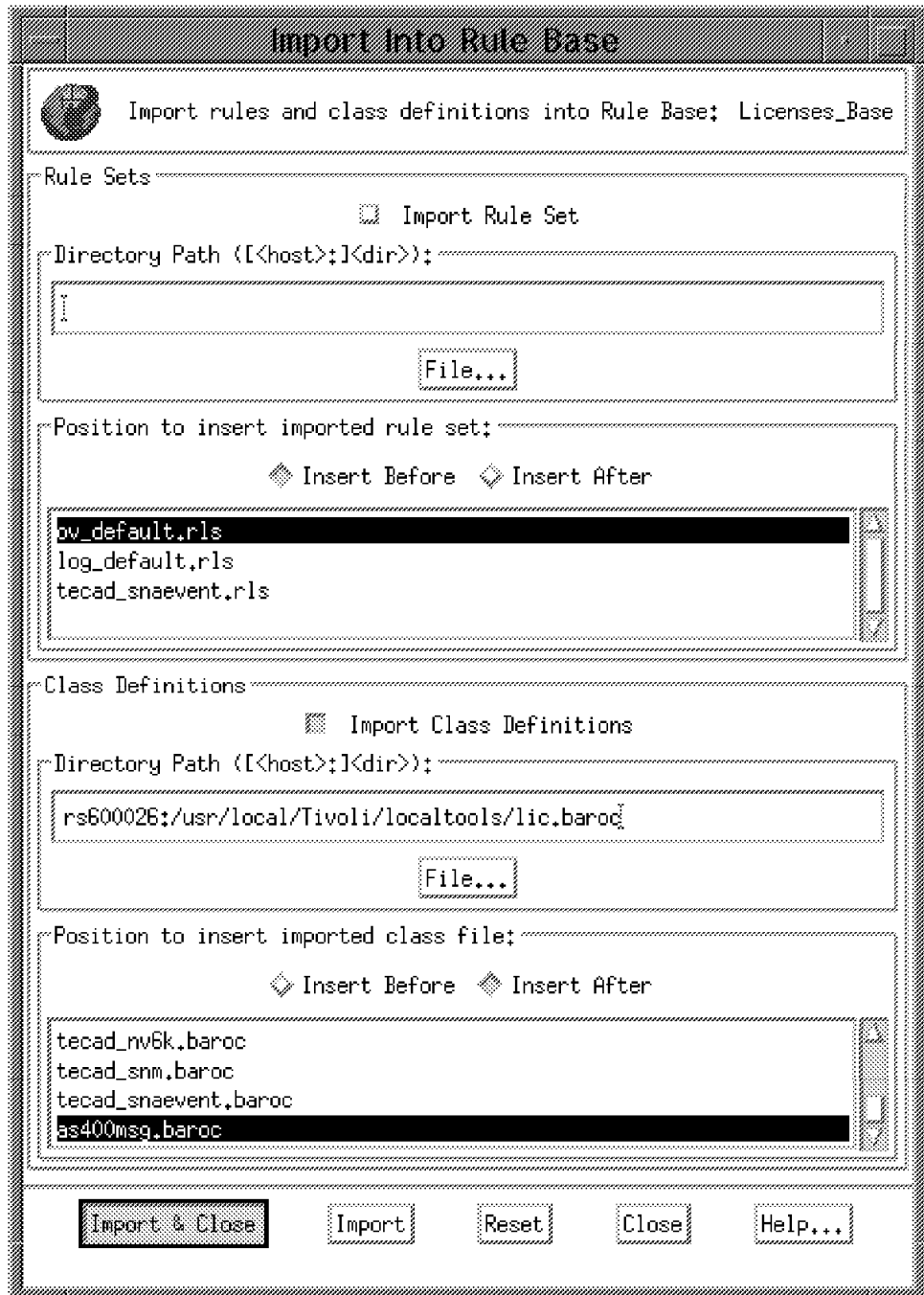


Figure 116. Import Into Rule Base

Only class definitions will be imported (not rule sets), so we will only use the lower part of the window. We click the **Import Class Definitions** button. Then we enter the directory path and file name for our lic.baroc file that was shown in Figure 109 on page 132. As our lic.baroc file refers to an event class already defined, we must import it after the baroc file where it is defined. We click the **Insert After** button, scroll down the list, and click on the very last entry. Finally, we click **Import & Close** and our lic.baroc will be imported to the Licences\_Base rule base.

## 5.2.9 Compile the Licences\_Base Rule Base

To verify that the rules and classes can be resolved correctly we must now compile our new rule base. We select **Compile...** from the pull-down of the Licences\_Base icon as shown in the next figure:

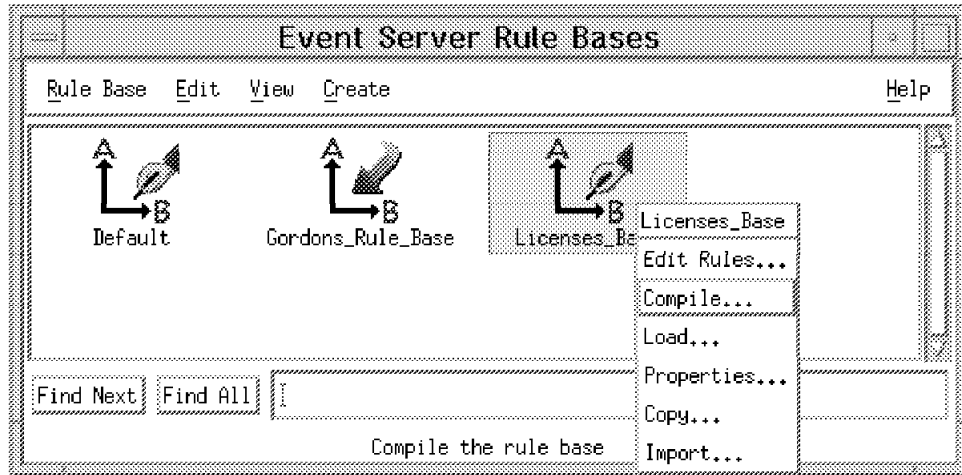


Figure 117. Select Compile Rule Base

We get an empty Compile Rule Base window. We click the **Compile** button. The compilation will take place, and the results can then be browsed in the window, as shown in the following figure:

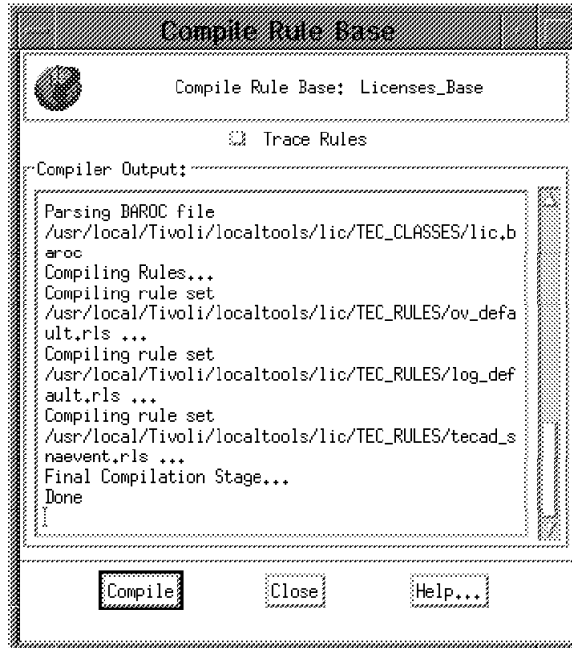


Figure 118. View Compile Results

Once we have verified a successful compilation, we close the Compile Rule Base window.

### 5.2.10 Make Licences\_Base the Active Rule Base

Now it is time to activate our new rule base.

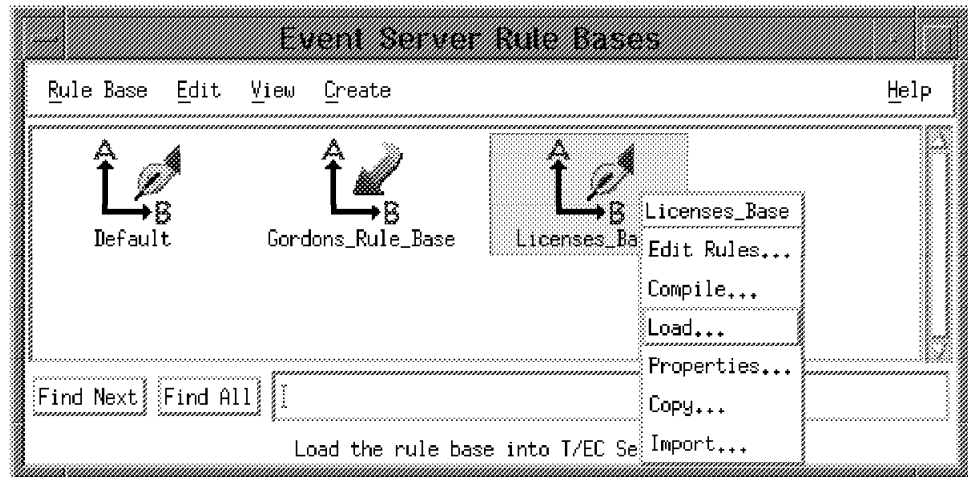


Figure 119. Load Rule Base

We click the right mouse button on the Licences\_Base icon, and select **Load...** from the pull-down menu. We will get a Load Rule Base window as shown in the following figure:

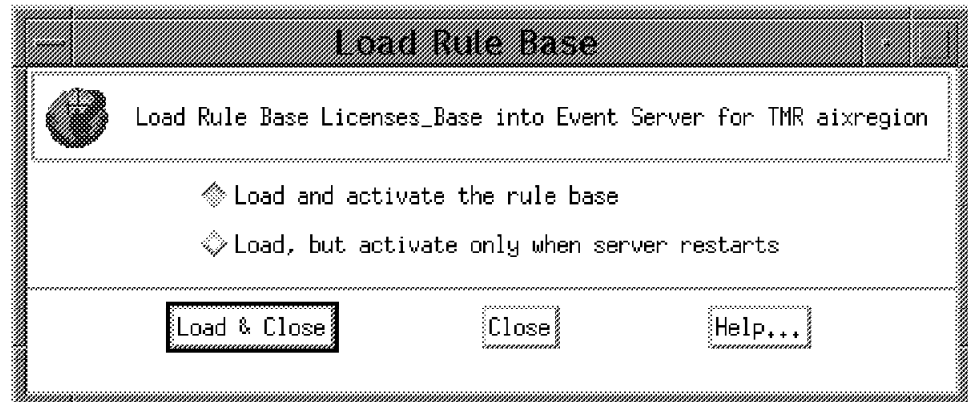


Figure 120. Load Rule Base Options

We click the radio button next to Load and activate the rule base. Next we click the **Load & Close** button. Our Licences\_Base rule base is now the current rule base which is indicated by the arrow pointing to its icon, as shown in Figure 121 on page 140.

#### Note

We found that in our environment the active event server did not recognize our new events even though we asked for the rule base to be activated when loaded. We had to stop and start the event server to get it all working.

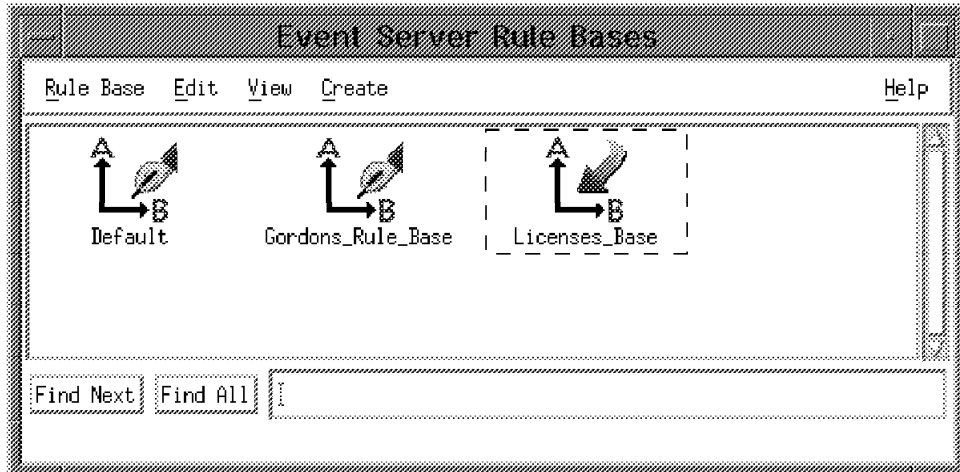


Figure 121. Licences\_Base Rule Base Is Active

### 5.2.11 Create a New Event Source

So far we have worked with the configuring, recognizing, and parsing parts of T/EC for our new events. Next we will take care of the filtering and monitoring definitions. We must create a new source definition LICENSES, and an event group that filters events coming from the LICENSES source.

On the TME 10 Desktop, we click on the Event Server icon with the right mouse button and select **Sources...** The T/EC Source List window shown in the next figure is displayed.



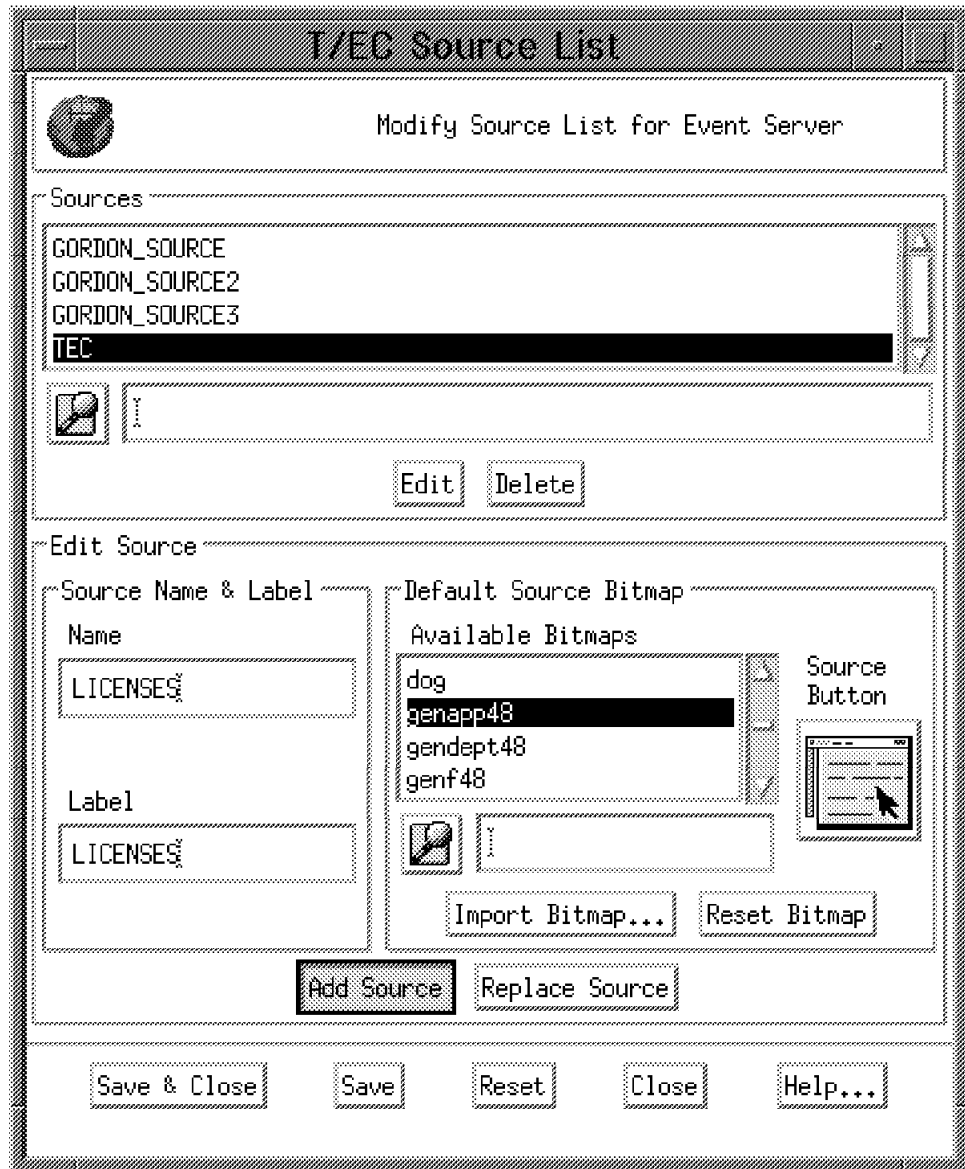


Figure 122. Add T/EC Source

In the lower part of the window we enter LICENSES in both the Name and the Label fields. We also select a suitable bitmap. We then press the **Add Source** button and our LICENSES source will be added to the list of sources. Finally, we press the **Save & Close** button, and exit the window.

### 5.2.12 Create a New Event Group

On the TME 10 Desktop, we click on the Event Server icon with the right mouse button and select **Event Groups...**. We will get the Event Group Management window shown in the next figure:

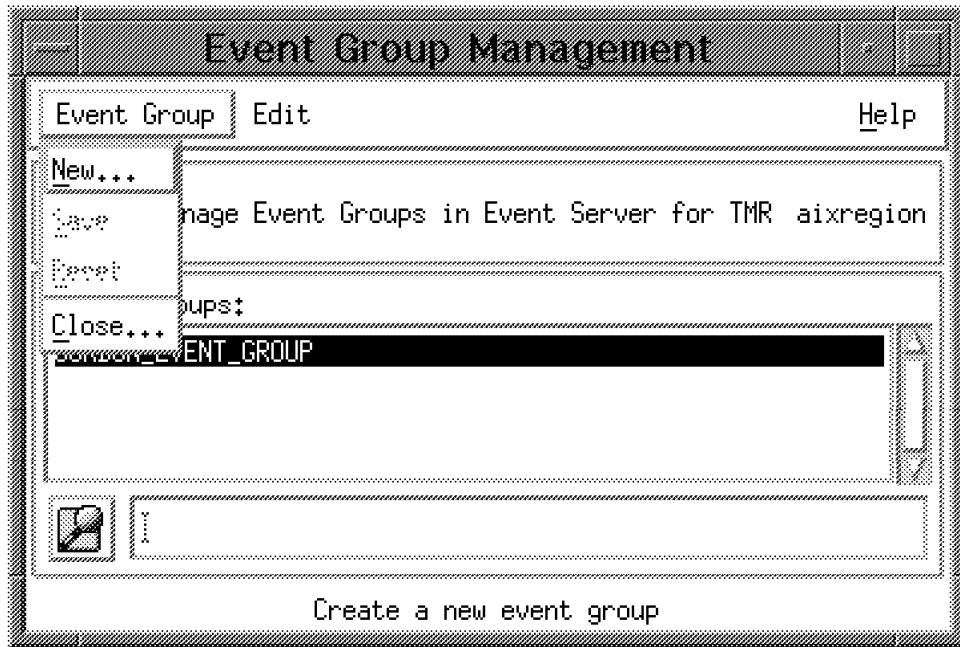


Figure 123. Event Group Management

From the Event Group Management window, we select **New...** from the Event Group pull-down. We will now get the New Event Group window as shown in the next figure.



Figure 124. Add the Event Group

We enter our new Event Group Name as License, and select a suitable icon from the scroll list. Finally we click the **Create** button.

As this is a new Event Group, there is no filter assigned to it yet. At least one filter is required, so the Edit Event Group Filters window will automatically pop up, as shown in Figure 125.

### 5.2.13 Assign a New Event Filter

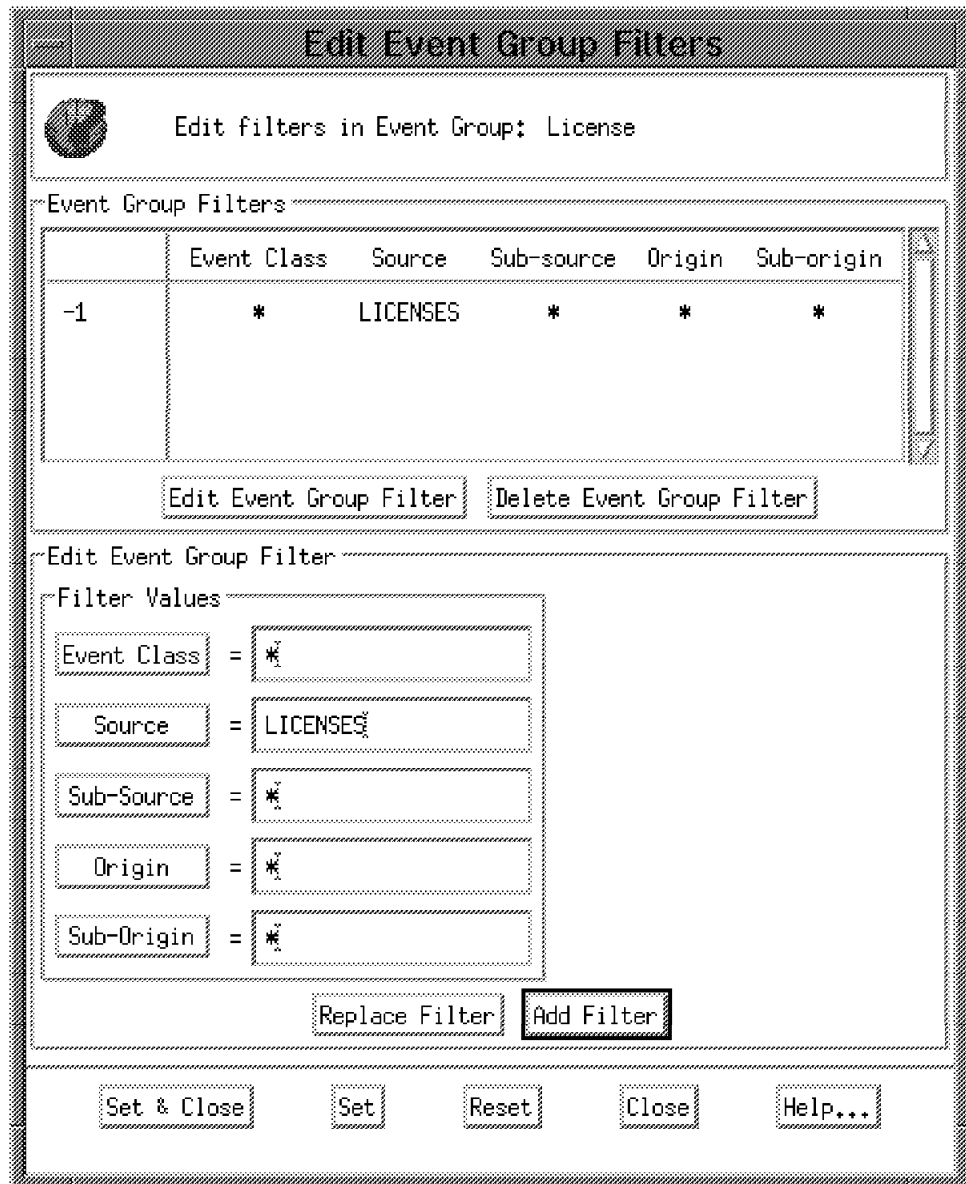


Figure 125. Create Filter for the Event Group

In the lower part of the window we enter the name of our source LICENSES in the Source field. We enter an asterisk (\*) in all the other fields. The filter is a positive filter, so the asterisks are like wild cards. Our filter will let through all events from the source LICENSES. We click the **Add Filter** button, and the filter will be shown in the list at the top of the window. Finally we click the **Set & Close** button, and we are back at the Event Group Management window.

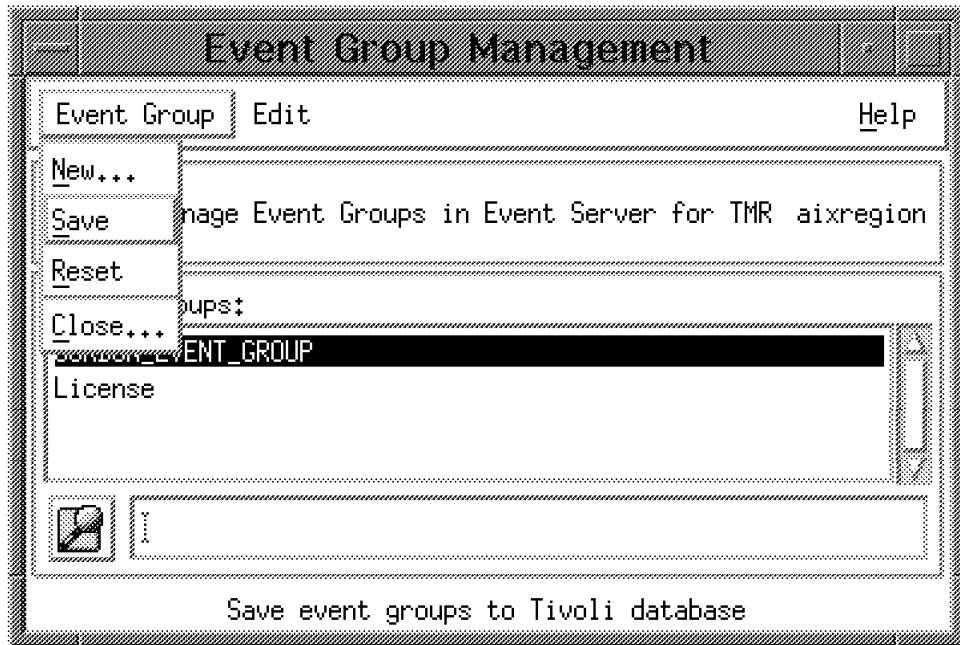


Figure 126. Save the Event Group

We are now done with the Event Group Management, but before we exit we select first **Save** then **Close** from the **Event Group** pull-down menu.

#### 5.2.14 Assign the New Event Group to an Administrator

We must now assign our new event group. On the TME 10 Desktop, we select **Assign Event Groups...** from the pull-down of the Event monitor icon as shown in the following figure:

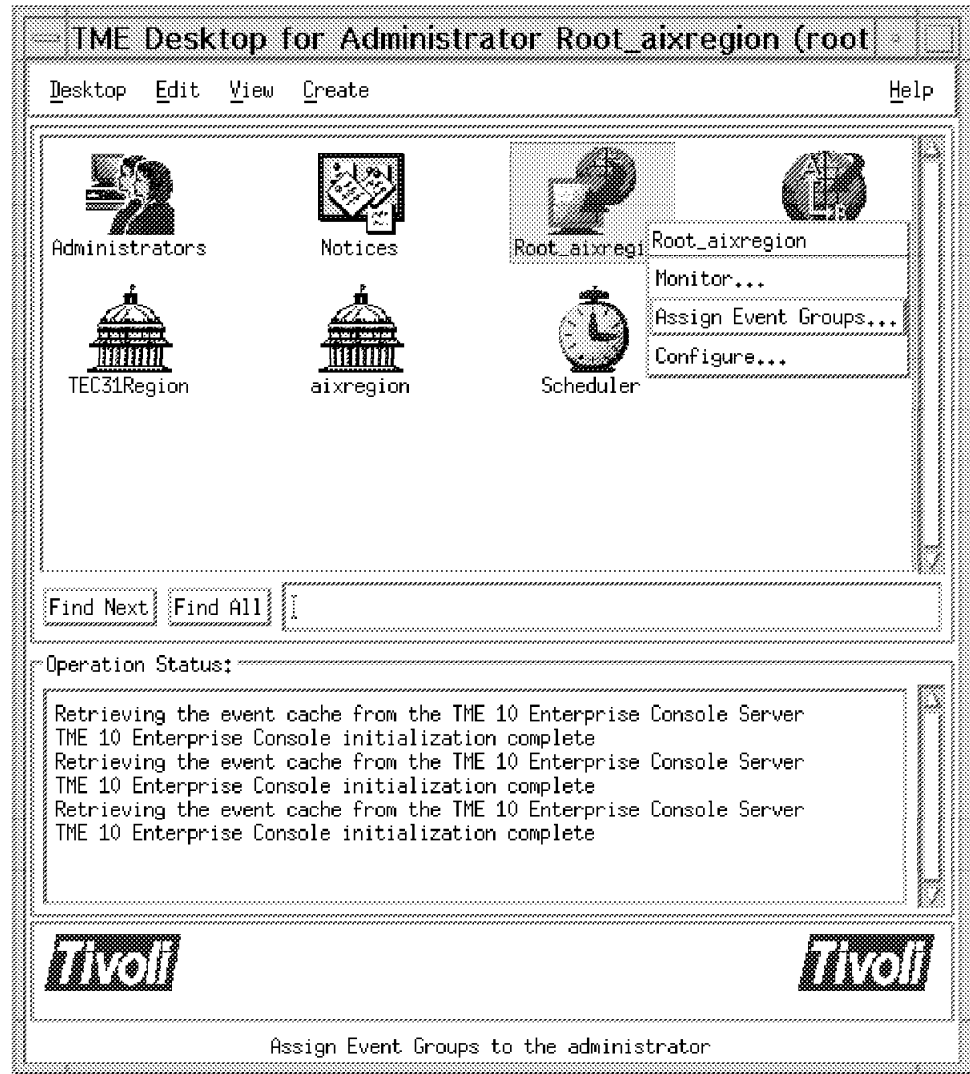


Figure 127. Calling the Assign Event Groups Window

We will get the Assign Event Groups window shown in the next figure.

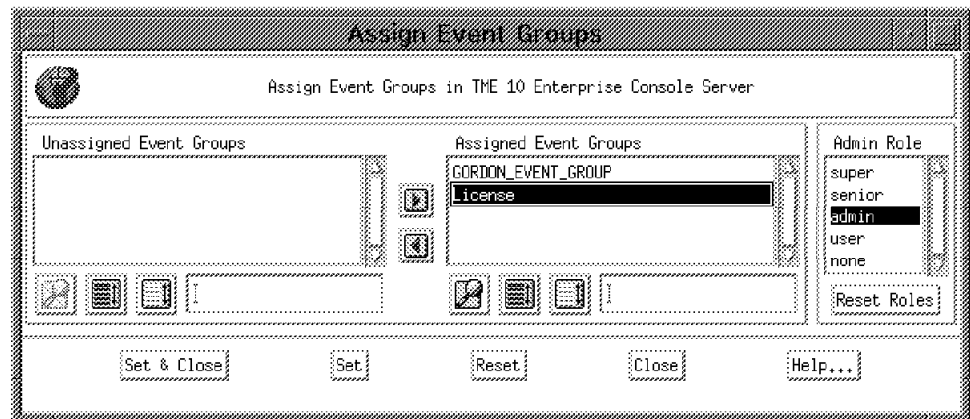


Figure 128. Assign Event Groups Window

Our new event group License is first seen in the Unassigned Event Groups window on the left. We select it and click on the right arrow to bring it over to

the Assigned Event Groups window on the right. We then select the Admin Role **admin** to be assigned to our new event group. Finally we click the **Set & Close** button.

## 5.2.15 Monitoring the Events

Now it is time to bring up the Monitor window to really watch for the events coming in. To get to the Monitor window, we select **Monitor...** from the pull-down menu of the Event monitor icon (see Figure 127 on page 145). Both the Source and the Event Group windows will show up as seen in the next figure:

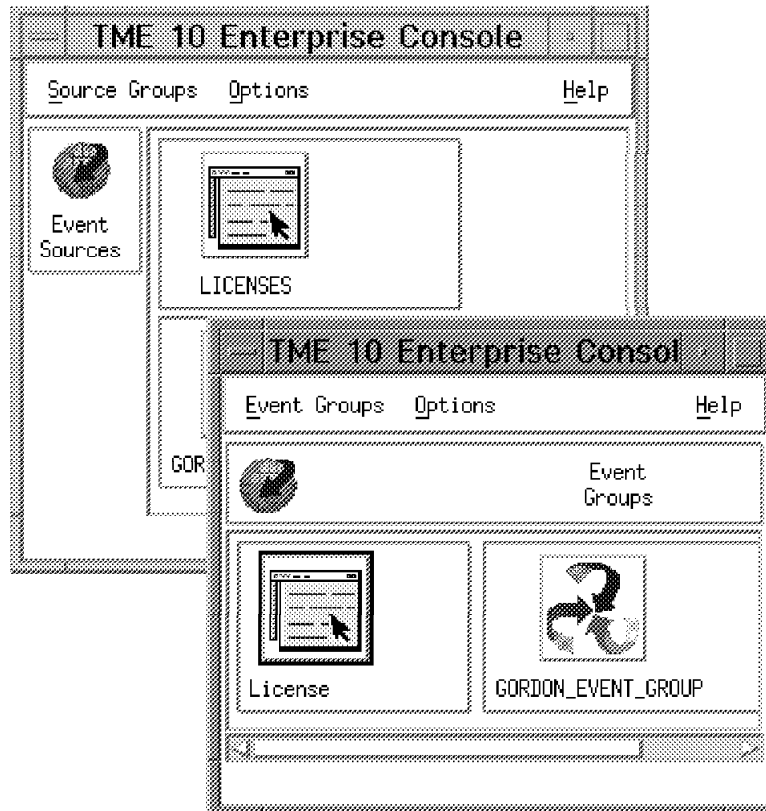


Figure 129. Event Groups and Sources

We now run our script `swlic_cmp.sh` (see Figure 106 on page 129) to check for missing or exceeded licenses. We entered some low values for available licenses, so we expect that the script will send some events to T/EC.

### Note

At this point, we discovered that T/EC would not recognize our new events until we shut down and restarted our Event Server. We used the command `wtdump1` to verify that the event did arrive, but it was not picked up by the filters.

When an event arrived at our Event Group, it was indicated by a thermometer icon next to the Event Group icon. The severity of the most severe event will also be indicated. The next figure shows that we have received a MINOR event in the Event Group License.

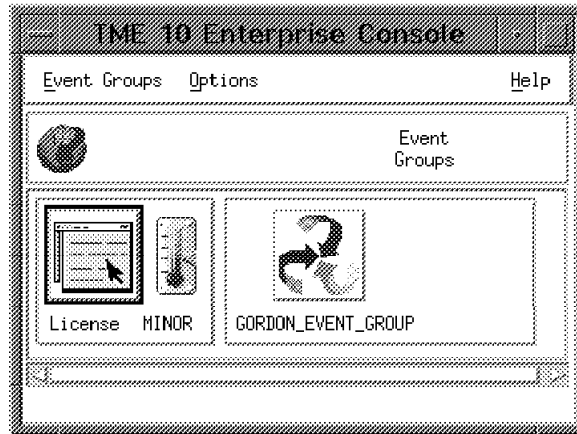


Figure 130. Event Group with Indicator

A click on the License icon will open up the License monitor window as shown in the next figure.

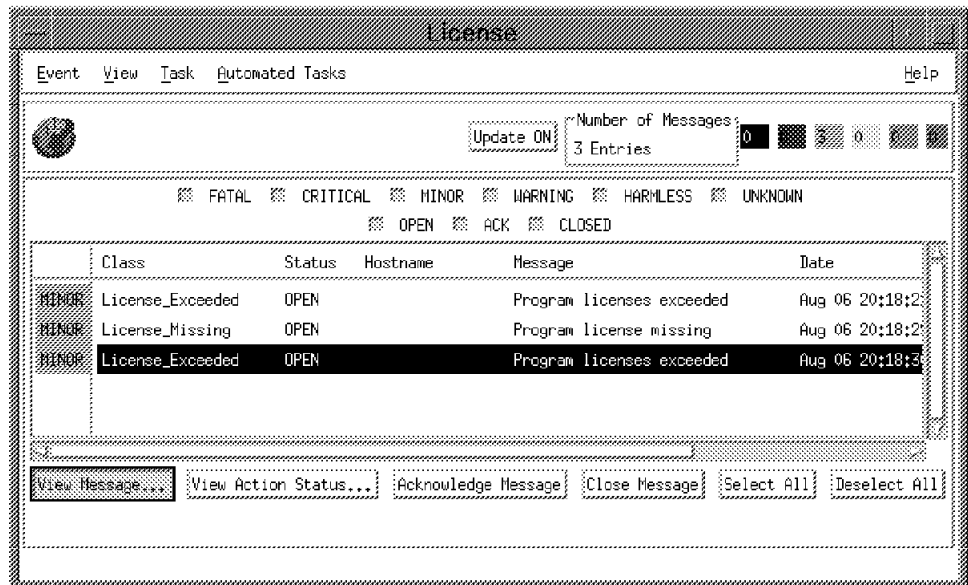


Figure 131. License Event Group Monitor

We have received three MINOR events. Two events of the class License\_Exceeded, and one event of the class License\_Missing. We want to know more about the details, so we select the third event, and click the **View Message...** button to get the Message Viewer window as shown in next figure.

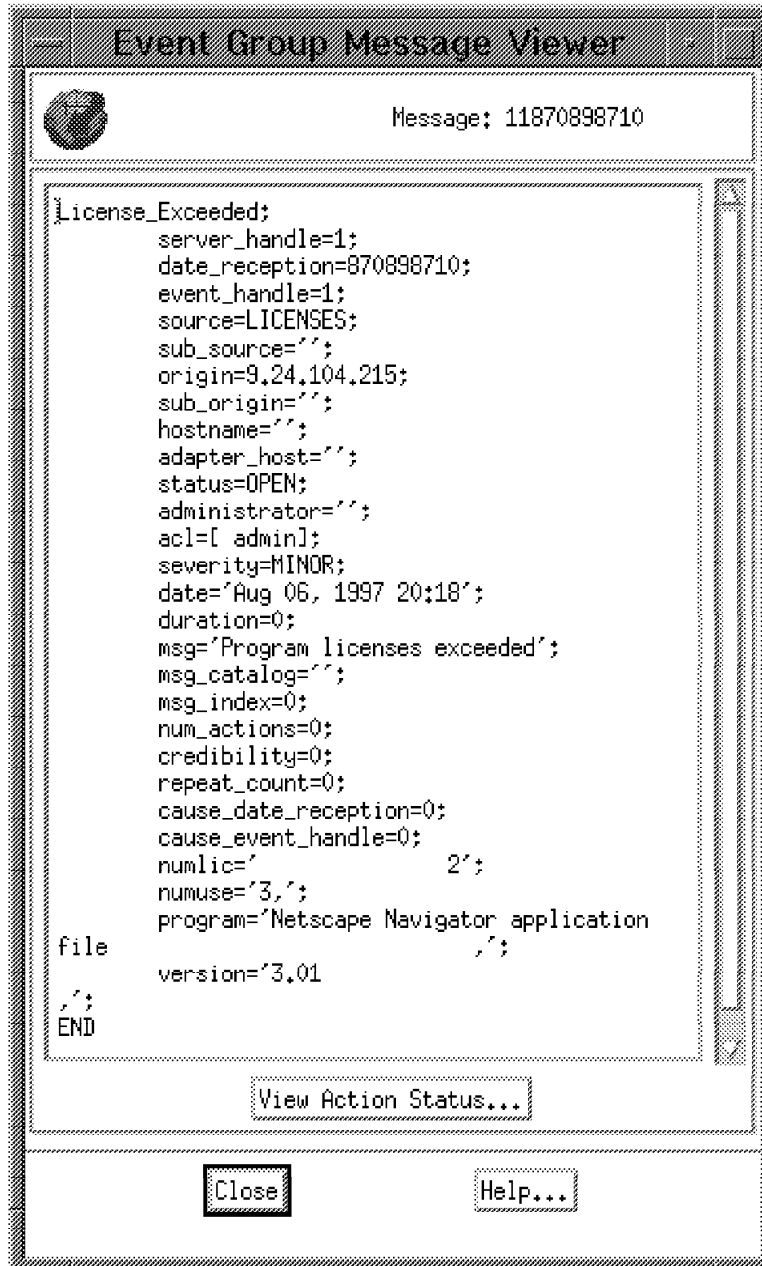


Figure 132. Event Detail

The last few lines show that Netscape Navigator Version 3.01 is licensed for 2 users (numlic=), but is being used by 3 users (numuse=).



## 5.2.16 Include the License Checking Script in a TME 10 Task

We have now verified that the routine works, and that the events are being received and monitored by TME 10 Enterprise Console. It would be nice to let the license compare script run automatically once a week. We can achieve that by using standard TME 10 Framework functions.

We will do the following:

- Create a TME 10 task, and include the `swlic_cmp.sh` script in the task.
- Execute the task to verify that the script can run in a TME 10 task.
- Create a TME 10 job, and include the TME 10 task in the job.
- Schedule the TME 10 job to run once a week.

We create a TME 10 task library in one of our policy regions, and call it `License_tasks`. In that task library we now want to create a task. We select **Task...** from the Create pull-down as shown in the next figure:

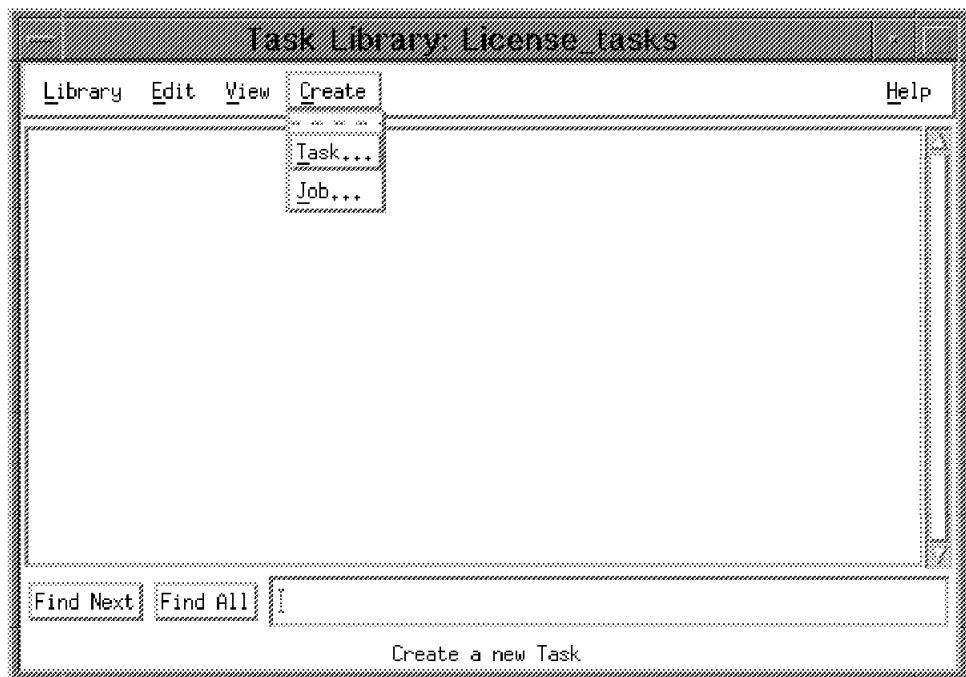


Figure 133. Create a TME 10 Task

We get a Create Task window, shown in the next figure, where we will enter the properties for the task.

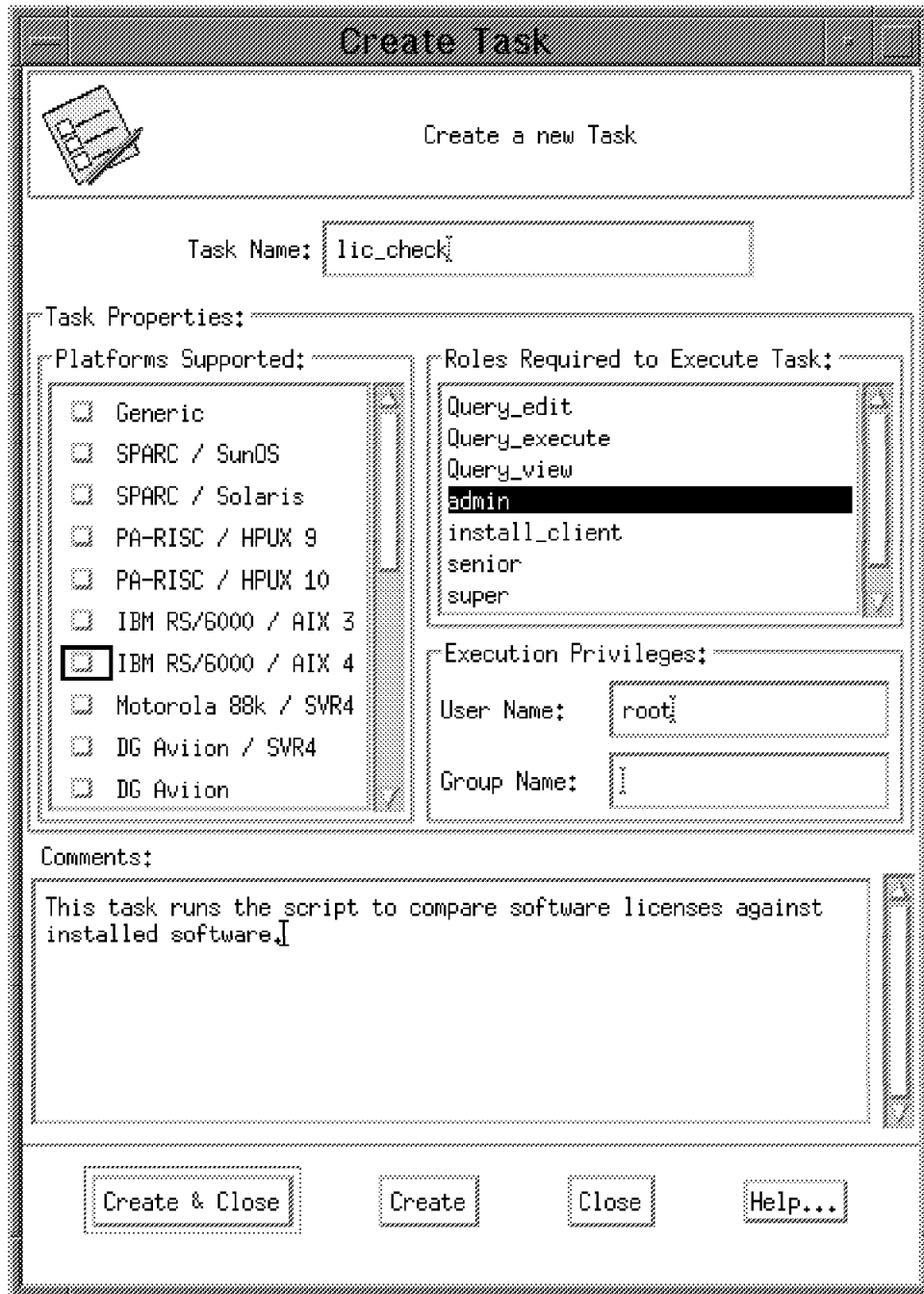


Figure 134. TME 10 Task Properties

We enter the task name `lic_check`. For the Roles Required we select **admin**. In the User Name field we enter `root`. This is because the script `swlic_cmp.sh` must run with root privileges in order to access the database. We also enter a short text in the Comments window, to help us to remember the purpose of this TME 10 task. And, since the host where the script will run is an AIX 4 machine, we click the button next to `IBM RS/6000 / AIX 4`.

We get the following window, and are prompted for a name and location of the executable associated with this task.

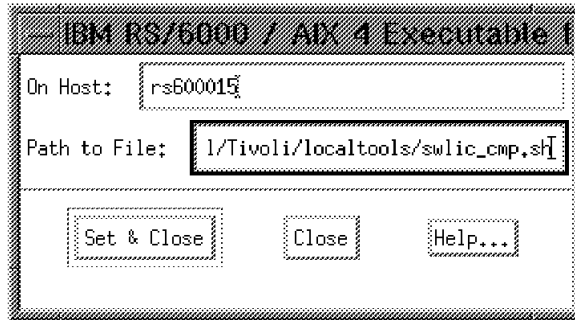


Figure 135. TME 10 Task Script Location

We enter rs600015 as the host name, and also the full path and file name for the /usr/local/Tivoli/localtools/swlic\_cmp.sh executable. We click the **Set & Close** button.

We then get back to the previous Create Task window (Figure 134 on page 150), and on that window, we click the **Create & Close** button. The task is now created and loaded in the TME 10 object database.

### 5.2.17 Execute the TME 10 Task

Now we want to verify that the script can be executed as a TME 10 task.

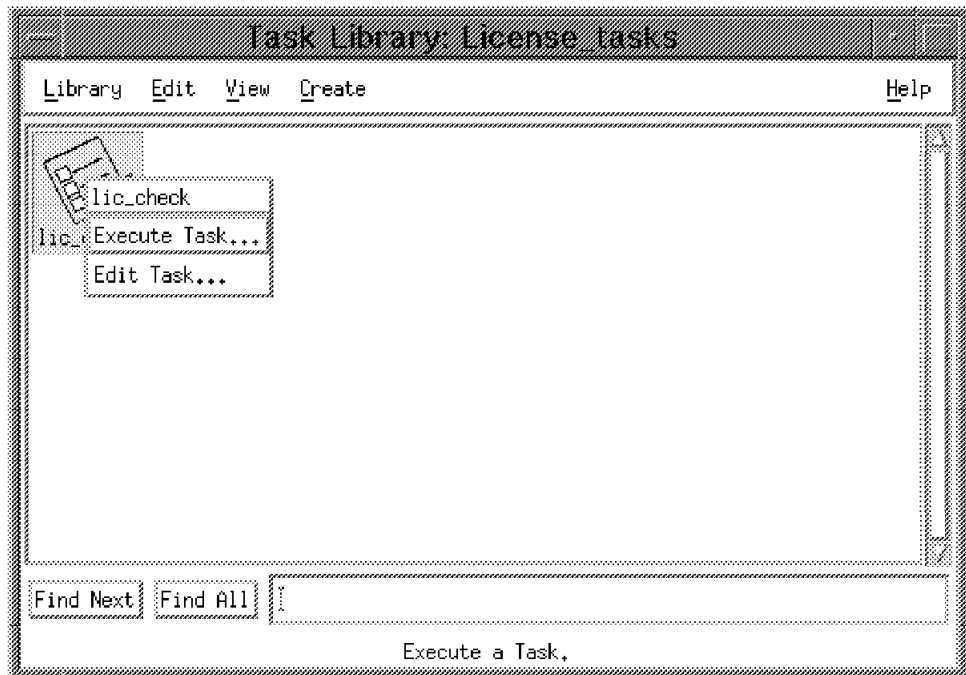


Figure 136. Execute the TME 10 Task

In the task library window we select **Execute Task...** from the pull-down on the lic\_check task icon. We get an Execute Task window, shown in the next figure, where we are prompted for the execute options for the task.

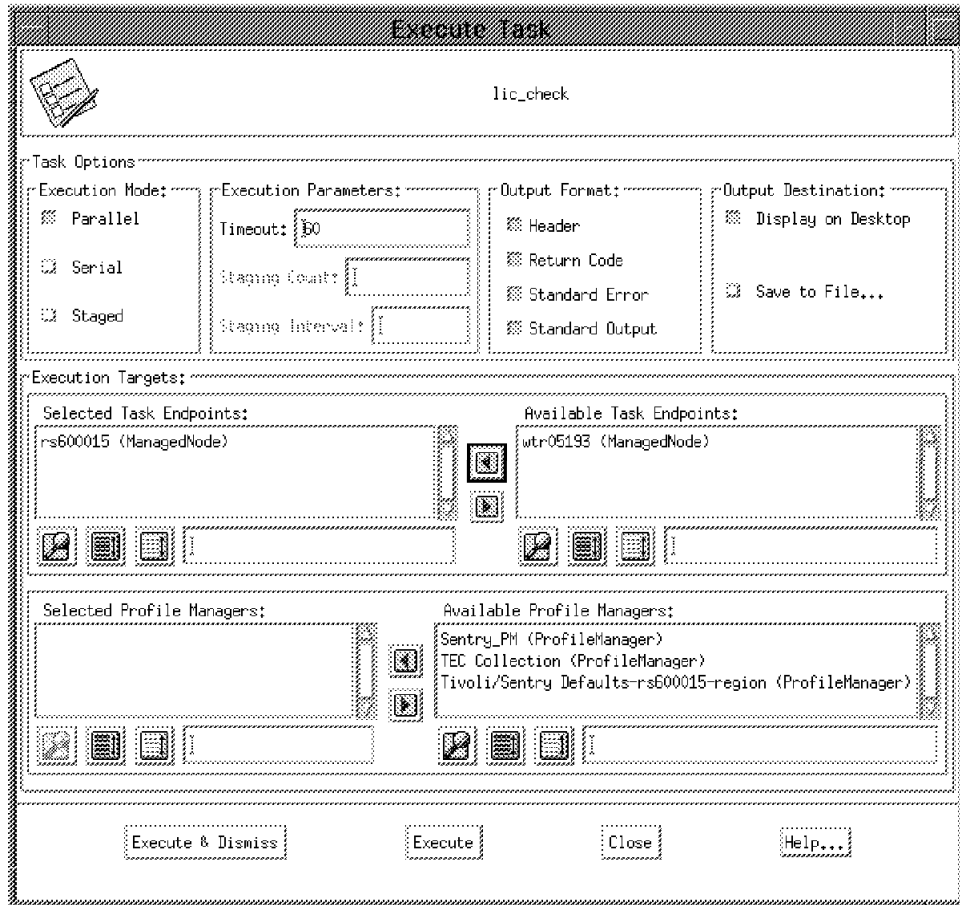


Figure 137. TME 10 Task Execute Options

In the Output Destination section, we click the radio button next to Display on Desktop to be able to view the output from the task. In the Execution Targets section, we move rs600015 from Available Task Endpoints to Selected Task Endpoints, using the left arrow button. We keep the rest of the options as the defaults shown in the figure. Finally, we click the **Execute & Dismiss** button.

The task will now execute. As we asked to get the output displayed on the desktop, we will get the output from the task in another window, as shown in the following figure:

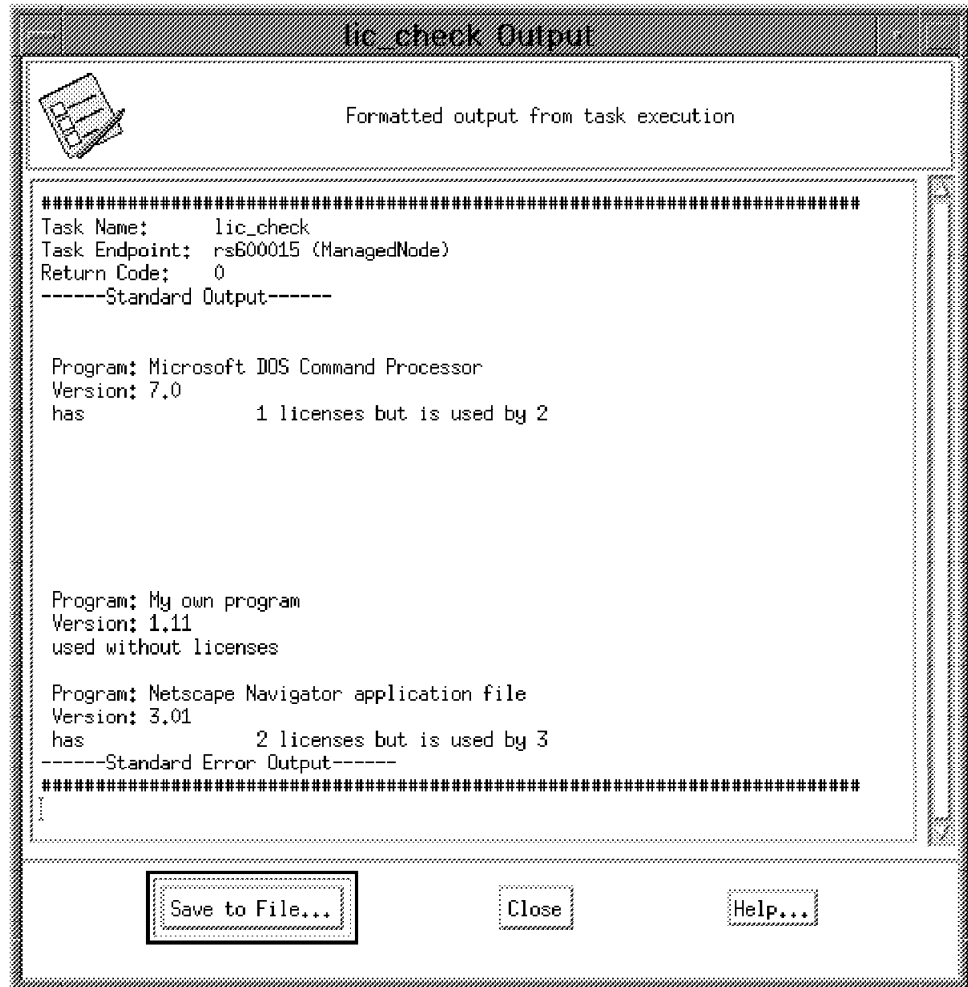


Figure 138. TME 10 Task Output

The return code is 0. Standard Output looks okay and Standard Error Output is empty. So, we consider the execution to be successful, and just click the **Close** button.

### 5.2.18 Create a TME 10 Job

We now have a working TME 10 task. But every time we want to execute it, we must enter the execute options as shown in Figure 137 on page 152.

We can avoid that by creating a TME 10 job to include the TME 10 task. A TME 10 Job is a container for both a task and the task execution options, so we will create a job.

In the task library, we select **Job...** from the Create pull-down as shown in the next figure:

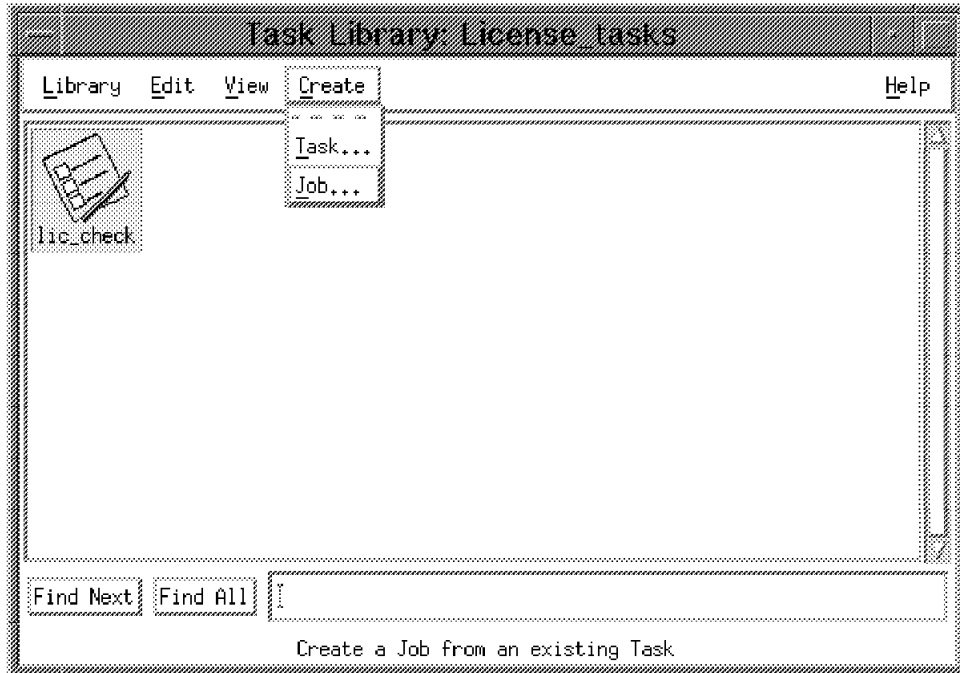


Figure 139. Create a TME 10 Job

We get the Create Job window, as shown in the next figure:

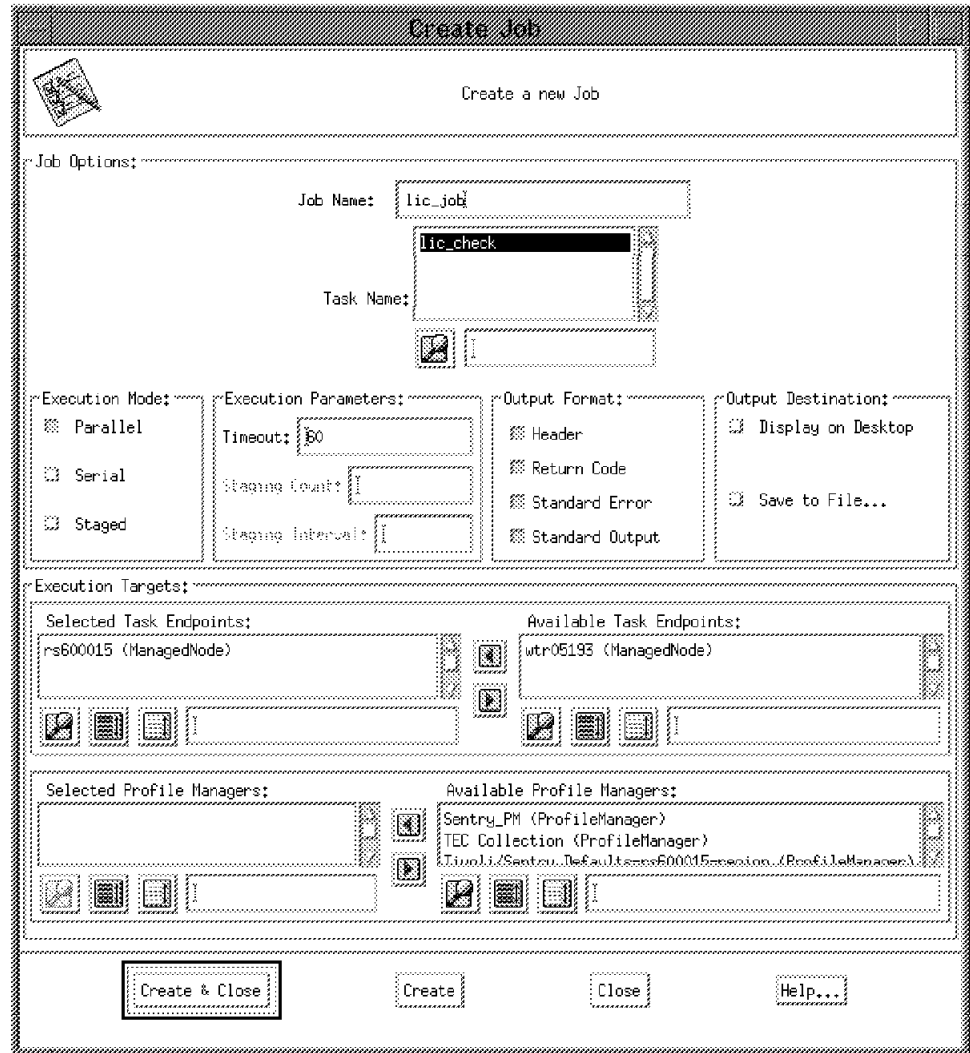


Figure 140. TME 10 Job Properties

We enter the same execution options as entered in the Execute Task window shown in Figure 137 on page 152, except that we do not click any button in the Output Destination section. In the Execution Targets section, we move rs600015 from Available Task Endpoints to Selected Task Endpoints, using the left arrow button. We keep the rest of the options as the defaults shown in the figure. In addition we also enter the Job Name lic\_job and select lic\_check to be the task connected to this job. Finally, we click the **Create & Close** button.

The new job lic\_job has now been created, which can be seen in the Task Library window as shown in Figure 141 on page 156.

### 5.2.19 Schedule the TME 10 Job

To make the license checking routine automatic, we use the scheduler built into the TME 10 Framework to execute the job once a week.

We make sure that we get both the Task Library window and the TME 10 Desktop root window available at the same time, as shown in the next figure:

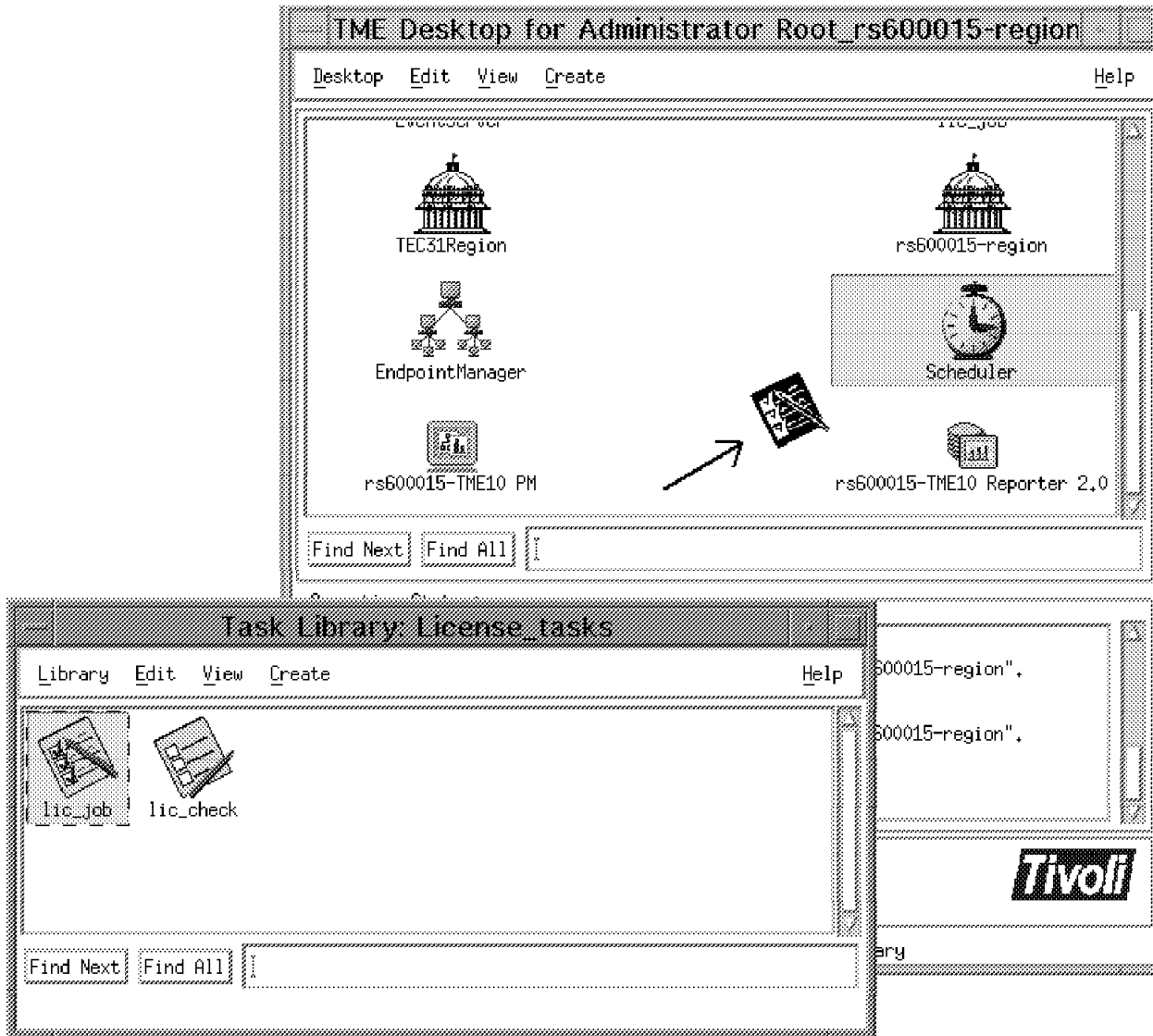


Figure 141. Schedule the TME 10 Job

To schedule the job, we point at the lic\_job job icon, hold down the left mouse button, drag the copy of the job icon over the Scheduler icon, and release the mouse button to drop the job icon on the Scheduler icon. As a result of this action, a window prompting for scheduling options will show up, as shown in the following figure:



**Edit Scheduled Job**

Job Name : lic\_job Job Id : 000003

Job Label : license\_check  Disable the Job.

Description:  
This is a job from the Task Library License\_tasks

Schedule Job For:  
Date: 8 / 10 / 97 Time: 4 : 00 AM PM  
Month Day Year Hour Minute

Repeat The Job:  
 Repeat the job indefinitely.  
 Repeat the job 0 times.  
The job should start every 1 week

When Job Complete:  
 Post Tivoli Notice: Available Groups...  
 Post Status Dialog on Desktop:  
 Send email to:  
 Log to File: File browser...  
Host :  
File :  
Set Retry/Cancel/Restriction Options...

Update & Close Close Help...

Figure 142. Scheduling Options

In the Job Label field, we enter license\_check. In the Description field we enter our description.

In the Schedule section, we select the date and time when we want the job to run for the first time. We select 4 AM on a Sunday.

We want the job to be run repeatedly every Sunday. In the Repeat section, we click the radio button next to Repeat the job indefinitely, enter the digit 1 in The job should start every field, and select **week** from the select button next to that field.

In the When Job Complete section, we click the radio button next to Post Tivoli Notice. We also select the Notice Group **Inventory** after pressing the **Available Groups** select button.

Finally we click the **Update & Close** button.

## 5.2.20 Summary

In the previous example, we have seen how to write simple scripts to access the TME 10 Inventory database, and how to use the information to build our own little application for checking program license usage.

Our script was written to run on AIX and to access Sybase, but the general idea is applicable also to other platforms and databases.

We also customized TME 10 Enterprise Console so the license administrator will see an event on his T/EC monitor if licenses are over-used.

The license checking script was saved as a Tivoli job, and was scheduled to run periodically.

---

## Chapter 6. Monitoring the TME 10 Inventory Database with TME 10 Distributed Monitoring

In this chapter we show how to use the new database monitors in TME 10 Distributed Monitoring 3.5 to monitor the RDBMS in which TME 10 Inventory stores its data.

---

### 6.1 Overview and Objective

We have TME 10 Inventory 3.1, Sybase for AIX, and TME 10 Distributed Monitoring 3.5 installed on our TMR server rs600015.

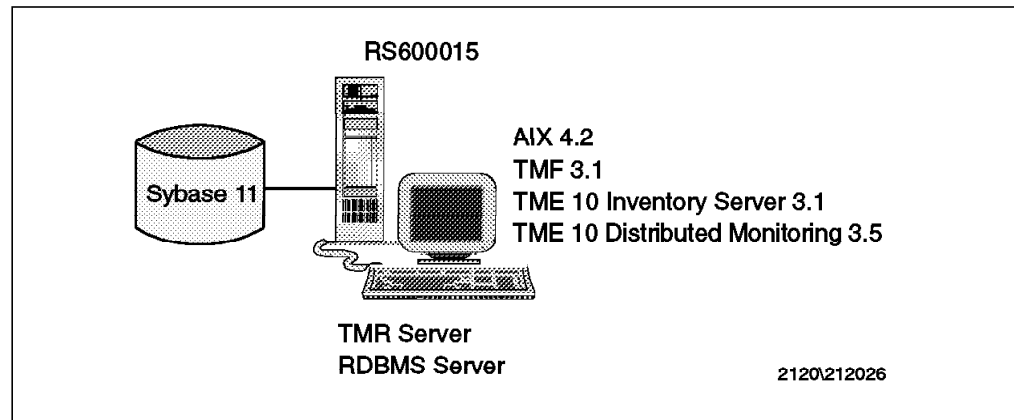


Figure 143. Environment for the Installation of TME 10 Distributed Monitoring, Sybase, and TME 10 Inventory

Since we experienced in other scenarios that the RDBMS often runs full when using TME 10 Inventory, we want to utilize TME 10 Distributed Monitoring to send a warning as soon as a certain threshold is exceeded in free space in the database.

The Sybase Monitoring collection that is delivered with TME 10 Distributed Monitoring 3.5 provides a set of monitors that can be used to monitor Sybase system parameters from the following categories:

- Sybase table space usage
- Sybase locks usage
- Sybase session usage
- Sybase engine status
- Sybase CPU an I/O usage
- Sybase message traffic
- Sybase disk reads and writes

All monitors use the ISQL facility to query the information needed from the Sybase RDBMS server.

The following figure shows this principle:

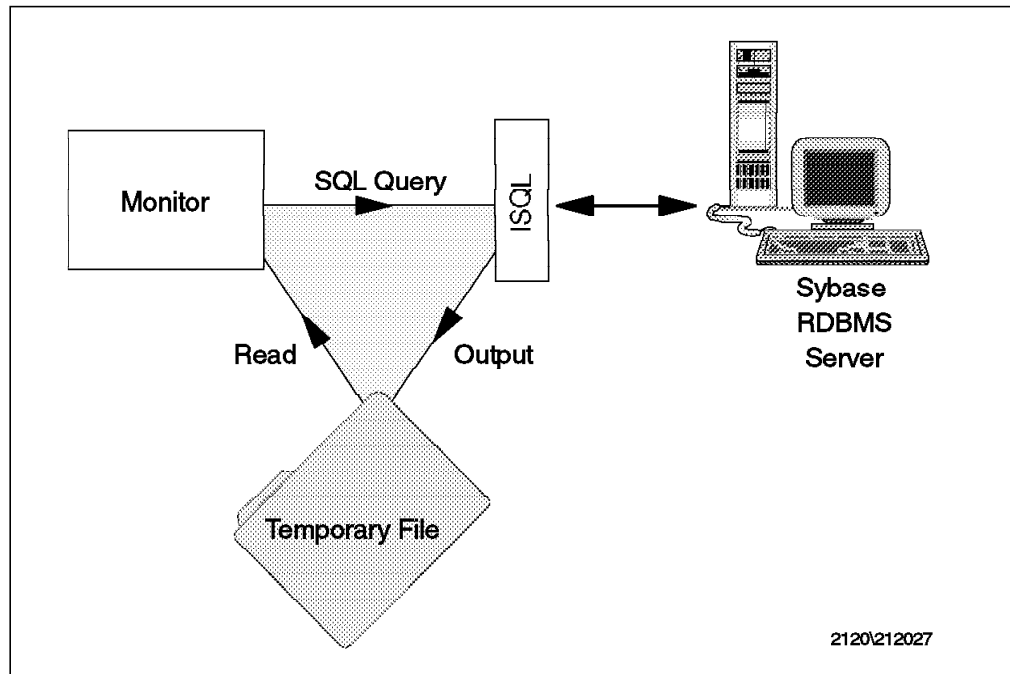


Figure 144. TME 10 Distributed Monitoring Talking to Sybase

Since ISQL is an interactive query facility, the output of a request is not directly usable by TME 10 Distributed Monitoring. Therefore, the output is first written to a temporary file that is then further processed.

## 6.2 Prerequisites

In order to be able to perform this scenario, we need TME 10 Distributed Monitoring 3.5, because this release contains additional monitors to monitor RDBMS systems. At least the following software needs to be installed to use the TME 10 Distributed Monitoring Sybase monitors in our example:

- TME 10 Framework 3.1 with the LCF Patch B or TME 10 Framework 3.2
- TME 10 Distributed Monitoring 3.5
- Sybase RDBMS Server (we use Version 11)

## 6.3 Installing the TME 10 Distributed Monitoring Sybase Monitoring Collection

After installing TME 10 Distributed Monitoring 3.5 we can install the monitoring collection for Sybase.

The following figure shows the TME 10 Install window:

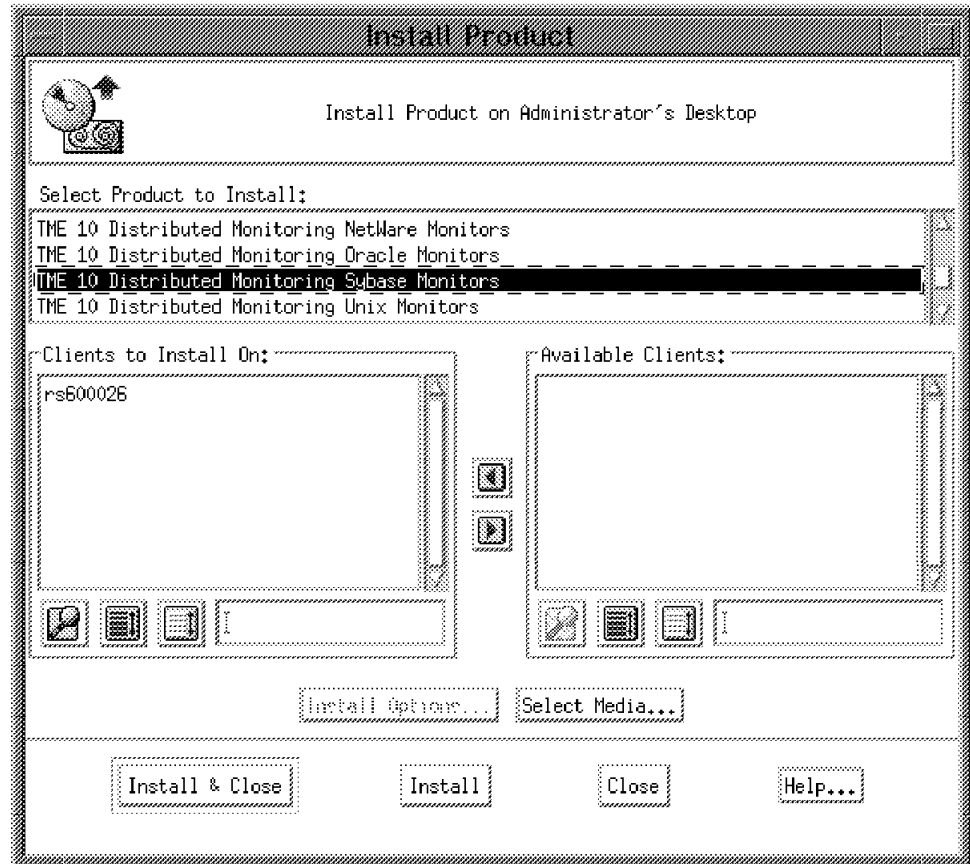


Figure 145. Installing Sybase Monitor Collection

After installing the TME 10 Distributed Monitoring Sybase Monitors the Sybase monitoring collection will be available to use, as we will show in the next section.

## 6.4 Setting Up a TME 10 Distributed Monitoring Profile to Monitor Sybase

The Sybase monitors can be created in a profile as any other TME 10 Distributed Monitoring monitor. We will perform the following detailed steps to set up the monitoring of the data space usage on the Sybase RDBMS database:

- Create a profile manager.
- Add subscribers.
- Create a profile.
- Edit the profile.
  - Add monitor
  - Edit monitor
  - Set monitoring schedule
  - Save monitor
- Distribute the profile.

So, first we want to create a profile manager in our existing policy region INVENTORY as shown in the next figure:

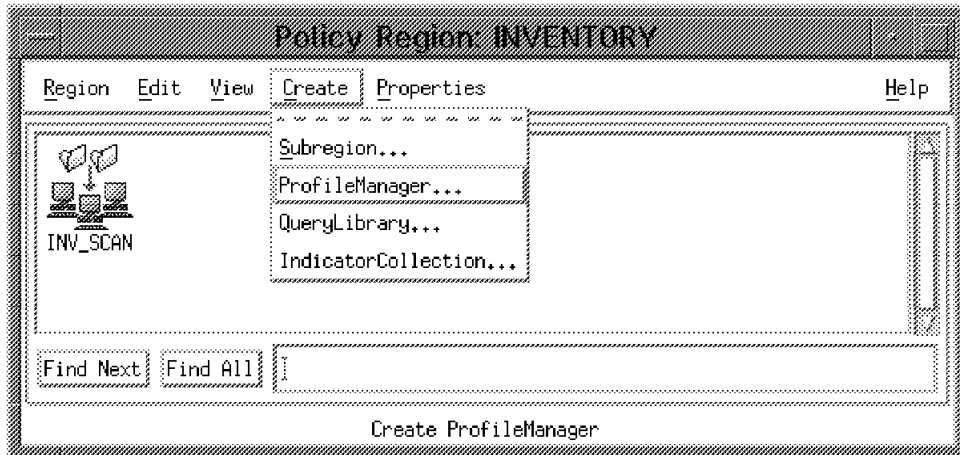


Figure 146. Create Profile Manager

We select **Create** from the menu bar. We then select **ProfileManager...** from the pull-down and get the window shown in the next figure:

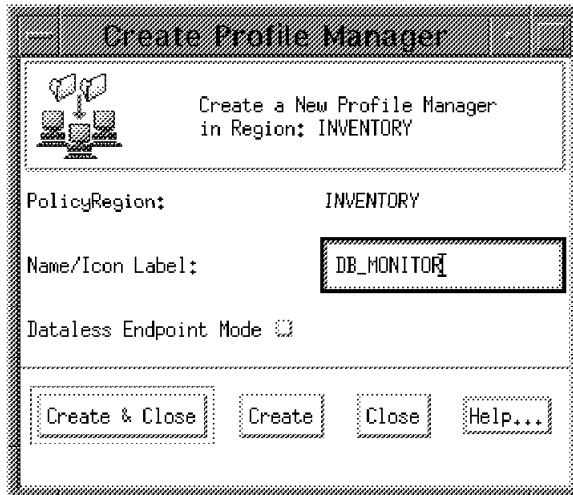


Figure 147. Give the Profile Manager a Name

We enter DB\_MONITOR as the name, and click the **Create & Close** button.

The profile manager will be created, and an icon will show up in our INVENTORY policy region. We then double click on our new **DB\_MONITOR** icon to open it. We will get an empty Profile Manager window.

We want to add a profile and a subscriber. We start with the subscriber. We select **Profile Manager** from the menu bar, and then **Subscribers....** We add our Sybase RDBMS server host as the subscriber, and the Profile Manager window will look as shown in the following picture:

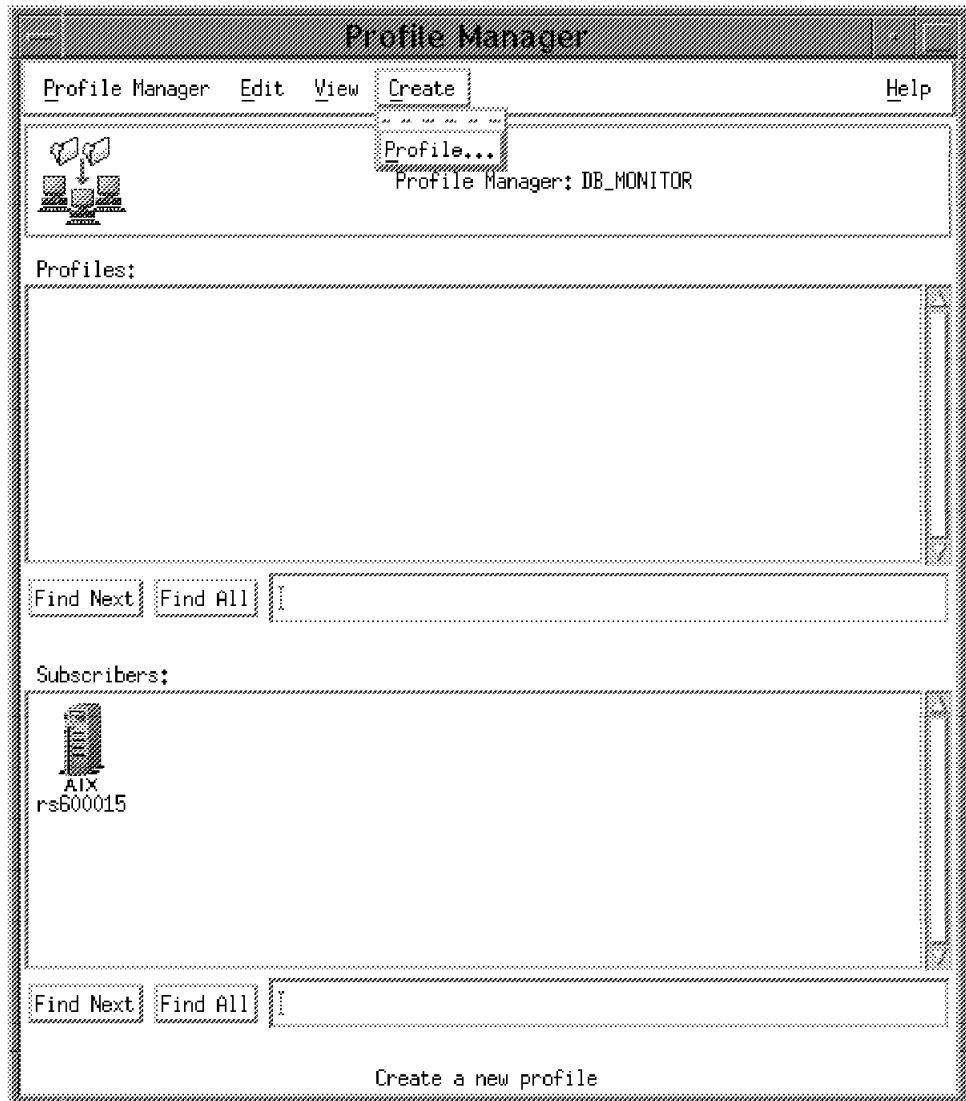


Figure 148. Profile Manager with Subscriber

We now want to add the profile to contain our database monitors. We select **Create** from the menu bar, and then **Profile...** from the pull-down menu. That will result in the following window being displayed:



Figure 149. Create Profile

We enter the name DB\_MON\_PRO. We select **SentryProfile** and then click the **Create & Close** button. The profile is created, and the profile icon shows up in the Profile Manager window as shown in the next figure:



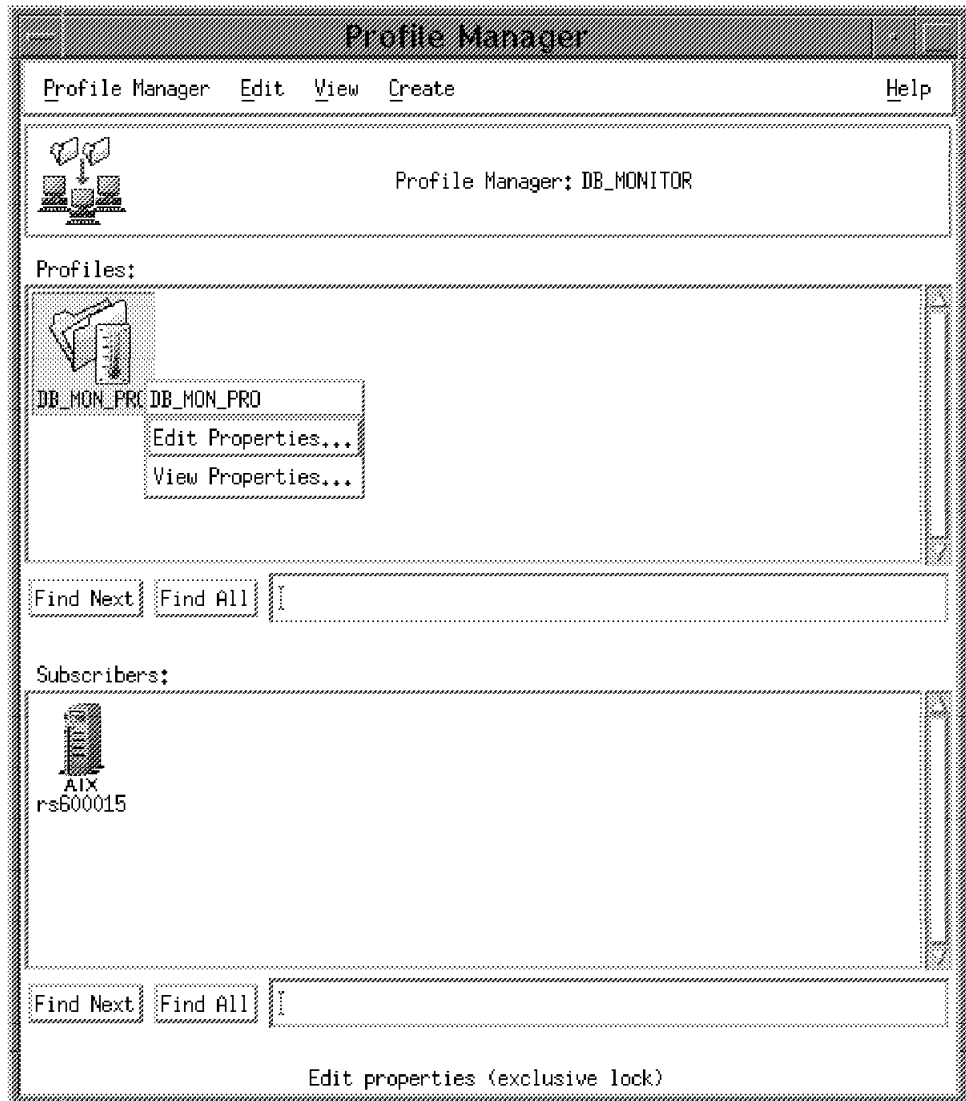


Figure 150. Profile Manager with Profile and Subscriber

Now we must edit the profile to add monitors and set their thresholds. We select **Edit Properties...** from the icon pull-down menu of the profile, and the Profile Properties window will be available as shown in the next figure:

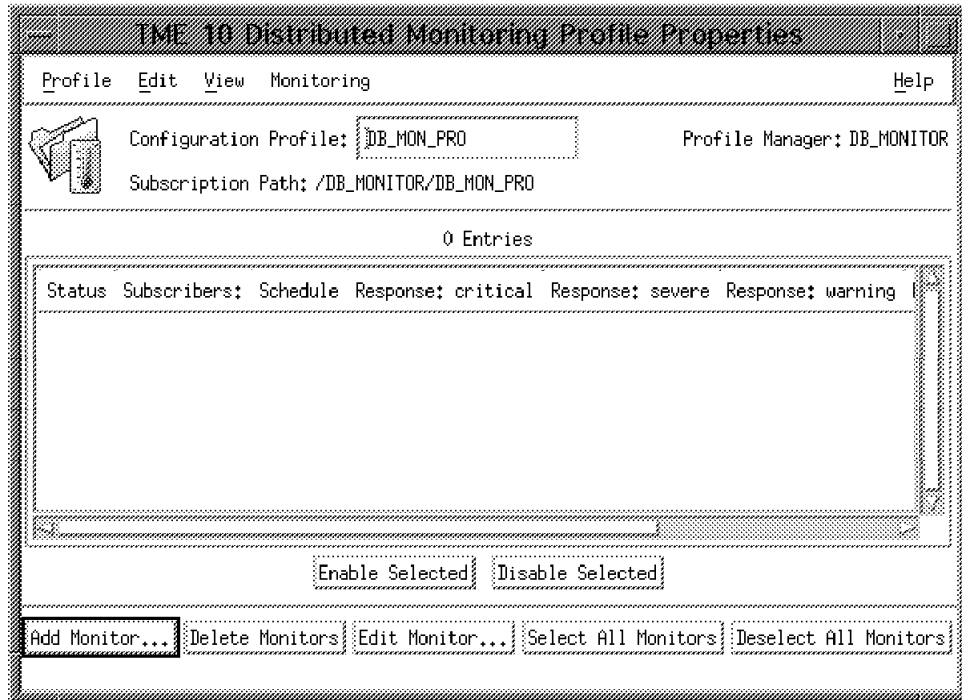


Figure 151. TME 10 Distributed Monitoring Profile Properties

The profile can contain various monitors, not only database monitors. The profile can be described as a collection of monitors that are to be distributed to the same subscriber or subscribers.

To add a monitor we click the **Add Monitor...** button. We will get the window shown in the next figure:

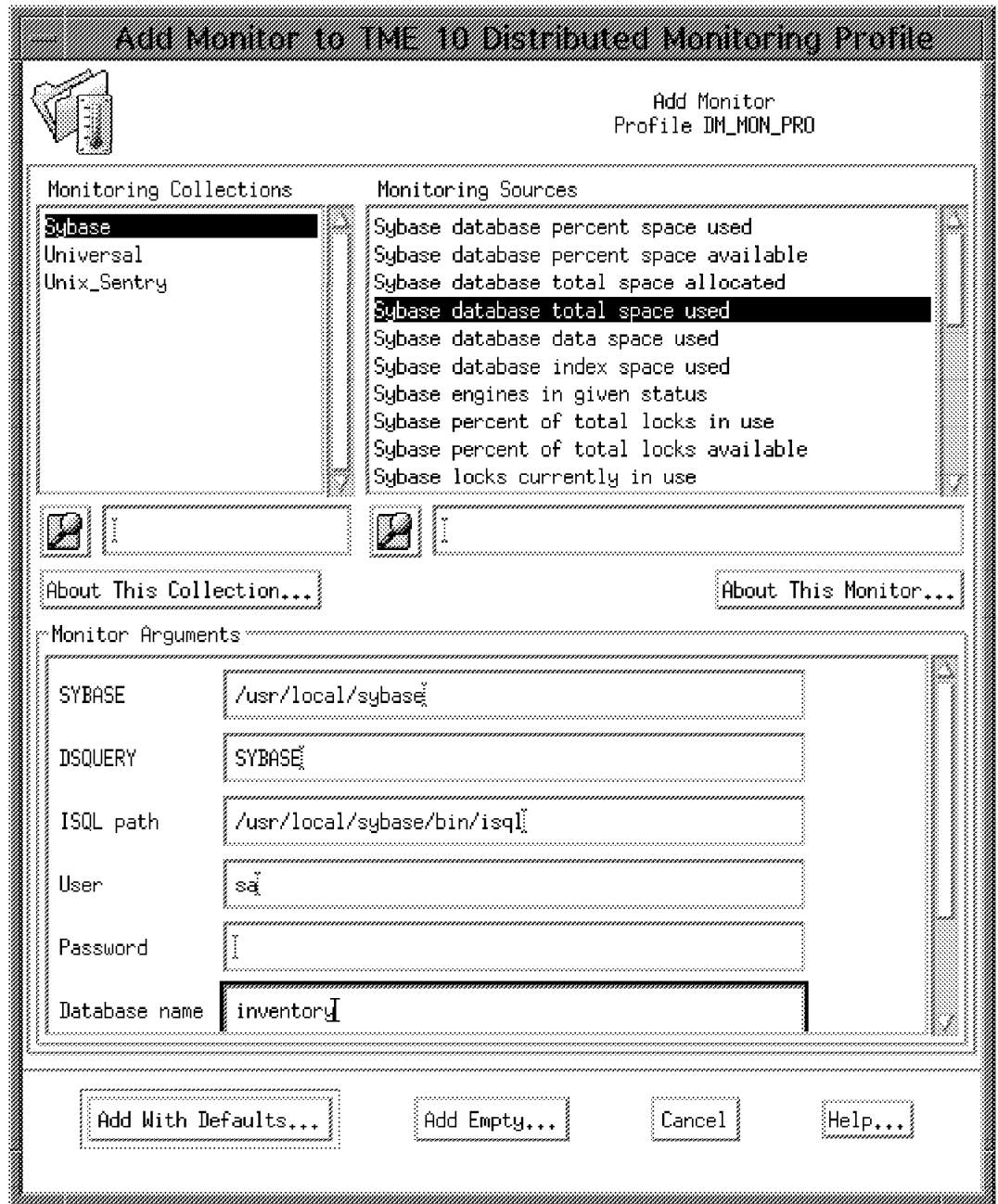


Figure 152. Add Monitor to TME 10 Distributed Monitoring Profile

First, we select from which monitoring collection we want to pick our monitor. We select the **Sybase** Monitoring Collection, and all the available Sybase monitors will show up in the list of Monitoring Sources.

We want to monitor the database usage. In our scenario we have a very small database, and we want a warning when the total space used has increased above a certain number of megabytes. We select the monitor **Sybase database total space used**. As a result we now get input fields for some additional monitor arguments in the lower part of the window.

We supply the following values:

- SYBASE is the environment variable that points to the Sybase installation directory. We enter /usr/local/sybase.
- DSQUERY is the environment variable that identifies the Sybase server. It is set during Sybase installation, and can be found in the \$SYBASE/interfaces file. We enter SYBASE.
- ISQL path is the full path and file name of where to find the isql executable. We enter /usr/local/sybase/bin/isql.
- User is the user ID for accessing ISQL. We use the Sybase administrator user and enter sa.
- Password is the password for the user ID above. Our Sybase RDBMS was installed without giving user ID sa a password, so we leave the field empty, which is accepted. It is advisable though to set a password. This can be done within isql using the stored procedure sp\_password.
- Database name is the name of the database that we want to monitor. We enter inventory.

Finally, we click the **Add Empty...** button, and get the Edit Monitor window shown in the next figure:

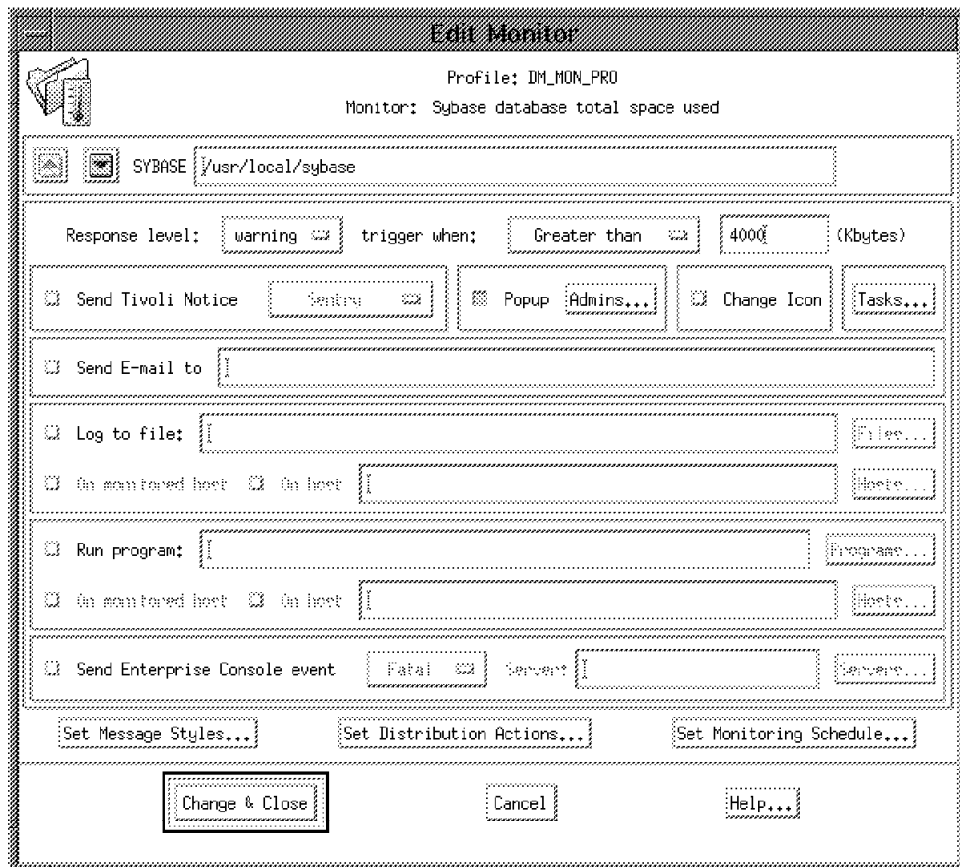


Figure 153. Edit Monitor

At the top of the Edit Monitor window, we can change the monitor arguments that we entered in the previous panel. What we now want to do is to define threshold values for our monitor, what kind of indication we want to see, and when we want the monitor to do its job.

From the select button next to Response level, we select **warning**. From the select button next to trigger when, we select **Greater than**. First, we want to test that our monitor works, so we define a value for total space used, that we think is below the actual figure. So, we enter 4000.

A simple kind of indication is to get a pop-up on the administrator's desktop. We click the **Popup** radio button. We also click the **Admins...** select button, and select our Tivoli root\_administrator from the list.

Then we click the **Set Monitoring Schedule...** button to get the following window:

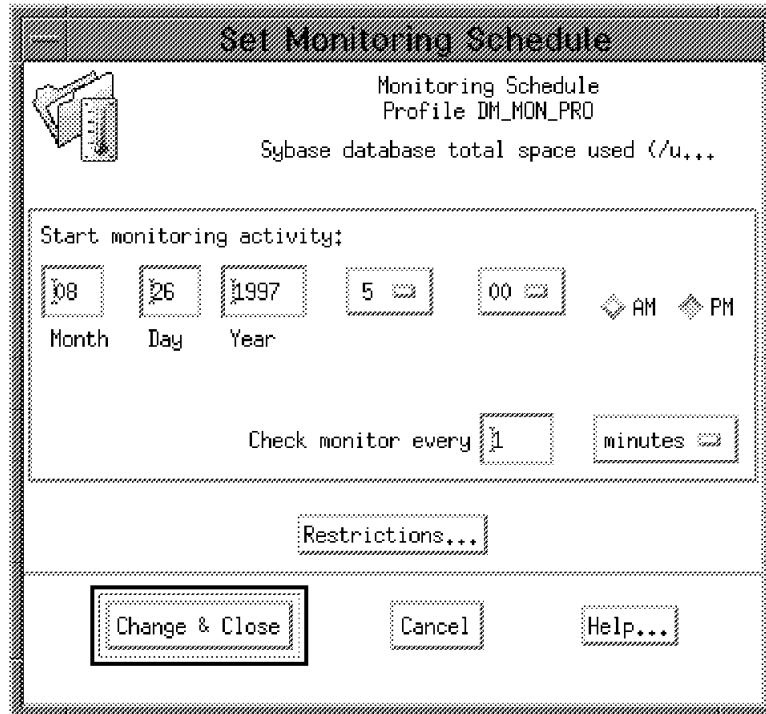


Figure 154. Set Monitoring Schedule

First, we want to verify that the monitor works, so we select that we want to check the monitor every minute. We then click the **Change & Close** button.

Back on the Edit Monitor window in Figure 153 on page 168, we click the **Change & Close** button to finish the definitions for this monitor. We will now have a line representing our monitor in the Profile Properties window as shown in the next figure:

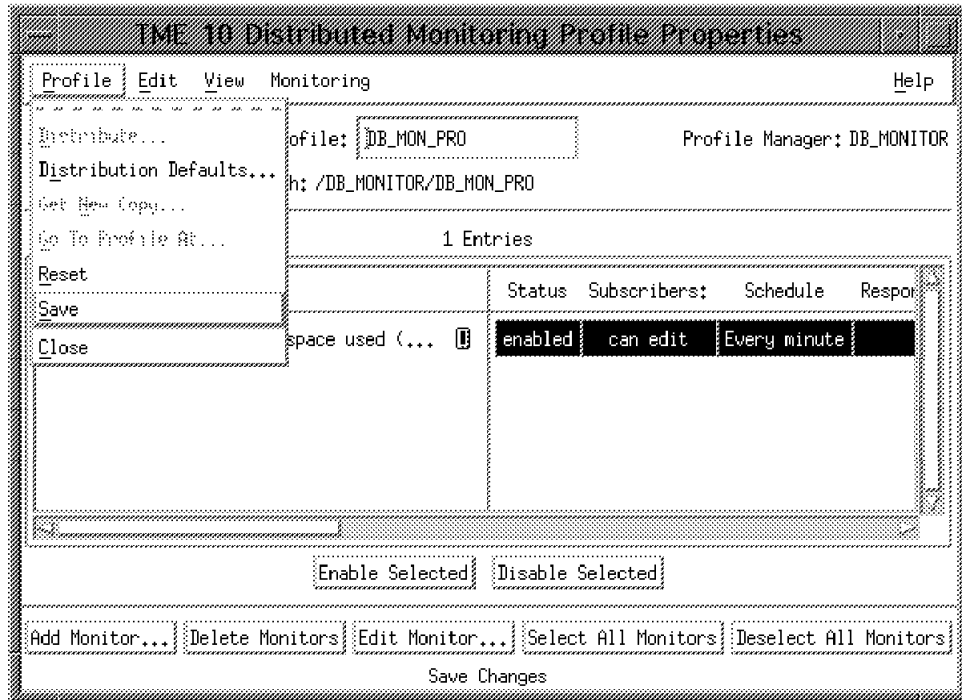


Figure 155. Save and Close Profile

Our newly defined monitor is in enabled state. We now select **Profile** from the menu bar. From the pull-down we first select **Save** then **Close**.

To have the monitor really do its job, we must now distribute the profile containing the monitor to the corresponding subscriber. So, in the Profile Manager window shown in Figure 150 on page 165, we drag the profile icon DB\_MON\_PRO onto the subscriber icon rs600015.

After a while we get the pop-up on our desktop as shown in the next figure:

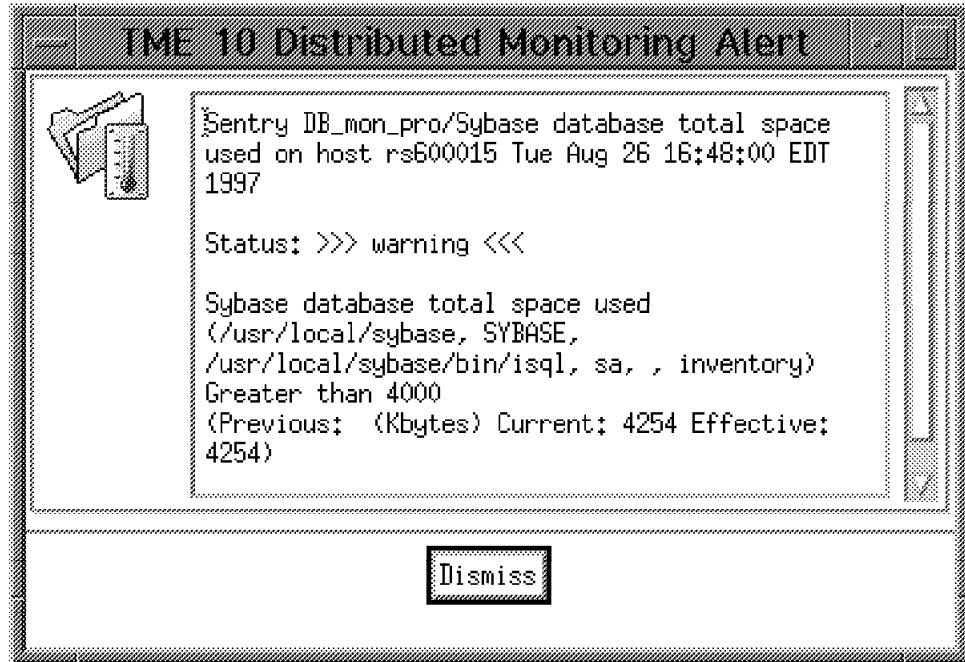


Figure 156. Pop-up from the Monitor

So, our small test value of 4000 KB (kilobytes) was low enough to trigger the monitor. As we can see, the current usage is 4254 KB.

As described in Figure 144 on page 160 the monitor uses ISQL to retrieve the database information, and stores the information in a temporary file. With the Sybase database total space used monitor we will notice that the file `db_spc_used.out` is created in the `/tmp` directory.

The monitor seems to work, so we can again edit it and change the schedule, so we do not get this pop-up every minute. A suitable interval could perhaps be once every hour. We will also change the value of Kbytes from 4000 to a value that matches the real limit that we want to apply for total space used.

This scenario showed just one of the TME 10 Distributed Monitoring Sybase monitors. The Sybase monitor collection contains more than 20 monitors to monitor session, CPU usage, disk reads, etc.

There is also a monitoring collection for Oracle, that supplies equivalent monitors for an Oracle RDBMS environment.





## Chapter 7. Setting Up TME 10 LAN Access for Use with TME 10 Inventory

In this chapter, we describe how to set up TME 10 LAN Access to retrieve inventory information from a desktop management suite and store that information in TME 10 Inventory.

This chapter is not meant as a full description of TME 10 LAN Access or the installation of the different software components. In this chapter, we will not repeat a description of every single step. For more details, see the following redbook:

- *Systems Management from an NT Server Point of View*, SG24-4723

### 7.1 Overview and Objective

The objective of this chapter is to show how inventory data of an existing NetFinity 5.0 environment can be collected and integrated in the TME 10 Inventory database.

We set up an environment as shown in the following figure:

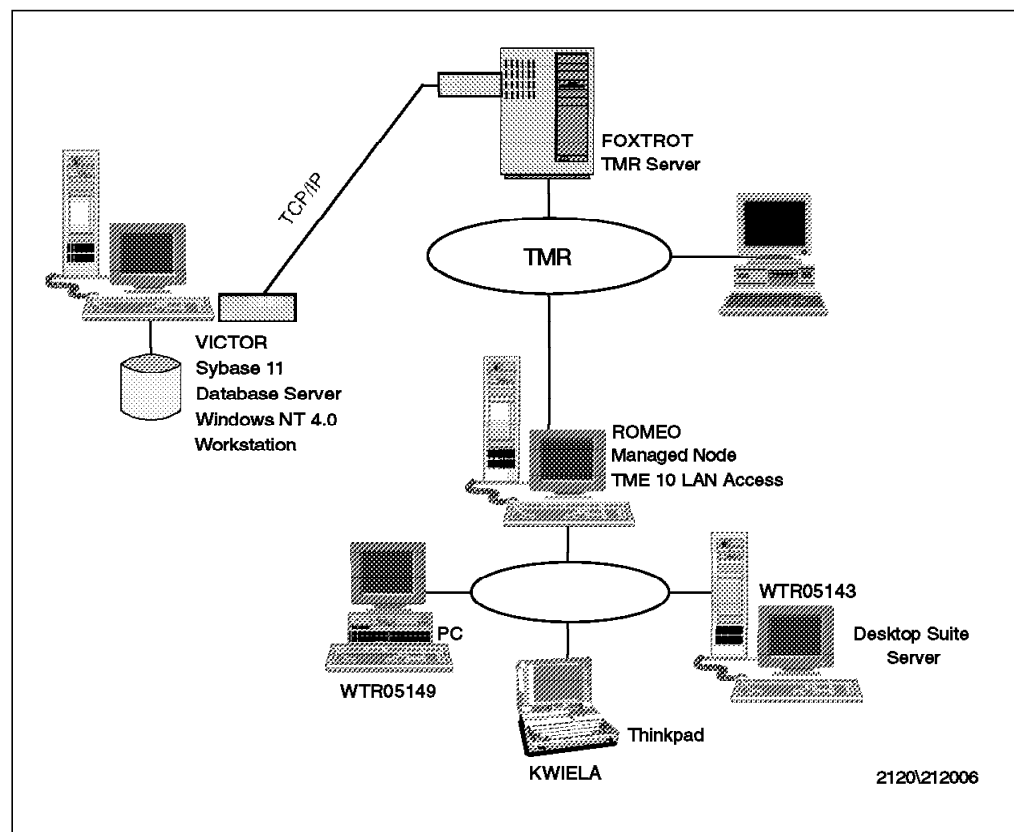


Figure 157. TME 10 LAN Access Environment

How does it all work?

- MPM providers use the Multi-Platform Management application programming interface (MPM API). The MPM API allows one management application to communicate with another management application by presenting a common API. The MPM API is an open specification and is available for use without royalties.
- Step 1: The Management Services Broker (MSB) utilizes the MPM API to provide a common interface for bi-directional communication with various PC administration tools.
- Step 2: The MSB processing engine gathers data from and distributes work to the various PC administration packages.
- Step 3: MSB uses Tivoli CORBA objects for direct integration with TME 10.

In more detail, we perform the following steps:

- Install the NetFinity 5.0 environment and also use other machines that are available in the network, that is, all other machines that have the NetFinity 5.0 services installed and running.
- Install TME 10 LAN Access Provider on the NetFinity 5.0 server.
- Install TME 10 LAN Access and TME 10 LAN Access Provider in the TMR.
- Create and configure a NetFinity 5.0 policy region in the TMR.
- Create and configure a LanAccessSite in the new NetFinity 5.0 region.
- Run a basic inventory scan.

We used the following equipment in our environment:

***Foxtrot***

Hardware : PS/2 Model 95, Pentium 60, 80 MB RAM, 1 GB Disk  
OS : Windows NT Server 4.0 with SP 2  
Protocol : TCP/IP  
Software : TME 10 Framework 3.1 Rev C  
TME 10 Inventory 3.1, Rev A  
TME 10 LAN Access, Version 1.1 (Beta)

***Romeo***

Hardware : PS/2 Model 95, Pentium 60, 64 MB RAM, 1 GB Disk  
OS : Windows NT Server 4.0 with SP 2  
Protocol : TCP/IP  
Software : TME 10 Inventory 3.1, Rev C  
TME 10 Inventory 3.1, Rev A  
TME 10 LAN Access, Version 1.1 (Beta)  
TME 10 LAN Access Provider for NT, Version 1.1 (Beta)

***Victor***

Hardware : PS/2 Model 95, Pentium 60, 32 MB RAM, 1 GB Disk  
OS : Windows NT Workstation 4.0  
Protocol : TCP/IP  
Software : Sybase Server Version 11

### **WTR05143**

Hardware : PS/2 Model 95, Pentium 60, 32 MB RAM, 2 \* 540 MB Disks  
OS : OS/2 Warp 4.0  
Protocol : TCP/IP, NetBIOS  
Software : NetFinity 5.0 Server Manager  
TME 10 LAN Access Provider for OS/2, Version 1.1 (Beta)

### **KWIELA**

Hardware : ThinkPad 760 ED, 48 MB RAM, 1.4 GB Disk  
OS : OS/2 Warp 4.0  
Protocol : TCP/IP, NetBIOS  
Software : NetFinity 5.0 Services Version 5.0

### **Several Other Machines in the Network**

Hardware : Misc  
OS : OS/2 Warp 4.0, OS/2 Warp Connect,  
Windows NT 4.0, Windows 95  
Protocol : TCP/IP, NetBIOS  
Software : NetFinity 5.0 Services and Version 4.0

---

## **7.2 Restrictions When Using TME 10 LAN Access with TME 10 Inventory**

The restrictions when using TME 10 LAN Access with TME 10 Inventory are:

- Universal times are not supported; time zones are ignored.
- If multiple TME 10 LAN Access Providers are operating on multiple NetFinity 5.0 managers and feed information to a single NT managed node running TME 10 LAN Access, many clients will appear multiple times (once through each manager). This phenomenon could cause problems. There is no mechanism established to filter multiple client occurrences.
- A manager of managers concept using several levels of NetFinity 5.0 managers is not supported.

---

## **7.3 Mapping for Discovered Inventory**

A mapping mechanism is necessary to translate the fields discovered by the LAN management tool into fields managed by TME 10 Inventory. Although the LAN tools all discover similar information, they store it in different fields according to their data model. For example, the field DISC\_SYSTEM\_TCP\_HOSTNAME of MPM translates to DNS Machine Name in SMS although both store the same information.

As the MPM definition is in some cases a subset and in other cases a superset of information provided by the different LAN tools, there is no direct mapping.

---

## **7.4 Installing TME 10 LAN Access**

The installation of TME 10 LAN Access is divided into 3 steps:

1. Install TME 10 LAN Access Provider on the NetFinity 5.0 server.
2. Install TME 10 LAN Access Provider on the managed node romeo, which we defined in our environment as the gateway to the NetFinity 5.0 server.

3. Install TME 10 LAN Access in the TMR.

**Note**

Be sure to do the installation steps in the same order as specified above.

As mentioned before, we will not repeat an explanation of the detailed installation processes. If you need to know more details, we would like you to refer to the redbooks that we listed in the first section of this chapter.

### 7.4.1 Installing TME 10 LAN Access Provider on the NetFinity 5.0 Server

Before you begin the installation, make sure that NetFinity 5.0 is installed and running on the node that you want to be the TME 10 LAN Access Provider. We selected WTR05143 to be the TME 10 LAN Access Provider and the NetFinity 5.0 Server Manager. To install the TME 10 LAN Access Provider, run the **setup.exe** from the source where the installation code resides.

The following window will appear:

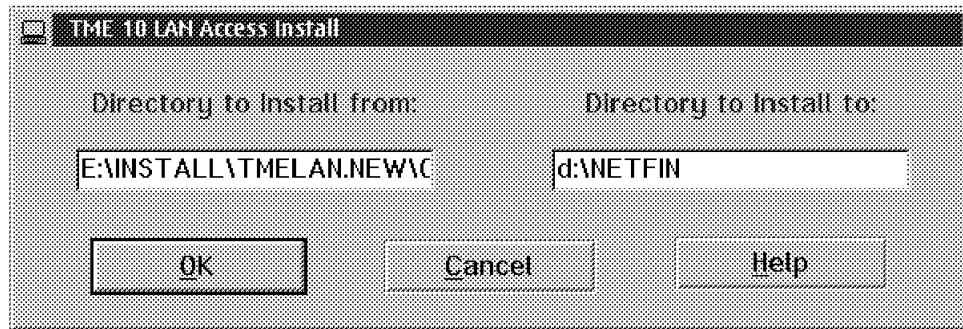


Figure 158. TME 10 LAN Access Install Window

Click on **OK** if you agree with the drive and the directory that the installation program offers by default. Otherwise change it accordingly.

You will see the following window on your screen:

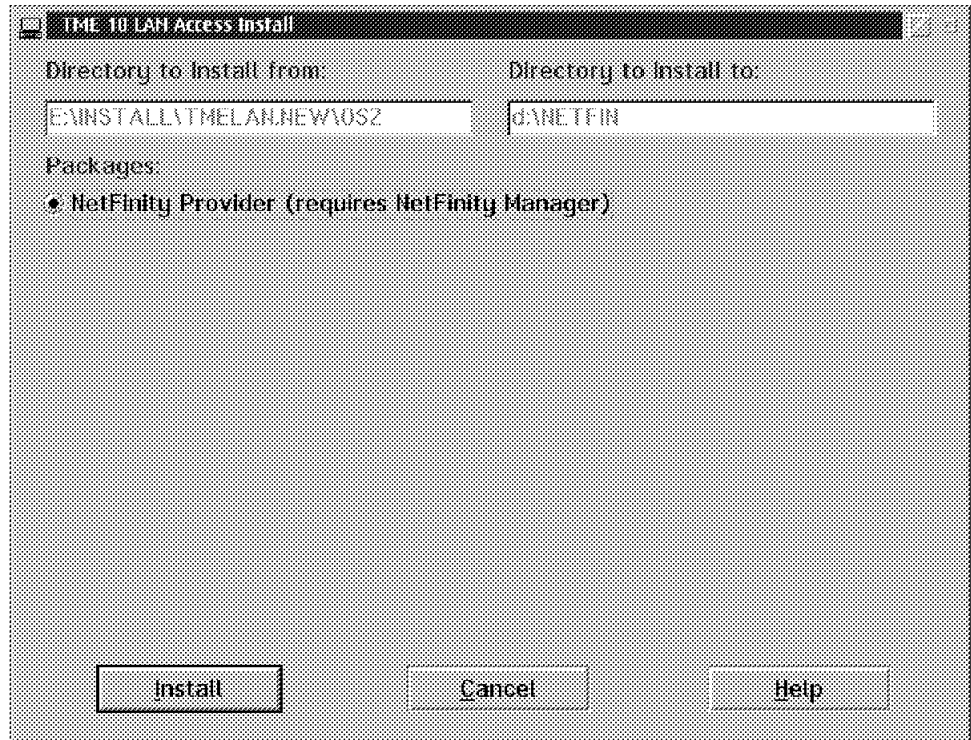


Figure 159. TME 10 LAN Access Install Window

Click on **Install** to start the installation.

The next figure shows the status of the installation.

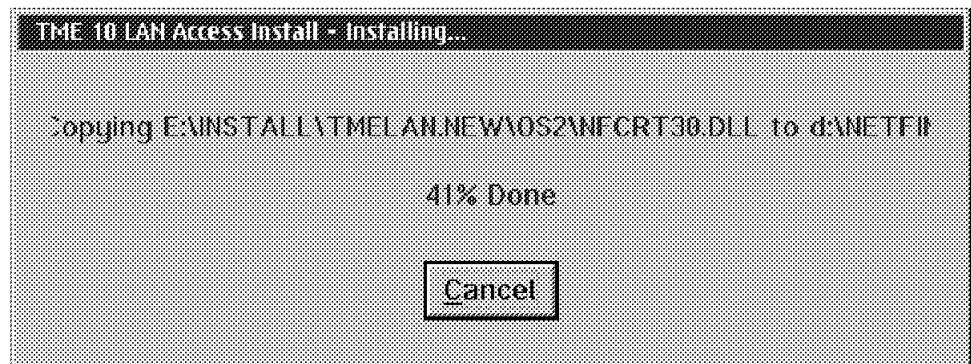


Figure 160. TME 10 LAN Access Install - Installing... Window

When all necessary files are installed, the following window will appear:

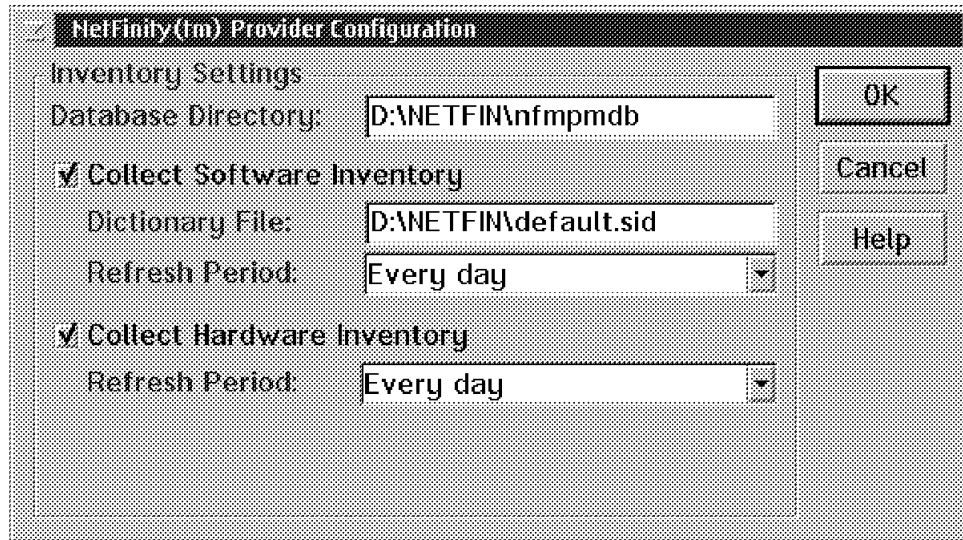


Figure 161. NetFinity Provider Configuration Window

The values you see are the default values offered during the initial install.

The Database Directory specifies the directory used to hold the collected hardware and software inventory data (typically about 50 KB per client system). If Collect Software Inventory is enabled, there will be a data file per client with the extension of SFT generated if it is your initial installation. Otherwise the files will be updated every time a new scan is started. If Collect Hardware Inventory is selected a data file with the extension of HDW is generated or updated accordingly.

The Refresh Period specifies the time period between re-scans of software or hardware inventory of NetFinity 5.0 clients.

The following values can be selected:

- Every hour, 4 hours, 8 hours, 12 hours
- Every day, 2 days, 4 days
- Every week, 2 weeks, 4 weeks, 8 weeks, 16 weeks

Click on **OK** when you are satisfied with the information in the window or accept the default values.

When the installation is finished, the next window will appear:

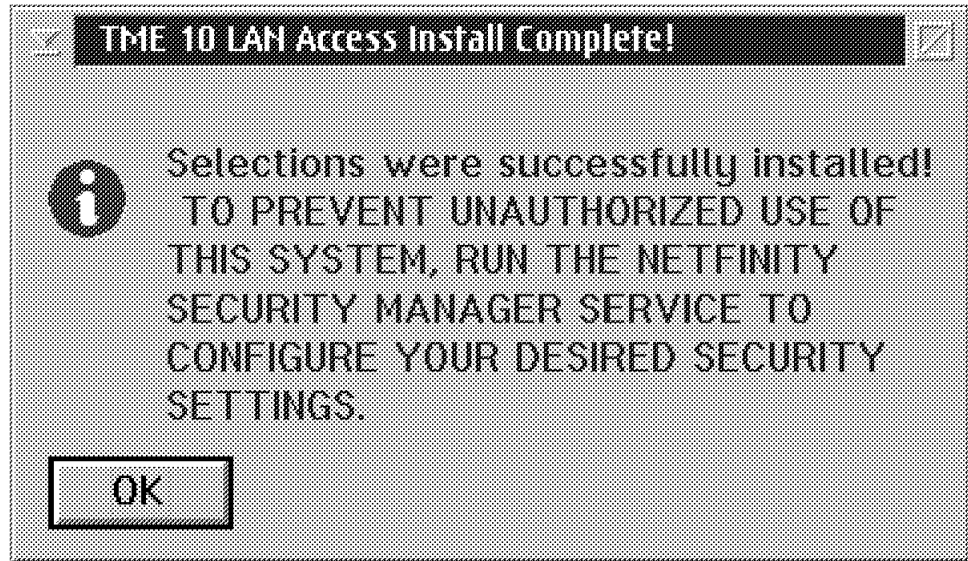


Figure 162. TME 10 LAN Access Install Complete! Window

Click on **OK** to complete the installation of the TME 10 LAN Access Provider.

As you can see in the next figure, there is a new TME 10 LAN Access folder generated on your OS/2 Desktop:

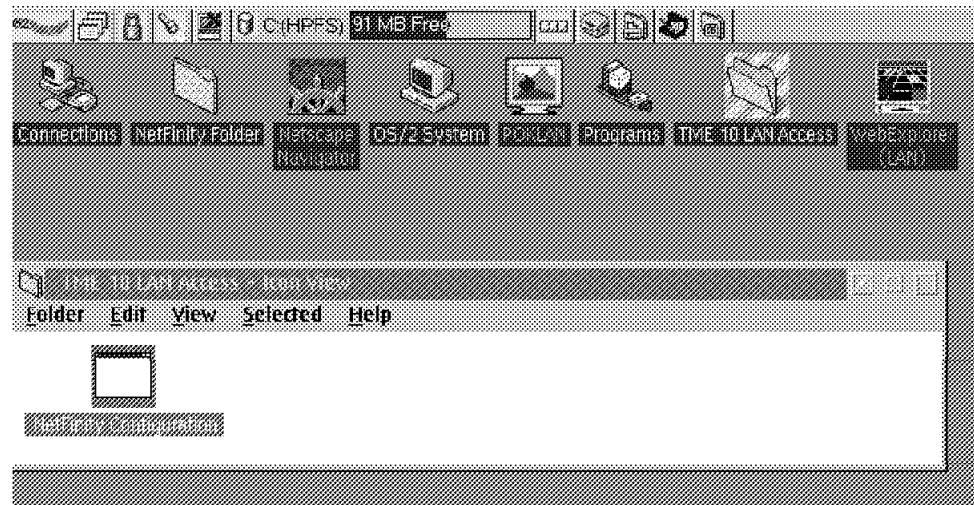


Figure 163. OS/2 Desktop Window

When you open this folder you see the NetFinity 5.0 Configuration program, that you already know from Figure 161 on page 178. By starting this program, you can change the previously discussed configuration after the initial installation whenever you have a need to do so.

**Note**

Be prepared that immediately after the first installation an initial inventory icon is started, which represents the first scan of your network. This may take some time, depending of the number of NetFinity 5.0 clients in your network. So, do not perform a system shutdown after TME 10 LAN Access Provider installation. If you do, the OS2.INI files might not be updated. Therefore, the desktop might be incomplete and you will have to run a reinstall.

If you want to reinstall the TME 10 LAN Access Provider do the following:

- Execute NETFBASE.exe SHUTDOWN from the command line.
- Delete the TME 10 LAN Access folder from the desktop.
- Reinstall TME 10 LAN Access Provider.

As discussed before, after the initial scan has completed, you can find the files for the Hardware and Inventory data in the NFMPMDB directory. Following is an example of the inventory data for the client KWIELA, which shows a few selected sections of the content of the data files generated by the NetFinity 5.0 TME 10 LAN Access Provider.

On the seventh line, counted from the top, for example you can see the serial number of the scanned machine, which happens to be 551N11L.

Look at the fourth line, counted from the bottom, there you can see the network name of the machine which is KWIELA.



```
DMI/DMTF]Operating System]001/0/1=0
DMI/DMTF]Operating System]001/0/2="OS/2"
DMI/DMTF]Operating System]001/0/3="4.00"
DMI/DMTF]Operating System]001/0/4=1
DMI/DMTF]Logical Memory]001/1=645
DMI/DMTF]Logical Memory]001/3=48128
DMI/DMTF]System Enclosure]002/2="551N11L "
DMI/DMTF]ComponentID]001/0/0=0
DMI/DMTF]ComponentID]001/0/4="551N11L "
DMI/DMTF]Bus Global Table]001/1/1=1
DMI/DMTF]Bus Global Table]001/1/2=5
DMI/DMTF]System Slots]003/0/1=0
DMI/DMTF]System Slots]003/0/2="PCI"
DMI/DMTF]System Slots]003/0/3=5
DMI/DMTF]Video]002/0/12=8
.
.
.
DMI/DMTF]Keyboard]003/1="CODEPAGE 850 COUNTRY GR SUBCOUNTRY 129 "
DMI/DMTF]Keyboard]003/2="88/89 key Enhanced Keyboard"
DMI/DMTF]Processor]003/0/1=0
DMI/DMTF]Processor]003/0/2=3
DMI/DMTF]Processor]003/0/3=11
DMI/DMTF]Processor]003/0/4="Pentium(tm)"
DMI/DMTF]Processor]003/0/5=133
DMI/DMTF]Processor]003/0/6=133
DMI/DMTF]Processor]003/1/1=1
DMI/DMTF]Processor]003/1/2=4
DMI/DMTF]Processor]003/1/3=11
DMI/DMTF]Processor]003/1/4="Pentium(tm)"
DMI/DMTF]Processor]003/1/5=133
DMI/DMTF]Processor]003/1/6=133
DMI/DMTF]System BIOS]001/0/1=0
DMI/DMTF]System BIOS]001/0/2="IBM"
DMI/DMTF]ComponentID]001/0/1="IBM"
DMI/DMTF]System BIOS]001/0/3="IBM ThinkPad 760"
DMI/DMTF]ComponentID]001/0/2="IBM ThinkPad 760"
DMI/DMTF]System BIOS]001/0/8=1997-01-20T00:00:00.000+0000
DMI/DMTF]Mouse]003/1=4
DMI/DMTF]Mouse]003/3=2
DMI/DMTF]Disk Controller]002/0/1=0
DMI/DMTF]Disk Controller]002/0/4=0
DMI/DMTF]Disk Controller]002/0/5="ST-506 CAM"
.
.
.
DMI/DMTF]General Information]001/1="KWIELA"
DMI/DMTF]General Information]001/3=NULL
DMI/DMTF]General Information]001/2=NULL
DMI/DMTF]General Information]001/4=NULL
```

Figure 164. Hardware Inventory Data

```

DMI/DMTF] ComponentID] 001/100/0=100
DMI/DMTF] ComponentID] 001/100/1="V Communications"
DMI/DMTF] ComponentID] 001/100/2="Unpacker"
DMI/DMTF] ComponentID] 001/100/3=NULL
DMI/DMTF] Software Component Information] 002/100/0=100
DMI/DMTF] Software Component Information] 002/100/1=NULL
DMI/DMTF] Software Component Information] 002/100/2=NULL
DMI/DMTF] Software Component Information] 002/100/3=NULL
DMI/DMTF] Software Component Information] 002/100/7="Unpacker"
DMI/DMTF] ComponentID] 001/101/0=101
DMI/DMTF] ComponentID] 001/101/1="Microsoft"
DMI/DMTF] ComponentID] 001/101/2="Backup"
DMI/DMTF] ComponentID] 001/101/3=NULL
DMI/DMTF] Software Component Information] 002/101/0=101
DMI/DMTF] Software Component Information] 002/101/1=NULL
DMI/DMTF] Software Component Information] 002/101/2=NULL
DMI/DMTF] Software Component Information] 002/101/3=NULL
DMI/DMTF] Software Component Information] 002/101/7="Backup"
.
.
.
DMI/DMTF] ComponentID] 001/107/0=107
DMI/DMTF] ComponentID] 001/107/1="IBM"
DMI/DMTF] ComponentID] 001/107/2="NetFinity 5.0 Services for OS/2"
DMI/DMTF] ComponentID] 001/107/3="5.0"
DMI/DMTF] Software Component Information] 002/107/0=107
DMI/DMTF] Software Component Information] 002/107/1="5"
DMI/DMTF] Software Component Information] 002/107/2="0"
DMI/DMTF] Software Component Information] 002/107/3=NULL
DMI/DMTF] Software Component Information] 002/107/7="NetFinity 5.0 Services for OS/2"
.
.
.
DMI/DMTF] ComponentID] 001/153/0=153
DMI/DMTF] ComponentID] 001/153/1="Microsoft"
DMI/DMTF] ComponentID] 001/153/2="Clipboard Viewer"
DMI/DMTF] ComponentID] 001/153/3=NULL
DMI/DMTF] Software Component Information] 002/153/0=153
DMI/DMTF] Software Component Information] 002/153/1=NULL
DMI/DMTF] Software Component Information] 002/153/2=NULL
DMI/DMTF] Software Component Information] 002/153/3=NULL
DMI/DMTF] Software Component Information] 002/153/7="Clipboard Viewer"
DMI/DMTF] ComponentID] 001/154/0=154
DMI/DMTF] ComponentID] 001/154/1="Microsoft"
DMI/DMTF] ComponentID] 001/154/2="Character Map"
DMI/DMTF] ComponentID] 001/154/3=NULL
DMI/DMTF] Software Component Information] 002/154/0=154
DMI/DMTF] Software Component Information] 002/154/1=NULL
DMI/DMTF] Software Component Information] 002/154/2=NULL
DMI/DMTF] Software Component Information] 002/154/3=NULL
DMI/DMTF] Software Component Information] 002/154/7="Character Map"

```

Figure 165. Software Inventory Data

After the initial scan has finished you should check that there is a pair of data files generated and stored in your NFMPMDB directory for every NetFinity 5.0 client. If some or all data are missing, check the security settings of the NetFinity 5.0 clients. Make sure that a user ID <PUBLIC> for each client exists

and that at least the following services are made available to the user ID <PUBLIC>:

- Software Inventory
- System Information

How to do this is shown in the following figure:

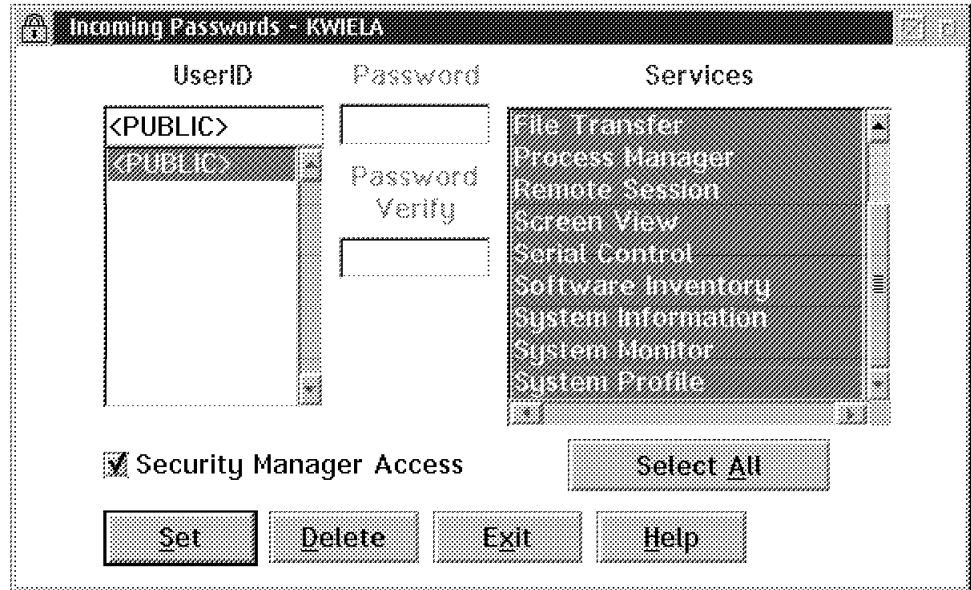


Figure 166. Incoming Passwords - KWIELA Window

You will get to this window when you double-click on the NetFinity folder that you can see in Figure 163 on page 179. Then double-click on the TME 10 NetFinity Service Manager, which opens a new folder that shows you the Security Manager. If you double-click on the Security Manager you will see Figure 166.

Try to connect your NetFinity 5.0 client from a NetFinity 5.0 server through the NetFinity 5.0 Remote System Manager. If you receive a screen prompting you to enter user ID and Password, as shown in the following figure, the user ID <PUBLIC> has been removed totally on this client. You have to re-enter the user ID <PUBLIC> and set the Services as explained before.

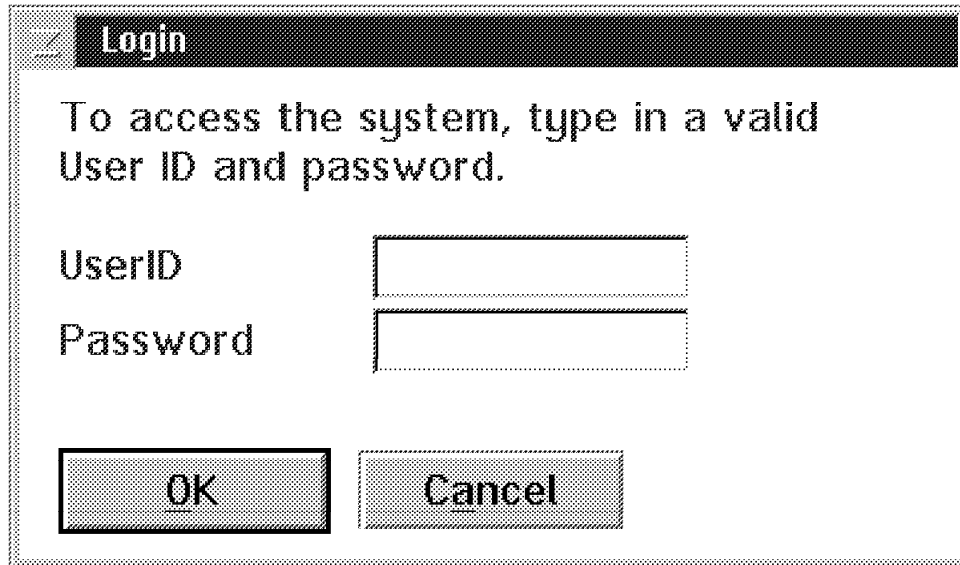


Figure 167. Login Window

#### 7.4.2 Installing TME 10 LAN Access Provider in the TMR

Although our managed node romeo is not really the TME 10 LAN Access Provider, the TME 10 LAN Access Provider has to be installed on the managed node also. You can think of the TME 10 LAN Access Provider as the link between the NetFinity 5.0 TME 10 LAN Access Provider and the Tivoli TMR.

The TME 10 LAN Access Provider is not to be installed on the TMR server.

To install the TME 10 LAN Access Provider on romeo (our managed node acting as the gateway to the NetFinity 5.0 server), open up the folder where the TME 10 LAN Access code resides and run the Setup.exe.

The following window will appear:

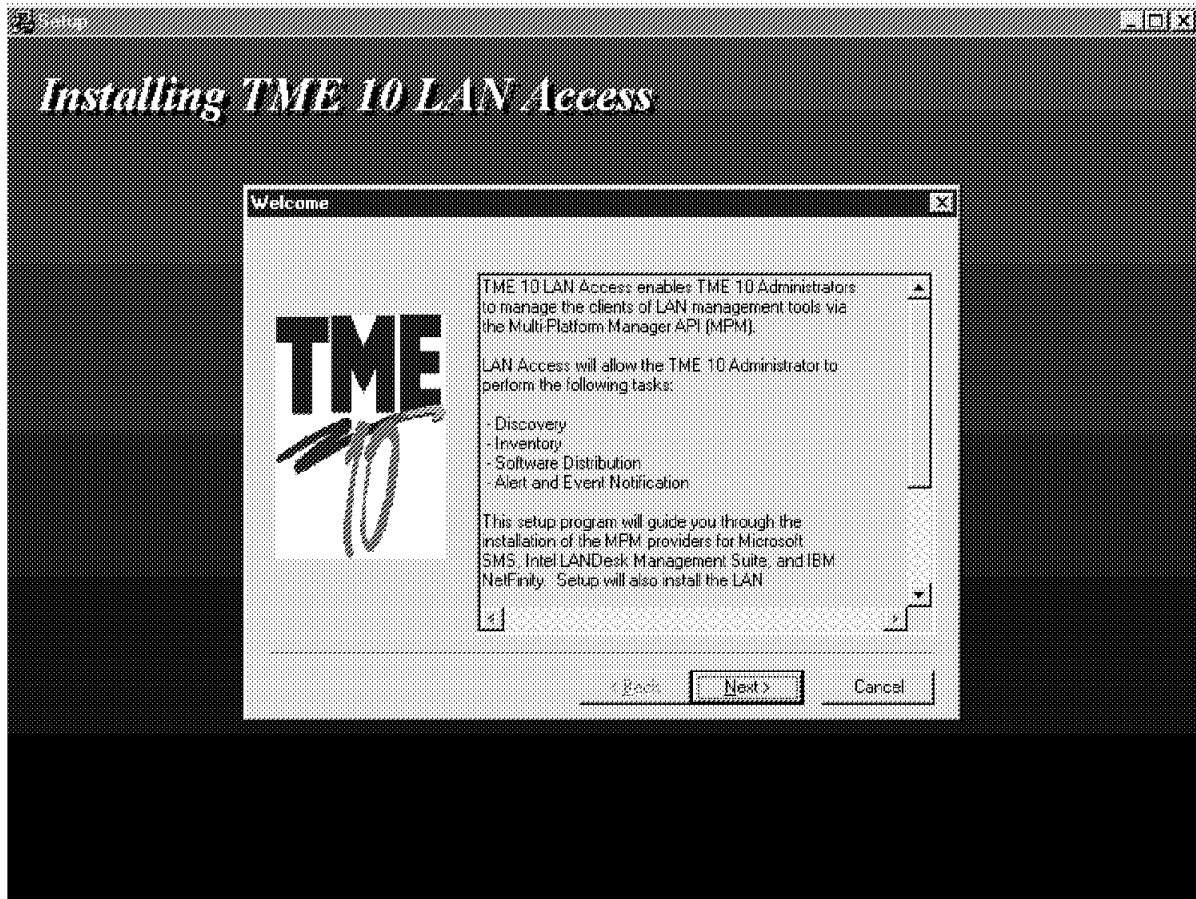


Figure 168. Setup Window

Click on **Next>** and then **Next>** again to accept the destination directory. Otherwise, change the destination accordingly.

The installation process will continue until the LAN Access Support Program - Authentication window appears:

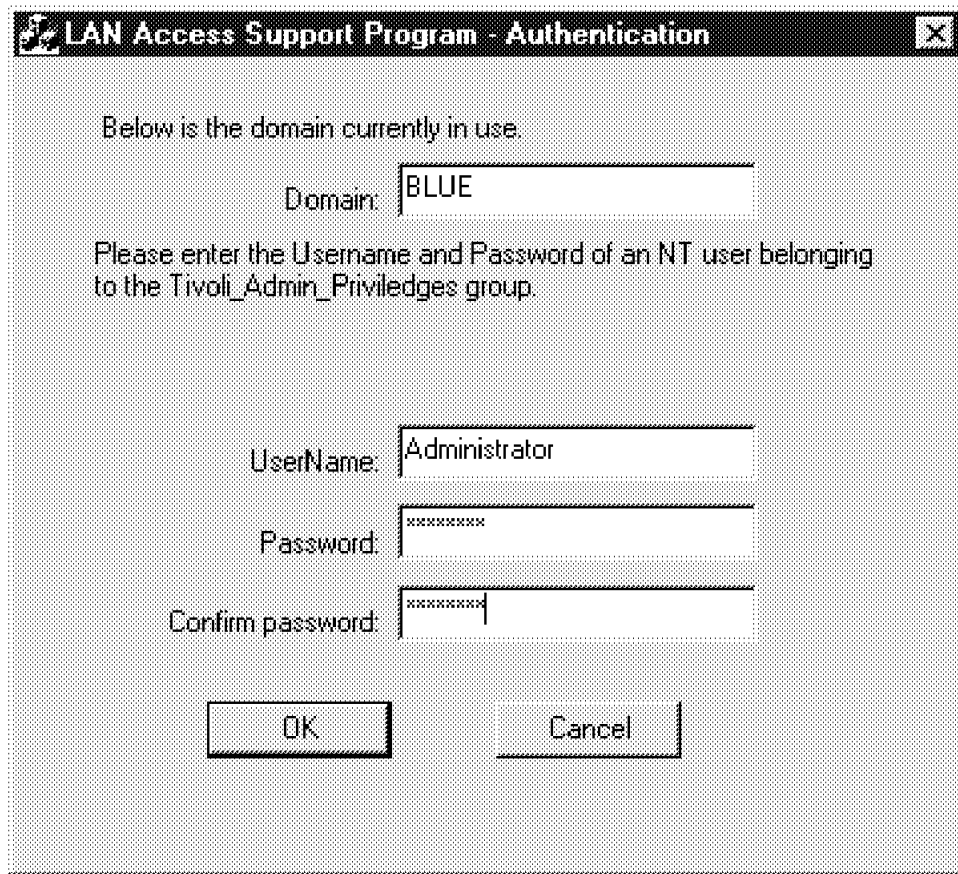


Figure 169. LAN Access Support Program - Authentication Window

Fill in the required information and click on **OK**.

- Domain: This is the name of the domain. In our scenario it was BLUE.
- UserName: Enter any user name that belongs to the Tivoli\_Admin\_Privileges group.
- Password: Make sure you spell the password correctly and repeat it to confirm your selected password.

The installation process will continue until the Network Driver Configuration window appears:

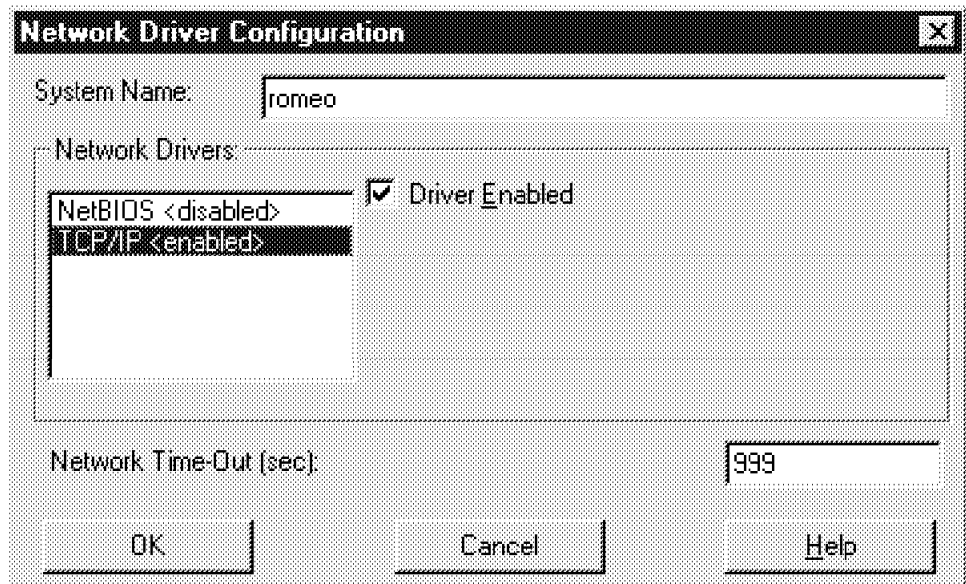


Figure 170. Network Driver Configuration Window

Now you have to define the network driver. The choices are NetBIOS or TCP/IP. In our scenario we enabled TCP/IP, since all our machines were configured for TCP/IP. Also for slow networks we recommend increasing the Network Time-Out value to the highest possible number. We selected 999 seconds.

When done, click on **OK**. In the next window (we do not show a picture of the window), click on **Finish** to restart the computer.

The machine will then be rebooted.

**Note**

Although it is possible to select both NetBIOS and TCP/IP you should use only one of them. If you enable both possible network drivers simultaneously, you will see some clients twice in your clients list. This would be for all clients that have been configured for both drivers. It probably would also cause some problems when two records for the same machine are generated for an insert/update in the same database.

### 7.4.3 Installing TME 10 LAN Access in the TMR

Make sure you have installed TME Framework Version 3.1 and Patch 3.1.1 as well as Patch 3.1.2 on the TMR and the managed node where you want to install TME 10 LAN Access.

Start your TME 10 Desktop and open the pull-down menu to install a new product. Set media according to where your installation code resides and start the installation.

When the following window appears, make sure to add both machines, the TMR server and the managed node. In our scenario, foxtrot is the TMR server and romeo is the NT managed node. Select **TME 10 LAN Access Version 1.1** and click on **Install & Close**.

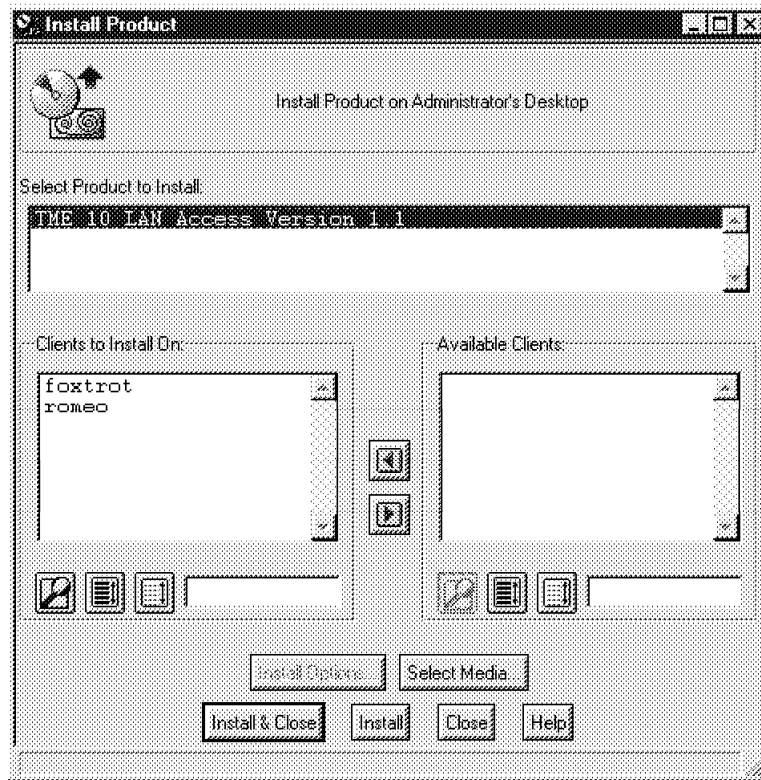


Figure 171. Install Product Window

A new window is opened. Click on **Continue Install**, when you are asked to do so.

When the installation is finished, the following window appears:



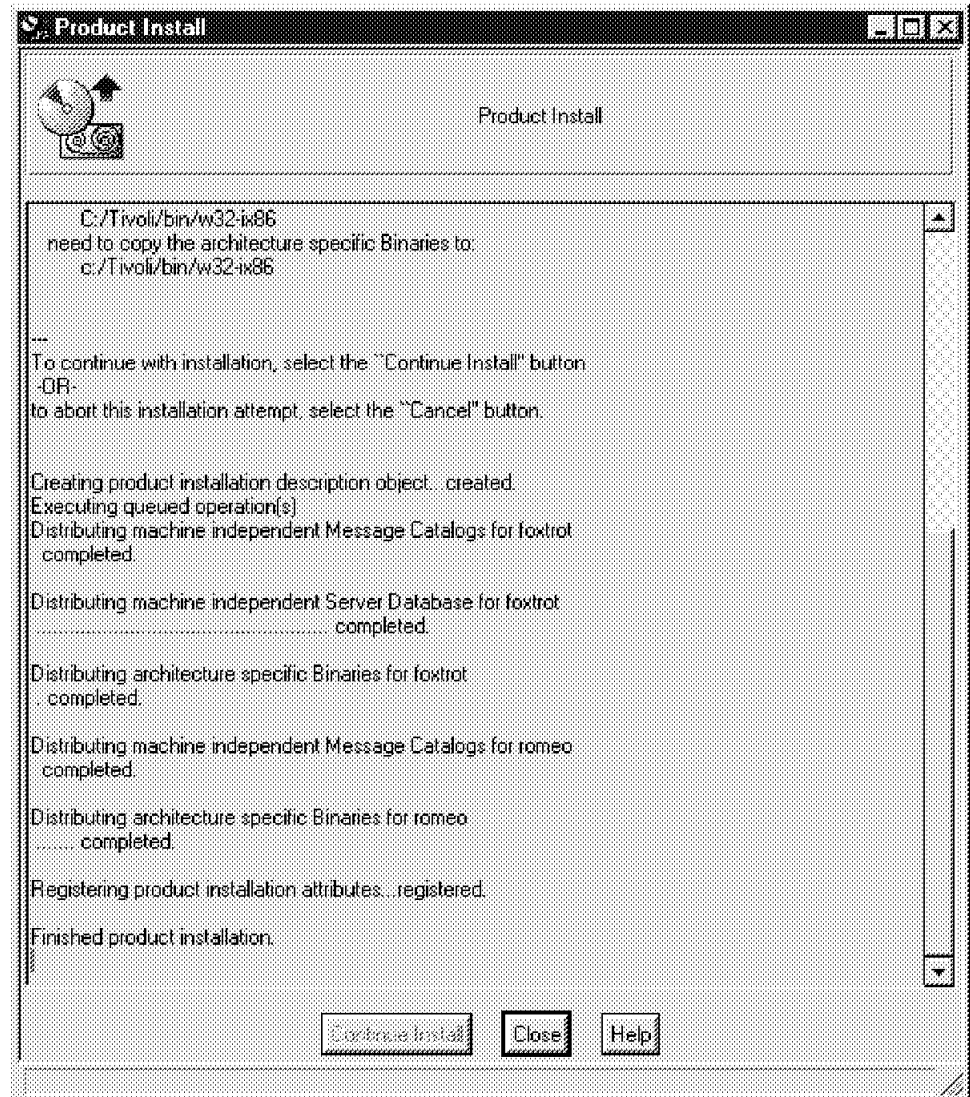


Figure 172. Product Install Window

Click on **Close** in this window.

The Installation of TME 10 LAN Access is now finished.

---

## 7.5 Creating and Configuring a New LAN Access Policy Region

To do some configuration in the TMR, do the following:

- Start the TME 10 Desktop and double-click on the **Administrators** icon. Select **Edit Resource Roles** from the pull-down menu and move all available roles to the current roles window side. Then click on **Change & Close**.
- Do the same for the Scheduler and all configured regions in your TMR. In our scenario, this is only the foxtrot-region.
- Open the foxtrot policy region and select **Properties** from the pull-down menu. Then select **Managed Resources** and move **LANAccessSite** from the Available Resources window to the Current Resources window and click on **Set & Close** as shown in the following figure.

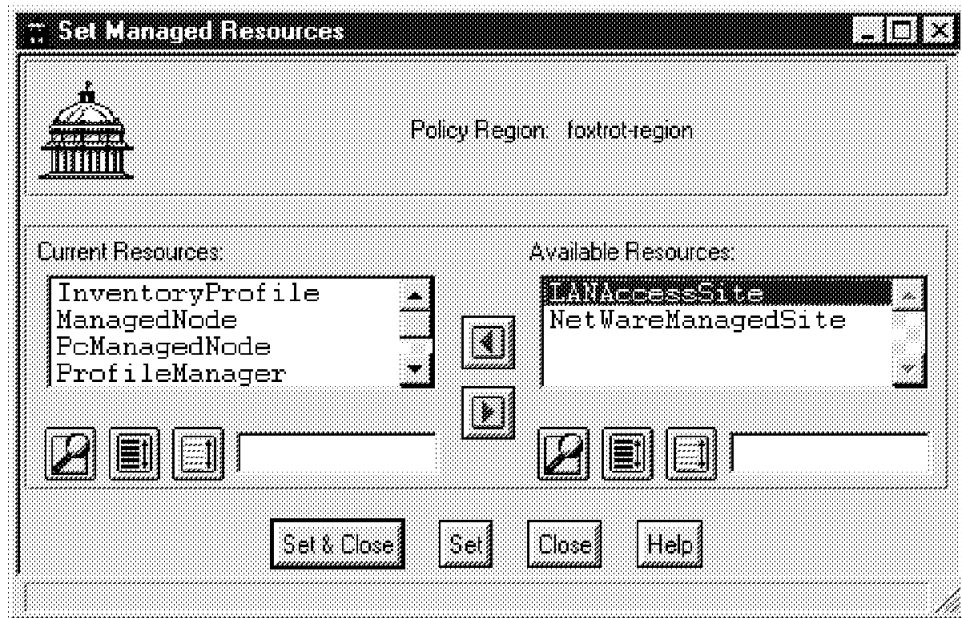


Figure 173. Set Managed Resources Window

Now you have to create a new policy region. Enter a name for the new policy region in the appropriate name field. We have chosen Lan Access Region as you can see in the following figure.



Figure 174. Create Policy Region Window

What you did before for Administrators, Scheduler and foxtrot-region repeat now for the Lan Access Region.

- Start the TME 10 Desktop and double-click on the icon for the Lan Access Region.

- Select **Edit Resource Roles** from the pull-down menu and move all Available Roles to the Current Roles side of the window.
- Click on **Change & Close**, which you can see in the following figure.

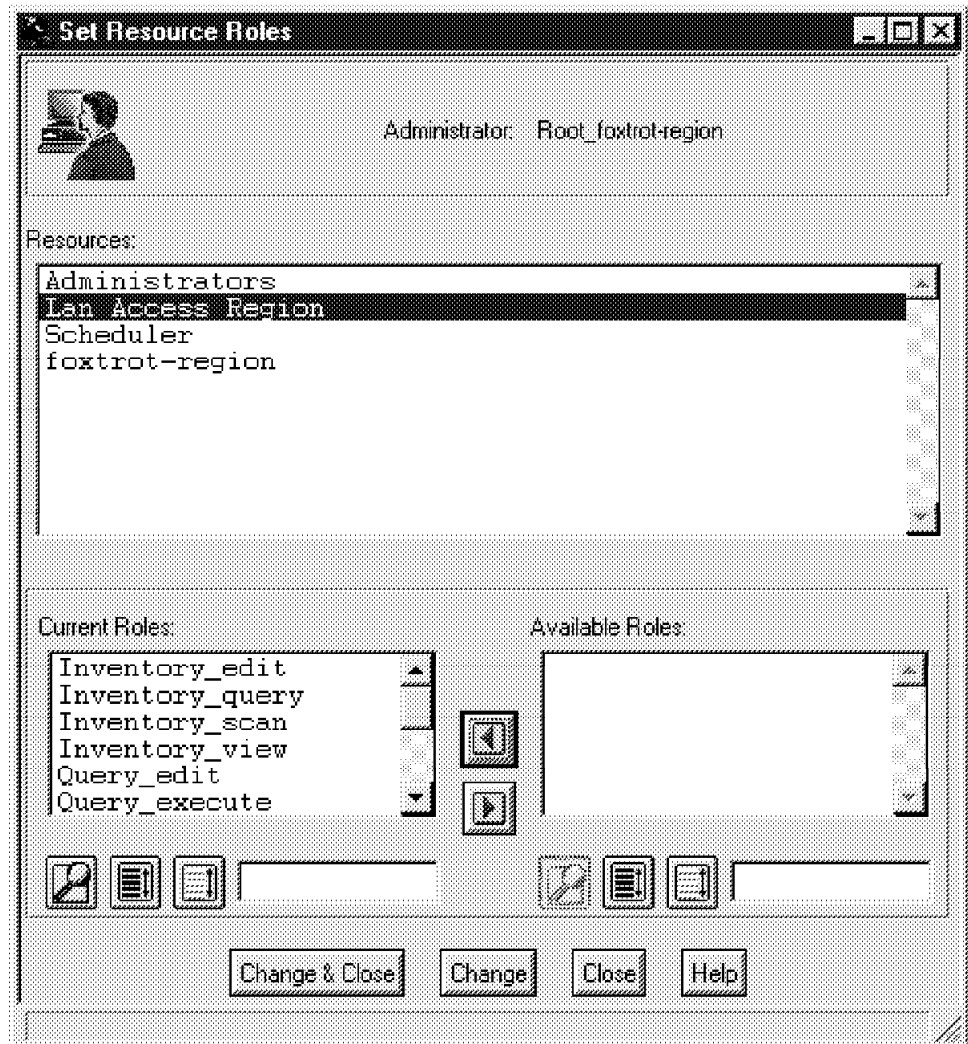


Figure 175. Set Resource Roles Window

The installation of all the TME 10 LAN Access components is now finished.

## 7.6 Performing an Inventory Scan

Now that we have finished the installation process we want to do a scan to collect the inventory data. That is, collect the data, that has been scanned by the TME 10 LAN Access Provider.

Do the following steps:

- Select **LANAccessSite...** from the policy region's **Create** pull-down menu to display the CreateLANAccess Site dialog.
- Fill out the fields for Log File Path and Staging Area Path as shown in Figure 176 on page 192.

- We found that an auto-detect for OS/2 machines does not work. In order to start the communication with our TME 10 LAN Access Provider we had to manually input the address in the address field.
- When you have done so, click on **Add** and then **Update & Close**.

**Note**

Please note that the host name in the address field is case sensitive.

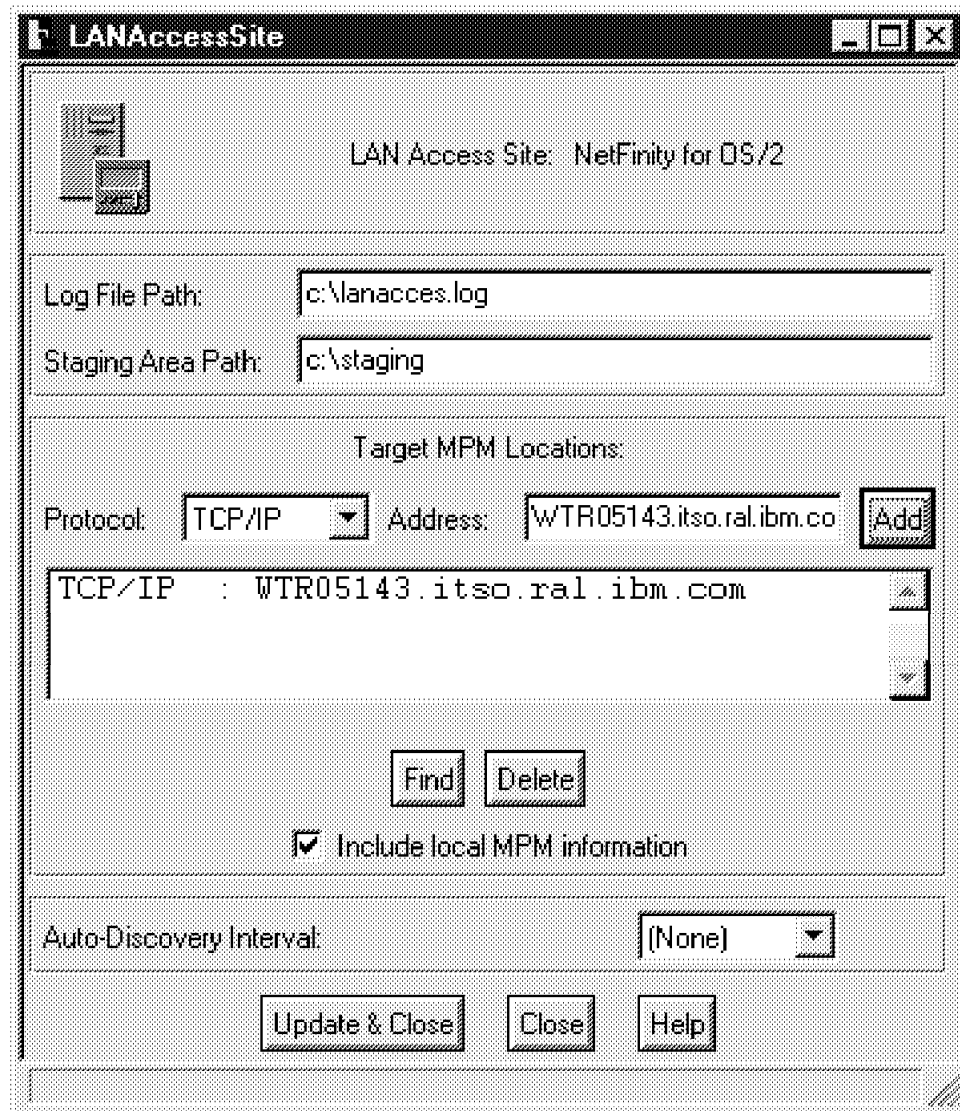


Figure 176. LANAccessSite Window

If you had clicked on the **Find** Button, you would have seen the Searching for MPM systems window instead, as shown in Figure 177 on page 193. We did not press the **Find** button, so you will see the figure as shown in Figure 178 on page 193.



Figure 177. LANAccessSite Window



Figure 178. Policy Region: Lan Access Region Window

Move the mouse pointer over the **NetFinity for OS/2** icon, and click on the right mouse button and then select **Create LAN Access Collection** from the pull-down menu.

The LANAccessCollection window will appear:

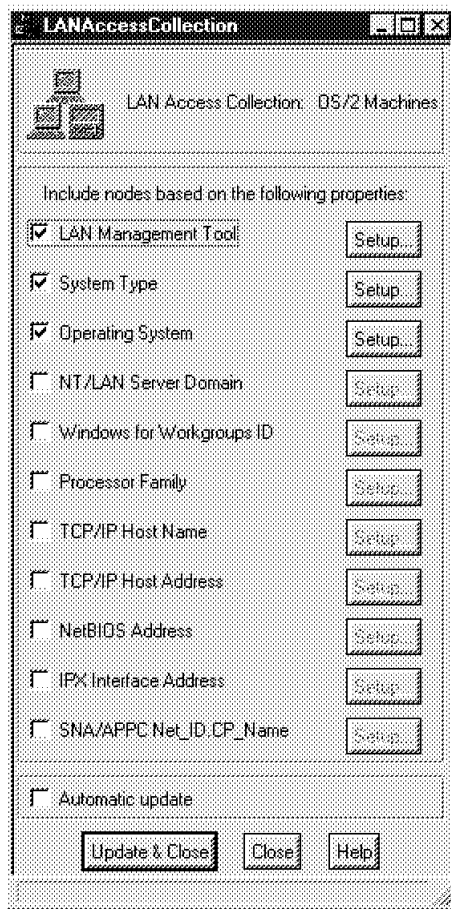


Figure 179. LANAccessCollection Window

In this window you can define your search and selection criteria.

When you have done the configuration according to your requirements click on **Update & Close**. Then your system will start a search and select the client systems that fit the selection and setup criteria you have defined in the previous window.

During that process the following window will appear:



Figure 180. LANAccessColl... Window

When you now double-click on the **NetFinity for OS/2** object (shown in Figure 178 on page 193) you will see all the machines that were selected.

See also the following figure which shows the small group of machines in our environment that had NetFinity 5.0 installed, and were thus detectable by TME 10 LAN Access.



Figure 181. OS/2 Machines Window

When you move the mouse pointer over the OS/2 machines you can open a pull-down menu that enables you to review the scanned software and hardware inventory data. As an example, we did it for KWIELA.

You can review the result in the following two figures. If you look for specific information, you may want to do a select on the database tables of your database and then select the appropriate information from your TME 10 Inventory database.

**Note**

It should be noticed again that the scanning process is not performed by TME 10 Inventory 3.1 itself. TME 10 Inventory 3.1 just uses the data that was scanned and generated previously by the TME 10 LAN Access Provider on the NetFinity 5.0 Server. How the inventory data, shown in Figure 182 on page 197 and Figure 183 on page 197, correlate with the data shown in Figure 164 on page 181 and Figure 165 on page 182, you can find out by manually comparing the data sources.



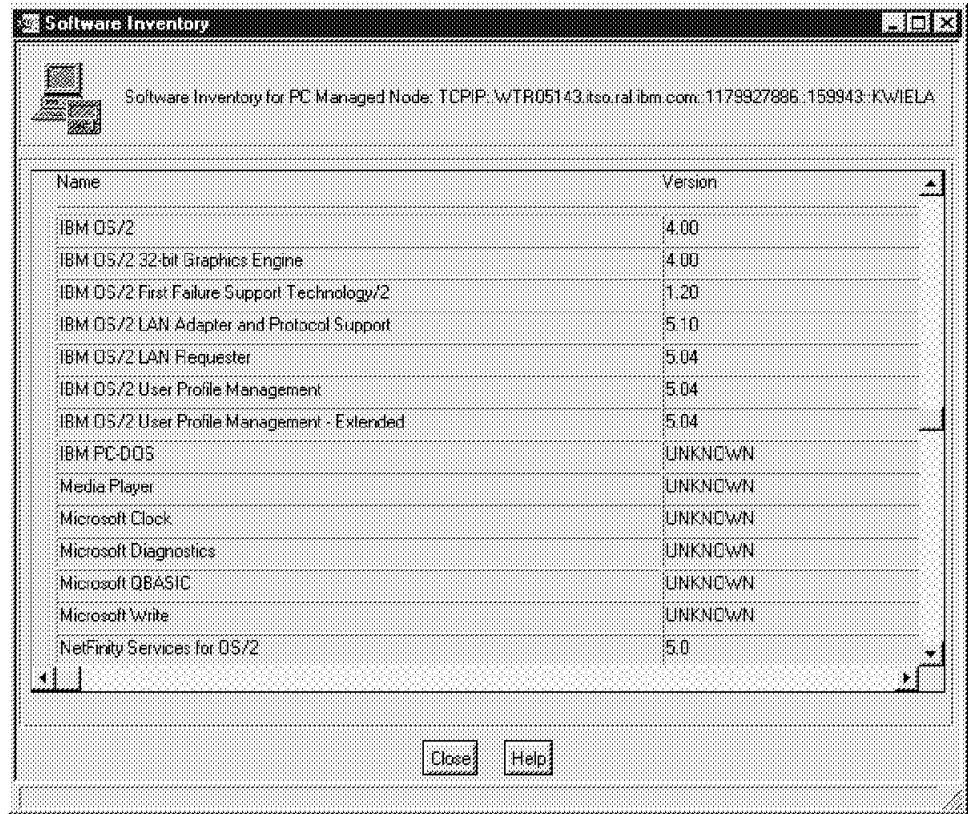


Figure 182. Software Inventory Window

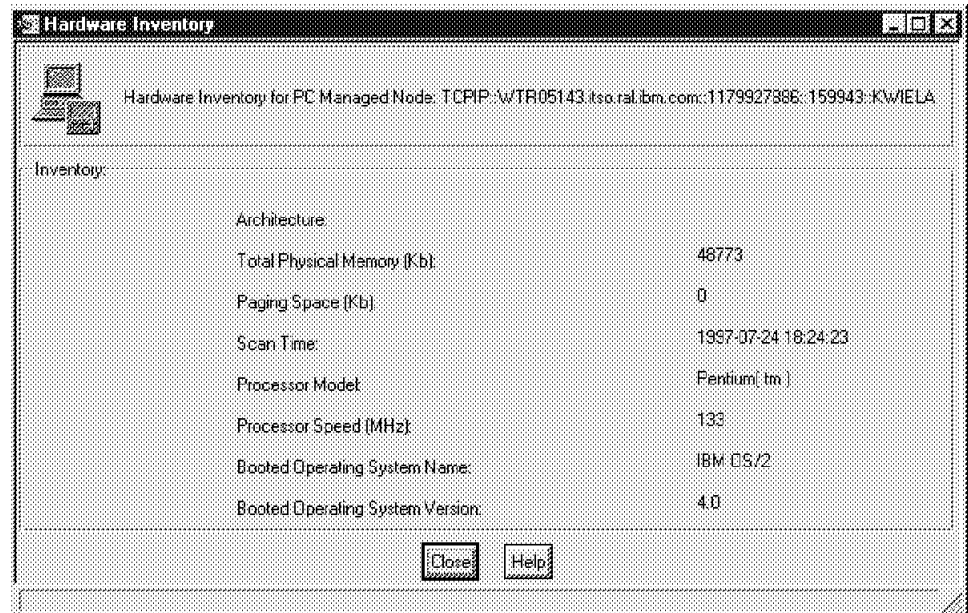


Figure 183. Hardware Inventory Window

## 7.6.1 Conclusion

As already explained in previous chapters, with TME 10 LAN Access you have been provided with an easy method for gathering inventory data from resources that are not part of a TMR. The data, however, integrated into the TME Inventory database, looks as if it was retrieved from within the TMR. This is extremely valuable, because, at this time, there is no other good way of scanning OS/2 machines.

---

## 7.7 Refreshing a TME 10 LAN Access Provider Scan

A new scan of the inventory data on a TME 10 LAN Access Provider is done automatically according to the refresh period set in the NetFinity Provider Configuration window. We showed this in Figure 161 on page 178.

There is no mechanism available yet to start a scan manually on a TME 10 LAN Access Provider.

However, there are two alternatives to force a new scan:

1. Reduce the refresh period to the lowest possible number, for example, every hour. The TME 10 LAN Access Provider will then start with a new scan within an hour.
2. Temporarily change the system time on the TME 10 LAN Access Provider, so the new time setting is ahead of the next scheduled refresh time period. This will start a new scan immediately after you make the change. Do not forget to change the time back to the original setting when the new scan is finished.

## Chapter 8. Using RDBMS Tools to Access TME 10 Inventory Data

In this chapter, we show how to query inventory data from any RDBMS client connected in the network. A RDBMS client is a workstation that has the database client software installed. We show what tools are available to query database information. With a few sample scripts, we also show how to generate various reports depending on a customer's requirements.

We use the following scenarios:

- Looking at an Oracle database from an NT client
- Looking at an Oracle database from an AIX client
- Looking at a Sybase database from an NT client
- Looking at a Sybase database from an AIX client

### 8.1 Overview of Our Environment

The following figure gives a brief overview of the RDBMS tools environment:

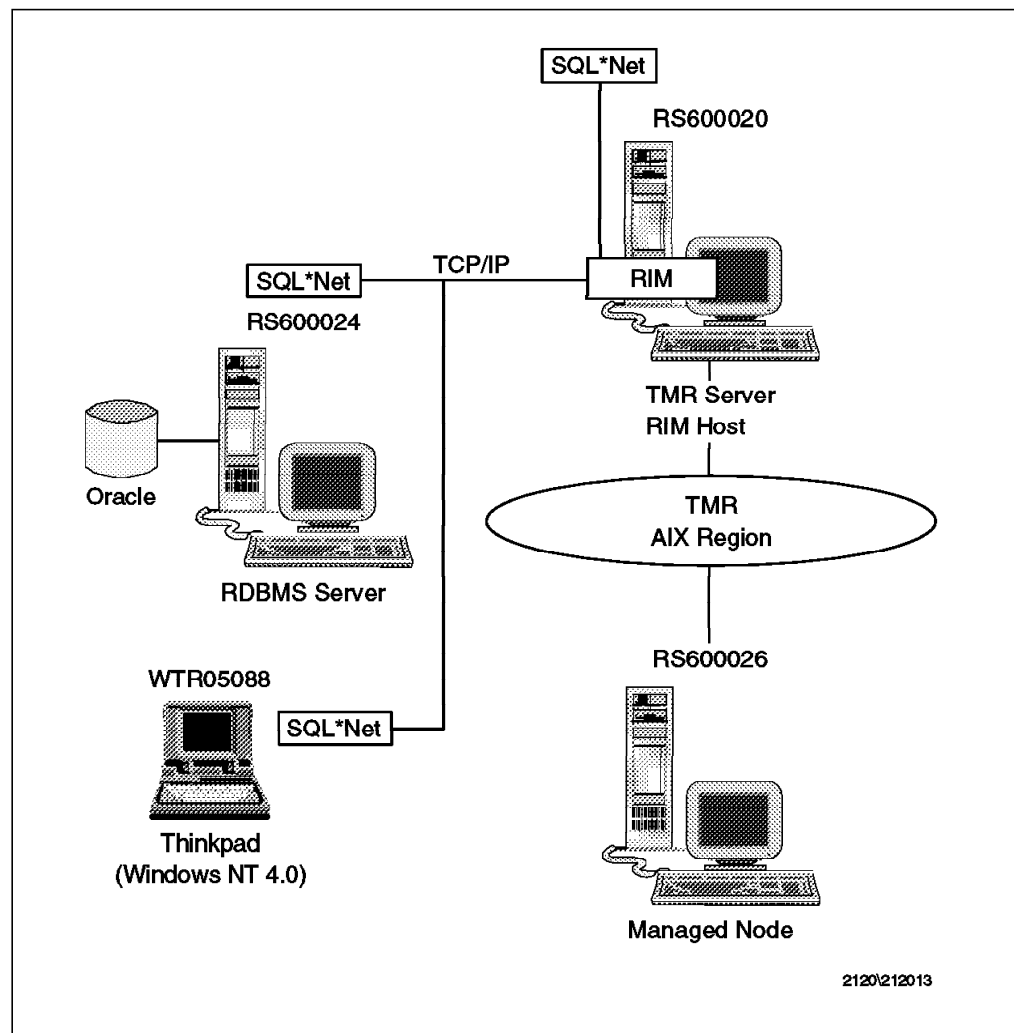


Figure 184. RDBMS Tools Environment

For a detailed description of the environment, we refer you to the following chapters:

- Chapter 4, “ Using TME 10 Inventory with Oracle for AIX” on page 63.
- Chapter 2, “Using TME 10 Inventory with Sybase on Windows NT” on page 13.

---

## 8.2 Working with an Oracle Database on an NT Client

In this environment the Oracle client is not part of any TMR. We want to prove, that in order to access inventory data collected in a Tivoli TMR, any workstation in the network can be used. You only must ensure the appropriate database software is installed on that client.

### 8.2.1 Installing Oracle for Windows NT 4.0 on an NT Client

Since the installation of Oracle on an NT client is not described in any chapter of this book, we will give a more detailed explanation of the installation process in this chapter.

We chose the ThinkPad WTR05088 to be our Oracle client. It has NT 4.0 already installed. Depending on which media your installation code resides on, you have to connect your client to the appropriate network resource or put a CD in your CD drive, if the installation code is available on a CD.

Run Setup.exe from the Oracle installation source code directory.

The Oracle Installer window will appear as shown in the following figure:

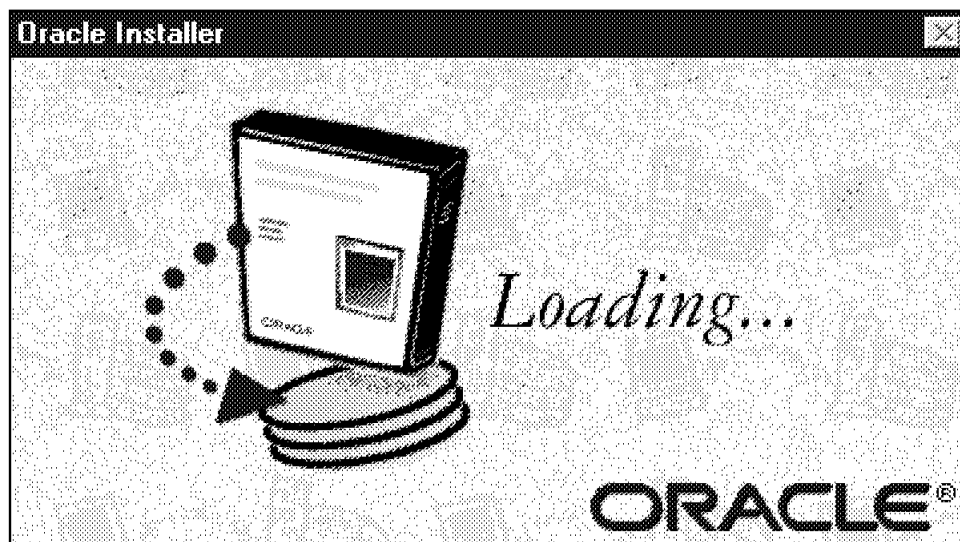


Figure 185. Oracle Installer Window

When the loading process is finished the following window will appear:

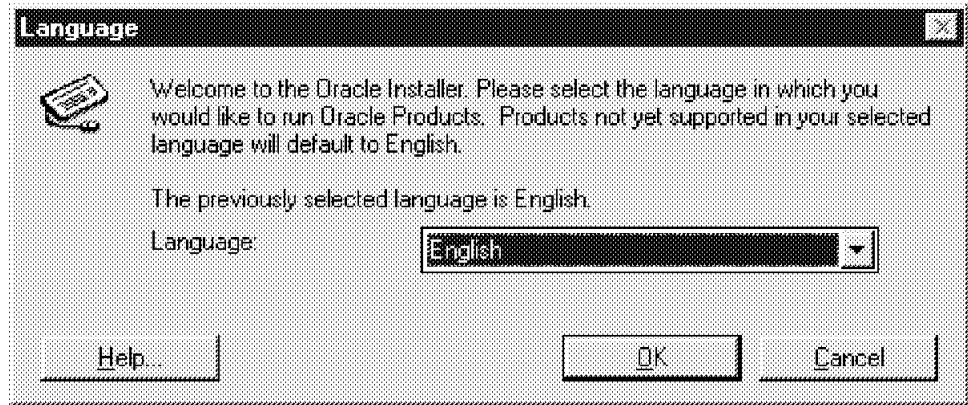


Figure 186. Language Window

Choose a different language from the language pull-down menu if you do not want English.

When you are finished, click on the **OK** button. The following window will appear:

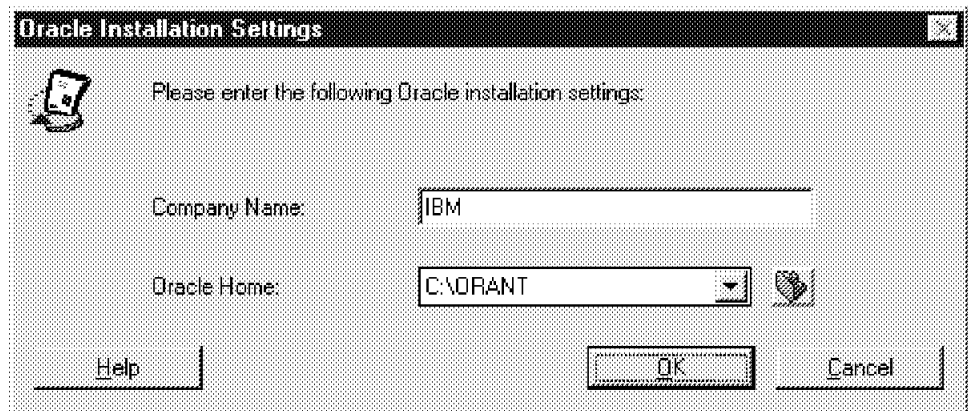


Figure 187. Oracle Installation Settings Window

Enter the name of your company in the Company Name field. If you are not satisfied with the default C:\ORANT enter the the drive and directory of your choice in the Oracle Home field. This is the place where the Oracle code will be installed on your client.

Click on the **OK** button when you are finished.

The following window will appear:

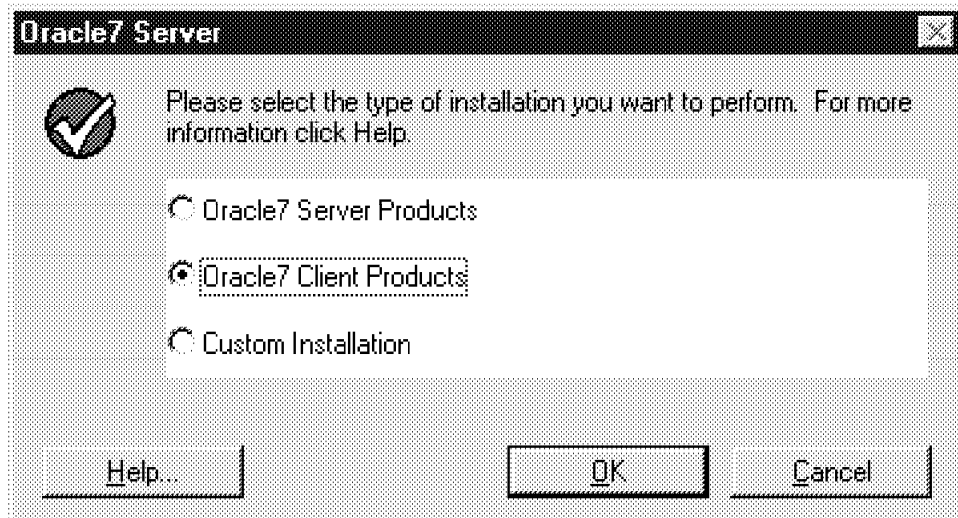


Figure 188. Oracle7 Server Window

We want to install the client version of Oracle, so we click on the radio button for **Oracle7 Client Products**.

Then click on **OK** and the following selection menu will appear:

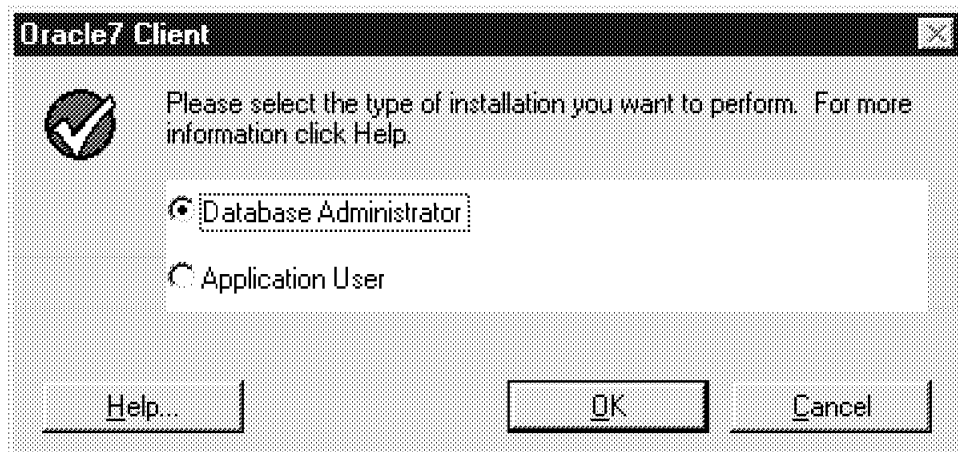


Figure 189. Oracle7 Client Window

If you want to do some database administration of the Oracle database from your NT client, you should select **Database Administrator**. A number of additional applications will be installed to do tasks as shown in the following list:

- Add, delete or modify users
- Backup, restore databases
- Import, export data to the database
- Any kind of monitoring

Otherwise click on the radio button **Application User**. Choosing **Application User** will completely satisfy your requirements to do queries on an Oracle inventory database.

When you have made your selection, click on **OK** to start the next window, as shown in the following figure:

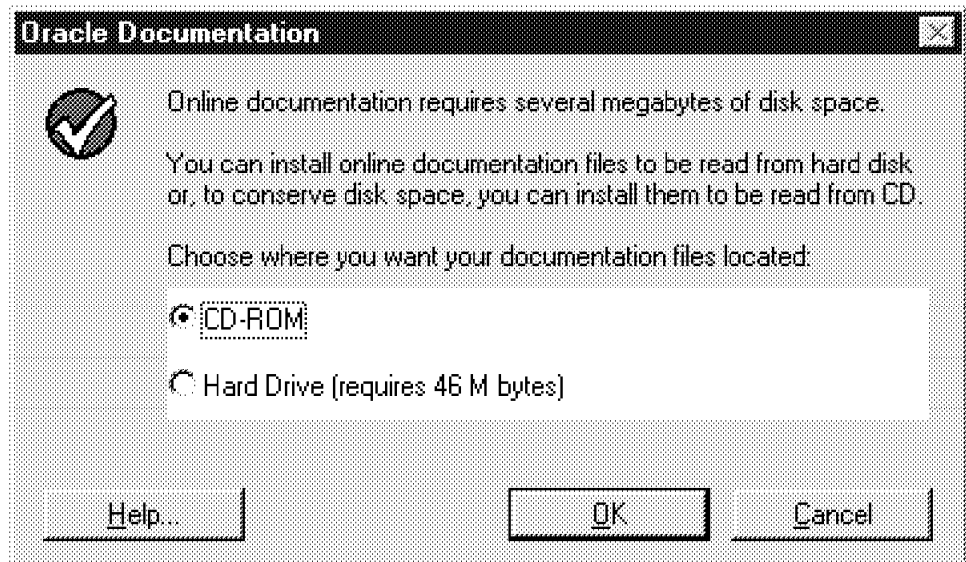


Figure 190. Oracle Documentation Window

Depending on how much space is left on your NT client, you may choose to install the online documentation files on your system or select **CD-ROM**. When you select **CD-ROM**, you will be requested to insert the Oracle CD in your CD-ROM drive, every time you ask for online help.

**Note**

We did not try out what happens when you select **CD-ROM** and your installation was started on a directory in the network.

When you have made your selection, click on **OK**.

This will start copying the Oracle code from your source directory to your client.

The installation progress is shown in the following figure:

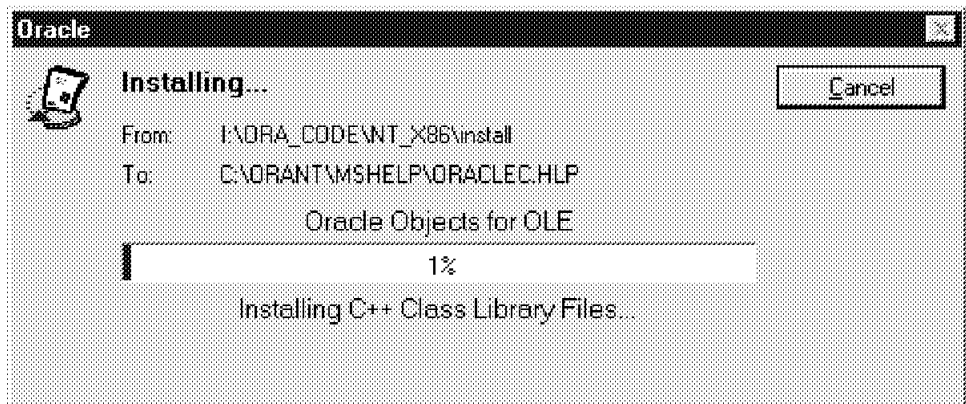


Figure 191. Oracle Window

When the installation process is finished, the following window will be shown on your screen:

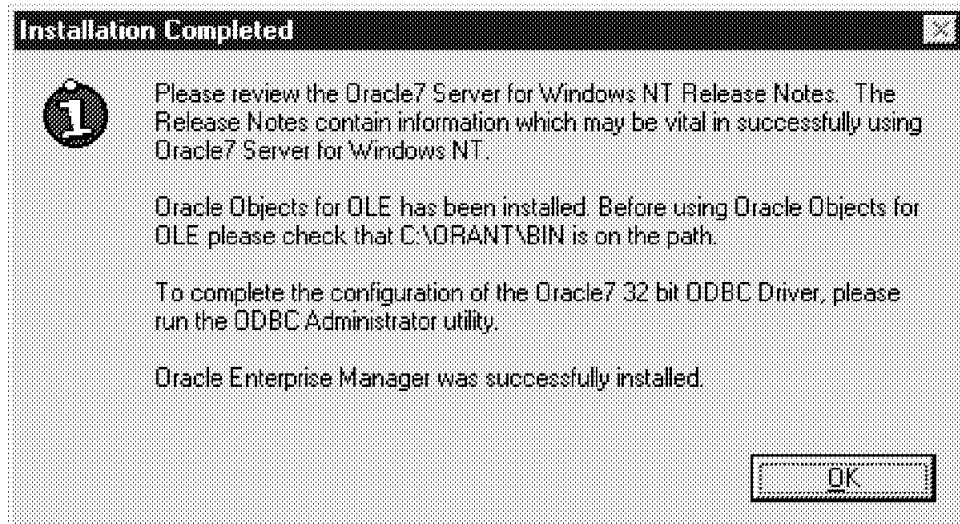


Figure 192. Installation Completed Window

Click on **OK** to complete the installation of the Oracle client on your NT client.

## 8.2.2 Configuring the Oracle Client for a Database Connection

You have two options to configure your Oracle client:

1. Get a copy of the configuration files TNSNAMES.ORA and SQLNET.ORA from another Oracle client that is connected to the same server you want to connect to or from the server itself. You find the two files in the ORACLE\_HOMENETWORKADMIN directory. Copy those two files into your ORACLE\_HOME\NETWORK\ADMIN directory.
2. Use the SQL\*Net Easy Configuration graphical user interface, to do the necessary configuration.

We choose option two in order to learn more about this process.

To start the program you also have two different options:

- Issue the following command on your Windows NT command line:  

```
ORAINST /INSTALL /PRD C:\ORANT\NETWORK\CFG
```

(Assuming your ORACLE\_HOME directory is C:\ORANT)
- On your NT desktop click on **Start** with the left mouse button. Then select **Programs** to explode the programs menu. Select **Oracle for Windows NT** to open the selection menu for a series of Oracle programs for Windows NT. Move the mouse pointer over the **SQL\*Net Easy Configuration** option and start this program with a click on the left mouse button.

Either option will start the same process.

You will see the following window:



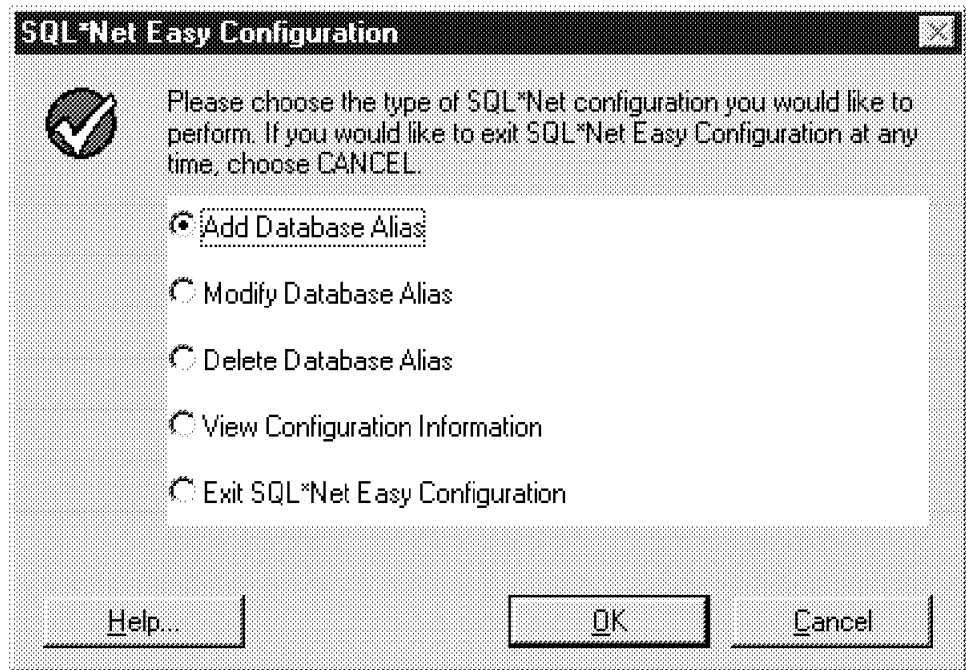


Figure 193. SQL\*Net Easy Configuration Window

On how the inventory database was installed and configured, we refer you to Chapter 4, “Using TME 10 Inventory with Oracle for AIX” on page 63. You want to add the configuration information to allow a connection to the Oracle inventory database, that was created in Chapter 4, “Using TME 10 Inventory with Oracle for AIX” on page 63. Select the radio button that is next to the Add Database Alias field.

Then click on **OK** to start the following window:

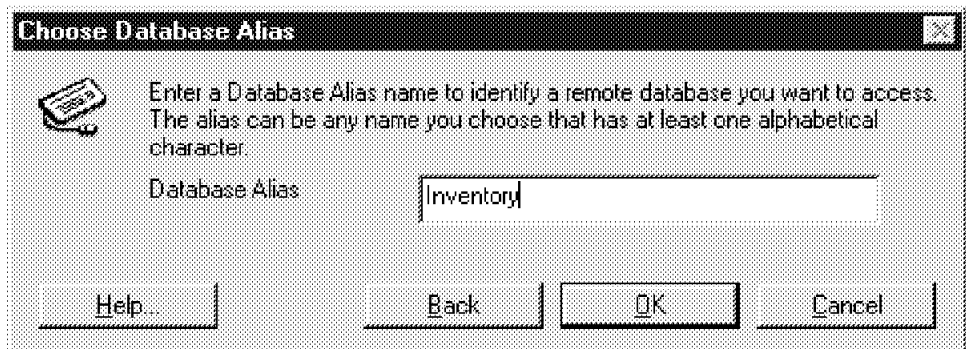


Figure 194. Choose Database Alias Window

In the Database Alias field you may enter any name that is meaningful to you. However, if you use more than one alias, be sure that the name is unique. We choose Inventory as the database alias name in our scenario. Later you may address your database by the instance name sid1 or by the alias Inventory, if you also choose Inventory as alias.

When you are finished click on **OK** to start the next window:

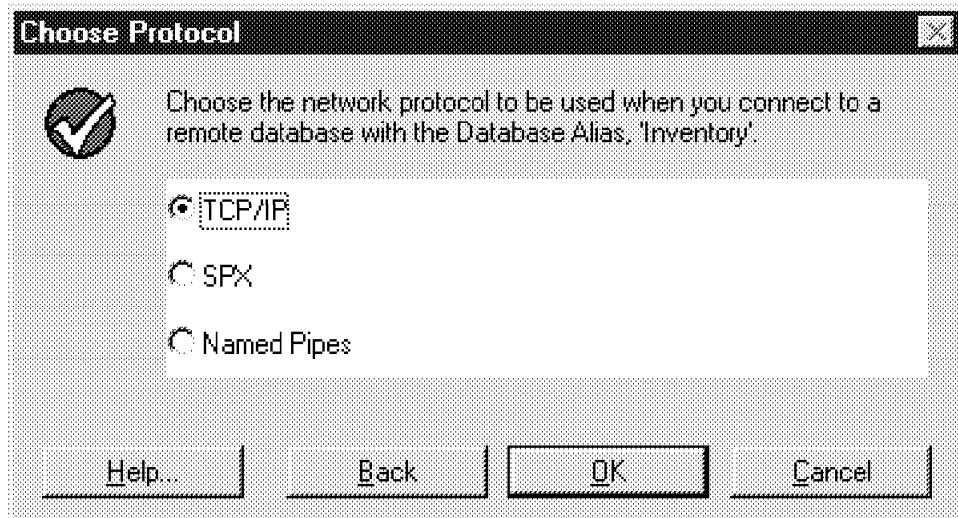


Figure 195. Choose Protocol Window

In the above window you are asked to choose your network protocol.

Make your selection and click on **OK** to start the following figure:

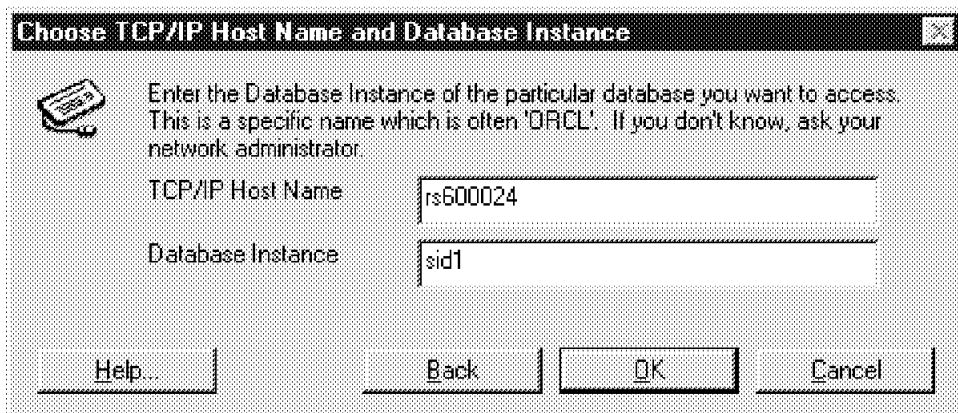


Figure 196. Choose TCP/IP Host Name and Database Instance Window

Enter the TCP/IP Host Name and the Database Instance in the appropriate data fields. TCP/IP Host Name is the machine where the Oracle database is installed on. Database Instance is the name of the Oracle database. If you do not know the information, you have to ask your database administrator.

You should know that we want to connect to a database created in a scenario that is explained in Chapter 4, "Using TME 10 Inventory with Oracle for AIX" on page 63. So, you have to get the information that was defined there.

When you have entered both of the required fields, click on **OK** to start the last window, which you can see in the following figure:

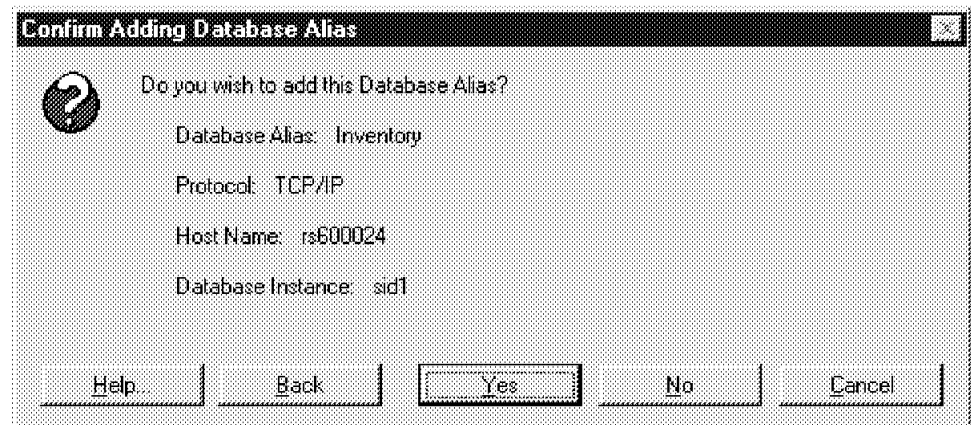


Figure 197. Confirm Adding Database Alias Window

You have now entered all the information that is needed. If you are not satisfied with the information shown in the above figure, you may step back by clicking on the **Back** button. You may then modify the information as needed.

When you are satisfied with the information click on **Yes** to complete the configuration process.

In the following figure, you can see a listing of sections of the TNSNAMES.ORA file, which we just generated.

```

#This is a SQL*Net Configuration file generated by
#SQL*Net Easy Configuration.
#Attention: Do not modify this file yourself.
#If you do, your SQL*Net Easy Configuration may not function properly.

Example1.world =
 (DESCRIPTION =
 (ADDRESS_LIST =
 (ADDRESS =
 .
 .
 .
)
)
)
Inventory.world =
 (DESCRIPTION =
 (ADDRESS_LIST =
 (ADDRESS =
 (COMMUNITY = tcp.world)
 (PROTOCOL = TCP)
 (Host = rs600024)
 (Port = 1521)
)
 (ADDRESS =
 (COMMUNITY = tcp.world)
 (PROTOCOL = TCP)
 (Host = rs600024)
 (Port = 1526)
)
)
 (CONNECT_DATA = (SID = sid1)
)
)
sid1.world =
 (DESCRIPTION =
 (ADDRESS_LIST =
 (ADDRESS =
 (COMMUNITY = tcp.world)
 (PROTOCOL = TCP)
 (Host = rs600024)
 (Port = 1521)
)
 (ADDRESS =
 (COMMUNITY = tcp.world)
 (PROTOCOL = TCP)
 (Host = rs600024)
 (Port = 1526)
)
)
 (CONNECT_DATA = (SID = sid1)
)
)
)

```

Figure 198. Content of the TNSNAMES.ora File

### 8.2.3 Testing the Connection to the Oracle Database

Before you continue, you should test your Oracle client installation. You can find out if the configuration you have done is correct by using a program that is provided with the Oracle installation.

On your NT desktop click on **Start** with the left mouse button. Then select **Programs** to explode the programs menu. Select **Oracle for Windows NT** to open the selection menu for a series of Oracle programs for windows NT. Move the mouse pointer over the **32-bit ODBC TEST** option and start this program with a click on the left mouse button.

On the window that is opened, click on **Connect** which will start another window. From this window, select **SQL Data Sources**. Depending on the alias you have chosen in Figure 194 on page 205, a list of the available SQL Data Sources is shown. Select **Inventory** and click on **OK**.

An additional window is started as shown in the following figure:

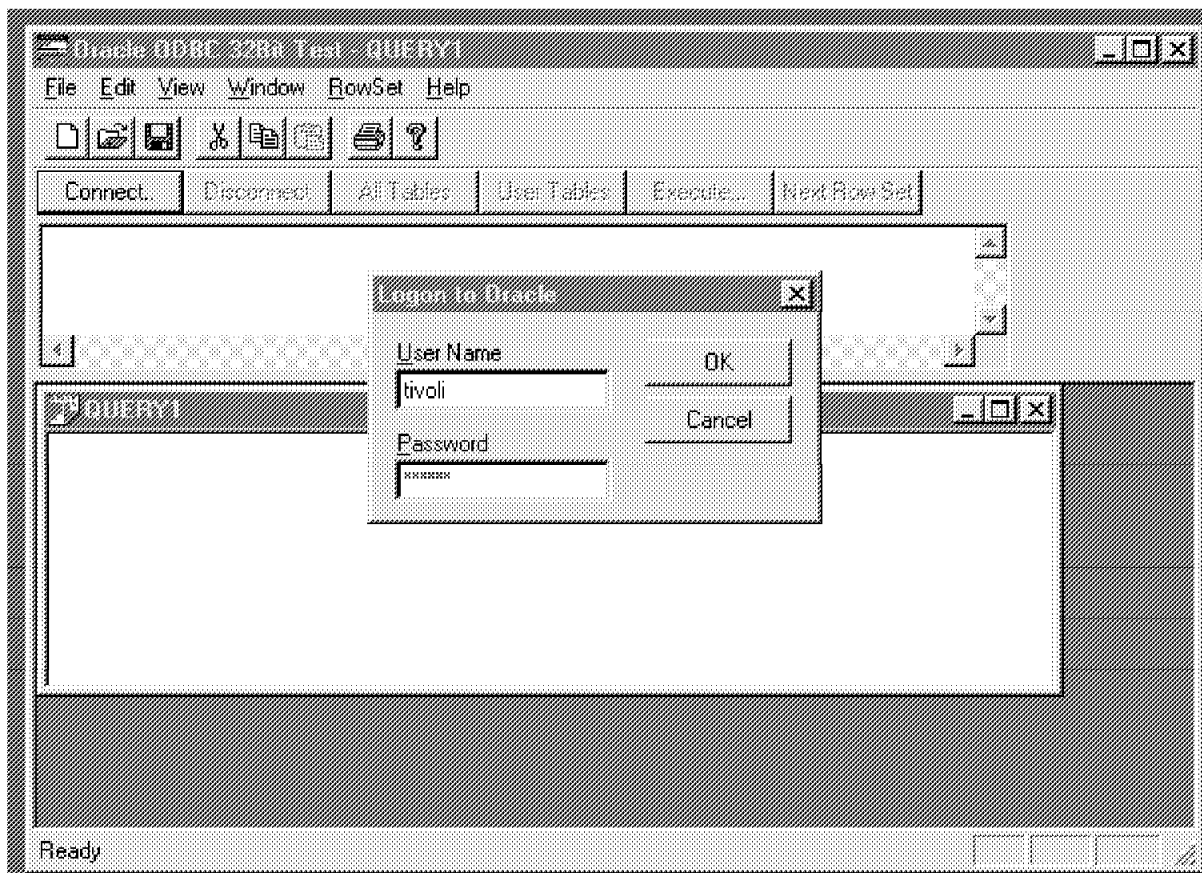


Figure 199. Login to Oracle Window

Enter the User Name and Password, that you have been given to access the Oracle database. In our scenario, both User Name and Password were chosen as tivoli, as it was defined in Chapter 4, "Using TME 10 Inventory with Oracle for AIX" on page 63.

If you click on **OK** and there are no error messages, you have a connection to the Oracle database.

Do one more test and enter the command `select * from inventorydata` as shown in Figure 200 on page 210.

After you have clicked on **Execute**, you will see the result of your query in the lower part of the following figure:

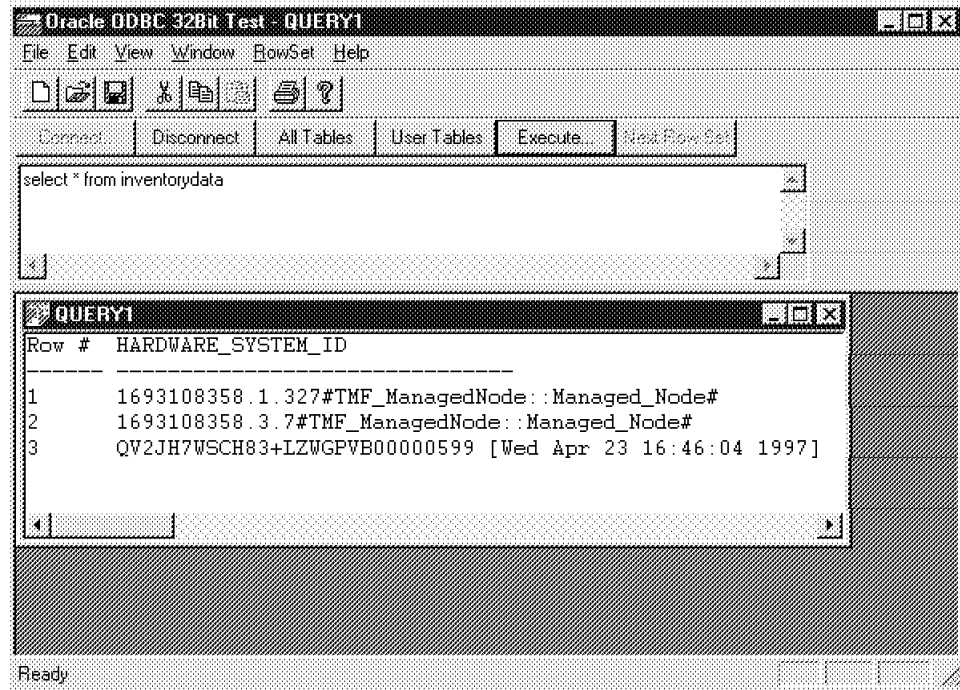


Figure 200. Oracle ODBC 32Bit Test - Query1 Window

If your Oracle installation is correct, you can successfully connect to your Oracle database.

## 8.2.4 Different Ways to Look at Oracle Inventory Data

Depending on what you prefer most, working on a command prompt or using a GUI, you may use the tools that are available as GUIs or use SQL\*PLUS as the command line interface. We show both alternatives and how to use them.

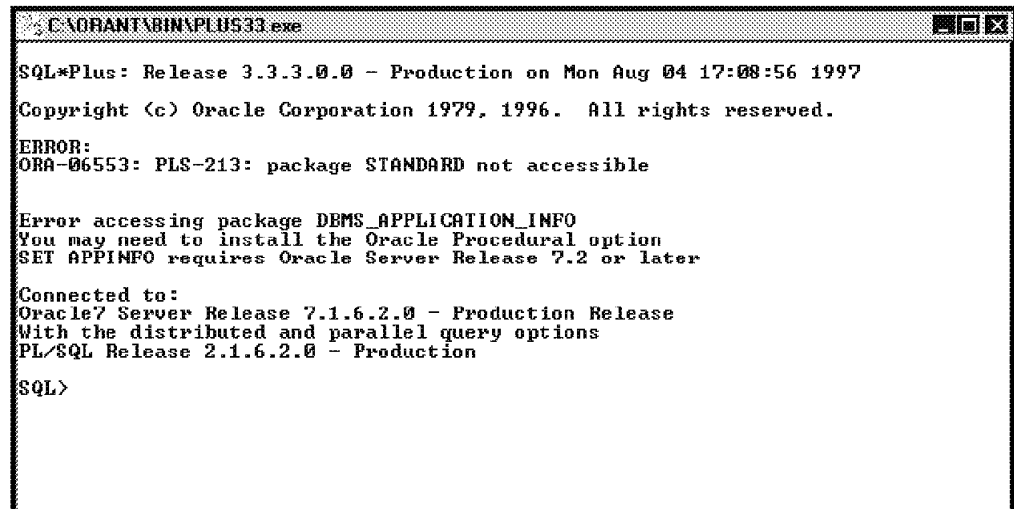
### 8.2.4.1 Using SQL\*PLUS

There are two alternatives to start SQLPLUS:

1. Enter SQLPLUS and the system will prompt you for your user ID and password.
2. Enter SQLPLUS and add a parameter that includes your user ID, password, and database name:

```
SQLPLUS TIVOLI/TIVOLI@Inventory
```

Both alternatives start a new session and you will see the following window:



```
C:\ORANT\BIN\PLUS33.exe
SQL*Plus: Release 3.3.3.0.0 - Production on Mon Aug 04 17:08:56 1997
Copyright (c) Oracle Corporation 1979, 1996. All rights reserved.
ERROR:
ORA-06553: PLS-213: package STANDARD not accessible

Error accessing package DEMS_APPLICATION_INFO
You may need to install the Oracle Procedural option
SET APPINFO requires Oracle Server Release 7.2 or later

Connected to:
Oracle7 Server Release 7.1.6.2.0 - Production Release
With the distributed and parallel query options
PL/SQL Release 2.1.6.2.0 - Production

SQL>
```

Figure 201. C:ORANTPSPLUS33.exe Window

**Note**

You can ignore the error message you see in the window. We could not find a cause for the message, but everything else seemed to worked fine.

When you see a **SQL>** prompt on your SQL\*PLUS window, you are connected to your inventory database.

From within this window you may execute any SQL command you wish.

As a first try, let's say you want to know all tables that exist in your inventory database.

Enter the following command on your SQL command prompt:

```
select * from all_catalog where owner='TIVOLI';
```

**Note**

Do not forget the semicolon at the end of your command. If you do, you will be prompted for a second line of your command. A command must be terminated with a semicolon before the command can be executed.

When you press Enter you will see the answer as shown in the following figure:

```

C:\ORANT\BIN\PLUS33.exe
TIUOLI SMTABLESPAGE_S TABLE
TIUOLI SMTemporarySEQUENCE SEQUENCE
TIUOLI SMUCENDPOINTSEQUENCE SEQUENCE
TIUOLI SMUCENDPOINT_S TABLE
TIUOLI SMWEEKLYENTRYSEQUENCE SEQUENCE
TIUOLI SMWEEKLYENTRY_S TABLE
TIUOLI SOFTWARE TABLE
TIUOLI SOFTWARE_COMPONENT TABLE
TIUOLI SOFTWARE_COMPONENT_FILE TABLE
TIUOLI SOFTWARE_COMPONENT_INSTANCE TABLE
TIUOLI SOFTWARE_COMPONENT_TASK TABLE
TIUOLI SOFTWARE_DISTRIBUTION_VIEW VIEW
TIUOLI SOFTWARE_FILEPACK TABLE
TIUOLI SOFTWARE_SIGNATURE_FILE TABLE
TIUOLI SOFTWARE_VERSION TABLE
TIUOLI SOUND_CARD TABLE
TIUOLI SW_COMPONENT_DEPENDENCY TABLE
TIUOLI SW_COMPONENT_RELATIONSHIP TABLE
TIUOLI TAPE_DRIVE TABLE
TIUOLI UNIX_SERVER TABLE
TIUOLI VIDEO_CARD TABLE

231 rows selected.

SQL>

```

Figure 202. C:\ORANT\PSPLUS33.exe Window

If you want the result also to be written to a spool file, enter the following command before you execute your SQL command:

```
spool C:\test\test.log
```

This will open a spool file. Any output that you see on your screen will also be written to this spool file. When you want to stop appending to the spool file, issue the following command:

```
spool off
```

You can see an example of the list output in the following figure:



```
SQL> select * from all_catalog where owner='TIVOLI';
```

| OWNER  | TABLE_NAME                | TABLE_TYPE |
|--------|---------------------------|------------|
| TIVOLI | ASSET                     | TABLE      |
| TIVOLI | BASENODE_VIEW             | VIEW       |
| TIVOLI | CDROM_DRIVE               | TABLE      |
| TIVOLI | COMMERCIAL_APPLICATION    | TABLE      |
| TIVOLI | COMPONENT_MONITOR         | TABLE      |
| TIVOLI | COMPONENT_MONITOR_PROFILE | TABLE      |
| TIVOLI | COMPUTER_SYSTEM           | TABLE      |
| TIVOLI | COMPUTER_SYSTEM_MEMORY    | TABLE      |
| TIVOLI | COMPUTER_VIEW             | VIEW       |
| TIVOLI | CONFIG_CHANGE_HISTORY     | TABLE      |
| TIVOLI | CONTROLLER_CARD           | TABLE      |
| TIVOLI | DESKTOP_PC                | TABLE      |
| .      |                           |            |
| .      |                           |            |
| .      |                           |            |
| TIVOLI | SMP_JOB_ID_               | SEQUENCE   |
| TIVOLI | SMP_JOB_INSTANCE          | VIEW       |
| TIVOLI | SMP_JOB_INSTANCE_         | TABLE      |
| TIVOLI | SMP_JOB_LIBRARY           | VIEW       |
| TIVOLI | SMP_JOB_LIBRARY_          | TABLE      |
| TIVOLI | SMP_JOB_TASK_INSTANCE     | VIEW       |
| TIVOLI | SMP_JOB_TASK_INSTANCE_    | TABLE      |
| TIVOLI | SMP_LONG_ID               | SEQUENCE   |
| TIVOLI | SMP_LONG_TEXT             | VIEW       |
| TIVOLI | SMP_LONG_TEXT_            | TABLE      |
| TIVOLI | SMP_REP_VERSION           | TABLE      |
| TIVOLI | SMP_SERVICES              | TABLE      |
| TIVOLI | SMP_SERVICE_GROUP_DEFN    | VIEW       |
| TIVOLI | SMP_SERVICE_GROUP_DEFN_   | TABLE      |
| TIVOLI | SMP_SERVICE_GROUP_ITEM    | VIEW       |
| TIVOLI | SMP_SERVICE_GROUP_ITEM_   | TABLE      |
| TIVOLI | SMP_SERVICE_ITEM          | VIEW       |
| TIVOLI |                           |            |

Figure 203. Sample SQL Spool File (Test.log)

When you need detailed information about the structure of a certain table, enter the following command:

```
describe inventorydata;
```

The result is a description of the different columns in the inventorydata table. You can see the result in the following figure:

```

C:\ORANT\BIN\PLUS33.exe
Error accessing package DBMS_APPLICATION_INFO
You may need to install the Oracle Procedural option
SET APPINFO requires Oracle Server Release 7.2 or later

Connected to:
Oracle7 Server Release 7.1.6.2.0 - Production Release
With the distributed and parallel query options
PL/SQL Release 2.1.6.2.0 - Production

SQL> describe inventorydata
Name Null? Type

HARDWARE_SYSTEM_ID NOT NULL VARCHAR2(64)
TIME_OBJECT_ID VARCHAR2(128)
TIME_OBJECT_LABEL VARCHAR2(32)
COMPUTER_ARCHITECTURE VARCHAR2(32)
PHYSICAL_MEMORY_KB NUMBER(38)
PAGING_SPACE_KB NUMBER(38)
COMPUTER_SCANTIME VARCHAR2(32)
PROCESSOR_MODEL VARCHAR2(32)
PROCESSOR_SPEED NUMBER(38)
BOOTED_OS_NAME VARCHAR2(16)
BOOTED_OS_VERSION VARCHAR2(16)

SQL>

```

Figure 204. C:\ORANT\PSPLUS33.exe Window

### 8.2.4.2 Writing SQL Scripts to Generate Reports

Now that we have done some basics, we want to write an SQL script that does the following:

- Select every machine from the INVENTORY and LOGICALDRIVE tables.
- Show the name of the logical drive.
- Show the total size of any logical drive.
- Show free capacity of any logical drive.
- Calculate the percentage of the used portion of every logical drive.
- Sum up the total capacity and free capacity for every machine.
- Write a report and spool it to C:\testtest1.log.
- Close the spool file when finished.

The following TEST1.SQL script will collect the information requested and generate a report according to the given requirements:

```
1 title 'Report Example'

2 set headsep !

3 column tme_object_label heading 'Machine' format a10 truncated
4 column logicaldrive_mountdir heading 'Drive' format a10 truncated
5 column logicaldrive_size_kb heading 'Total!Size! KB ' format 999999990
6 column logicaldrive_free_kb heading 'Free!Size! KB ' format 999999990
7 column Perc heading 'used! % ' format 99999

8 break on tme_object_label skip 2

9 compute sum of logicaldrive_size_kb logicaldrive_free_kb
 on tme_object_label

10 set linesize 79
11 set pagesize 30
12 set newpage 0

13 spool test1.lst

14 select tme_object_label,
15 logicaldrive_mountdir,
16 logicaldrive_size_kb,
17 logicaldrive_free_kb,
18 round((logicaldrive_size_kb - logicaldrive_free_kb)
 /logicaldrive_size_kb,2)*100 "Perc"
19 from inventorydata, logicaldrive
20 where inventorydata.hardware_system_id
21 = logicaldrive.hardware_system_id
22 and logicaldrive_free_kb > 0;

23 spool off

24 quit
```

Figure 205. Sample SQL Script File (Test1.sql)

**Note**

The line numbers in the script are not part of the code. They are used only as an aid to refer to a certain line.

To run the SQL script, you have to execute the following command:

```
SQLPLUS tivoli/tivoli@Inventory @c:\test\test1.sql
```

**Note**

Please note, that you do not have to start your SQLPLUS window first. You can do it all with one command:

- Start the SQLPLUS session.
- Execute the TEST1.SQL script.
- Close the SQLPLUS session when finished.

The following is the sample report that was generated:

```

Mon Aug 04
 Report Example
 Total Free
 Size Size used
Machine Drive KB KB %

rs600020 / 16384 1248 92
 /home 16384 10136 38
 /usr 1409024 781392 45
 /var 131072 14904 89
 /tmp 32768 15032 54
 /oracle_li 114688 32344 72

sum 1720320 855056
rs600026 / 16384 6928 58
 /home 8192 6392 22
 /usr 524288 35360 93
 /var 65536 13952 79
 /tmp 24576 21816 11
 /oracle_li 106496 24144 77

sum 745472 108592

```

Figure 206. Sample Output List (test1.lst)

In another example, you may want to get a report for all logical drives of every machine that show more than 80% utilization.

You have to modify the script shown in Figure 205 on page 215; remove line 9 and add line 20 as shown in the following TEST2.SQL script:

```

1 ttitle 'Report Example'

2 set headsep !

3 column tme_object_label heading 'Machine' format a10 truncated
4 column logicaldrive_mountdir heading 'Drive' format a10 truncated
5 column logicaldrive_size_kb heading 'Total!Size! KB ' format 999999990
6 column logicaldrive_free_kb heading 'Free!Size! KB ' format 999999990
7 column Perc heading 'used! % ' format 99999

8 break on tme_object_label skip 2

9 set linesize 79
10 set pagesize 30
11 set newpage 0

12 spool test2.lst

13 select tme_object_label,
14 logicaldrive_mountdir,
15 logicaldrive_size_kb,
16 logicaldrive_free_kb,
17 round((logicaldrive_size_kb - logicaldrive_free_kb)
18 /logicaldrive_size_kb,2)*100 "Perc"
19 from inventorydata, logicaldrive
20 where inventorydata.hardware_system_id
21 = logicaldrive.hardware_system_id
22 and (logicaldrive_size_kb - logicaldrive_free_kb)/logicaldrive_size_kb
23 >.8
24 and logicaldrive_free_kb > 0;

25 spool off

26 quit

```

Figure 207. Sample SQL Script File (Test2.sql)

When you execute the following command, you will get a report as shown in Figure 208.

SQLPLUS tivoli/tivoli@Inventory @c:\test\test2.sql

```

Mon Aug 04
 Report Example
 Total Free
 Size Size used
 KB KB %

rs600020 / 16384 1248 92
 /var 131072 14904 89

rs600026 /usr 524288 35360 93

```

Figure 208. Sample Output List (Test2.lst)

### 8.2.4.3 Using the Oracle SQL Worksheet

The same functions you performed using SQL\*PLUS commands, you can also do using a graphical user interface (GUI).

To start the SQL Worksheet, do the following:

On your NT desktop click on **Start** with the left mouse button. Then select **Programs** to explode the programs menu. Select **Oracle Enterprise Manager** that opens the selection menu for a series of Oracle programs for Windows NT. Move the mouse pointer over the **SQL Worksheet** option and start this program by clicking on the left mouse button.

A new window is opened that prompts you for the following information:

- Username
- Password
- Service
- Connect As

Enter your user ID and password. As already done in previous sections of this scenario, we used tivoli for both user ID and password.

In the Service field enter the alias for the Oracle database, which was defined as Inventory. You may, however, also use the native name of the Oracle database, which is sid1. What terms to use depend on the definitions that were made during database creation, refer to Chapter 4, "Using TME 10 Inventory with Oracle for AIX" on page 63.

The Connect As field is already filled with the default value, which is Normal. You do not need to change this value.

Click on **OK** when you have entered all information. This will start the SQL Worksheet GUI.

The Oracle SQL Worksheet is divided into two parts. The upper part is for the output, while you enter your commands in the lower part of the window. You may enter the commands manually, or read in script files, that have been provided to you.

To show the relation between doing SQL queries using SQL\*PLUS and SQL Worksheet, enter the same select statement that was used in the example in Figure 207 on page 217.

To execute the command that you entered, click on the **Execute** button.

#### Note

If you do not know which of the buttons is the Execute button, move the mouse pointer over the offered button selection. On the lower right of the mouse pointer, there will be a small help window created, that contains a short explanation of the function of the button.

How the window looks when you execute the command is shown in the following figure:

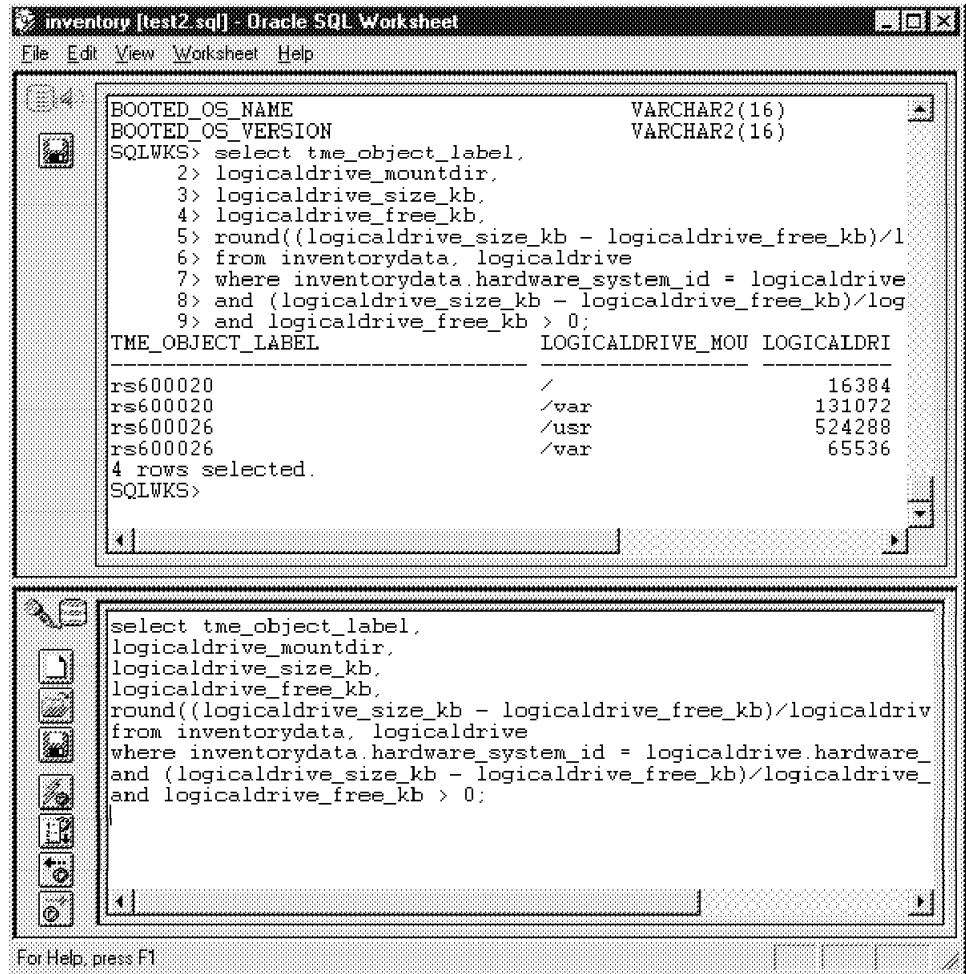


Figure 209. Inventory [Test2.sql] - Oracle Worksheet Window

The advantage of using a GUI is that you can scroll over to different sections of your output. This is especially important if the output width and length exceed the screen size.

There are other nice features that might be helpful. How to use the Command History, for example, is shown in the following figure.

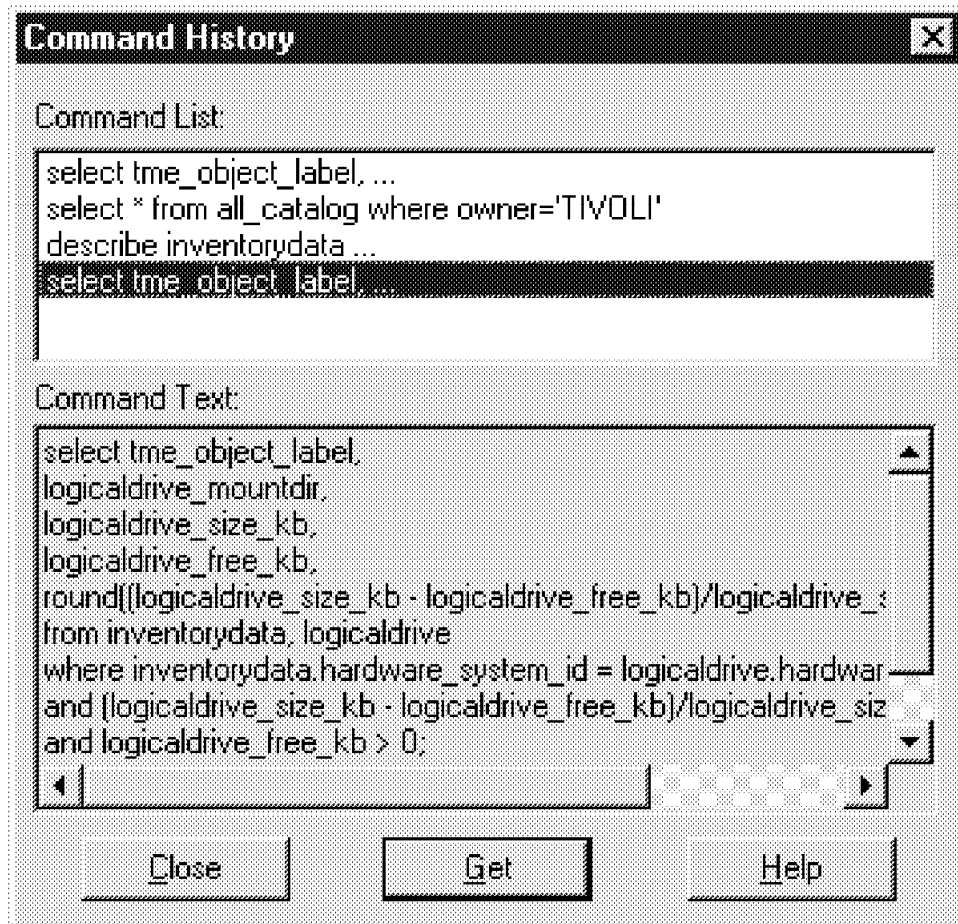


Figure 210. Command History Window

Move the cursor using the up or down key of the keyboard, or use the mouse pointer to select a previously issued command. Click on **Get** and then **Close** to close the Command History window. You may now execute the selected command on the Oracle SQL Worksheet.

#### 8.2.4.4 Using the Oracle Schema Manager

In order to do queries on the inventory database you need to know detailed information about the database model and the database structure. You certainly do not know all the different tables and their data fields by heart. Therefore, you need tools to query this information, before you start queries of the database data.

In the previous sections, you have seen various methods on how to get to this information. In this section, you will get acquainted with the Oracle Schema Manager, which is another graphical user interface.

On your NT desktop click on **Start** with the left mouse button. Then select **Programs** to explode the programs menu. Select **Oracle Enterprise Manager** that opens the selection menu for a series of Oracle programs for Windows NT. Move the mouse pointer over the **Schema Manager** option and start this program clicking on the left mouse button.

The following window will be started:



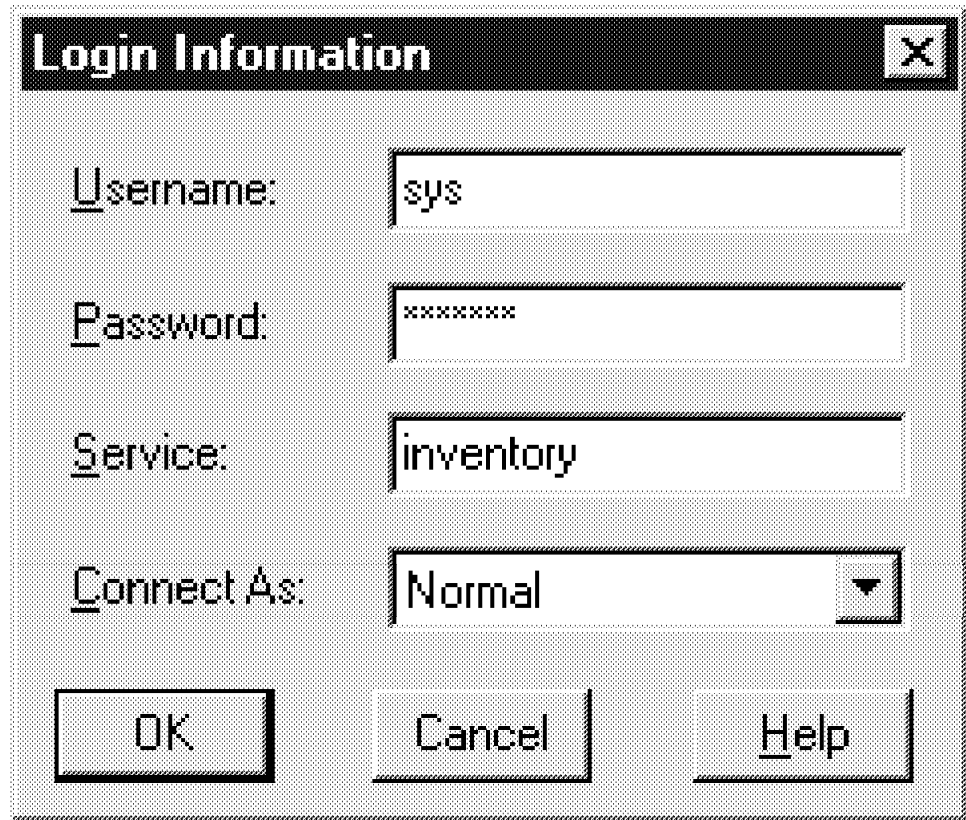


Figure 211. Login Information Window

Unfortunately, if you enter the user ID tivoli, you will get an error message.

The error message is shown in the following figure:



Figure 212. Oracle Schema Manager Window

To access the Schema Manager, you have to extend the privileges for the user ID tivoli. For this example you may also use an administrator user ID and password. So, enter a Username sys and the Password oracle. Also enter the database alias inventory as Service. Accept the default entry in the Connect As data field.

When you click on **OK**, the following figure will appear:

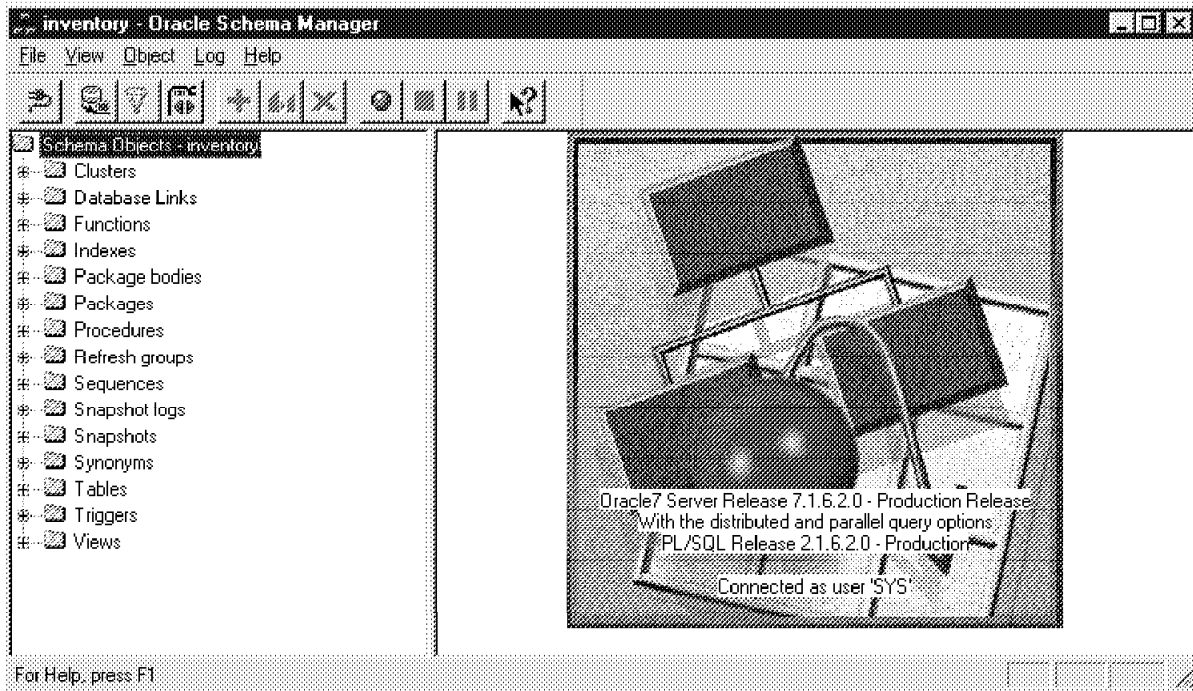


Figure 213. Inventory - Oracle Schema Manager Window

You may now start to discover what the database really looks like. If you move the mouse pointer over the window and point to a + sign that is next to an object you want to know more about, click on the left mouse button to open the tree structure.

You probably want to know more about the structure and content of the view INVENTORYDATA. When you expand the Views object, you will see several owners of database views. The ones that might be of interest to you are the TIVOLI views. Expand the TIVOLI object and you will see the whole list of all views. With the left mouse button click on INVENTORYDATA and you will see the following figure.

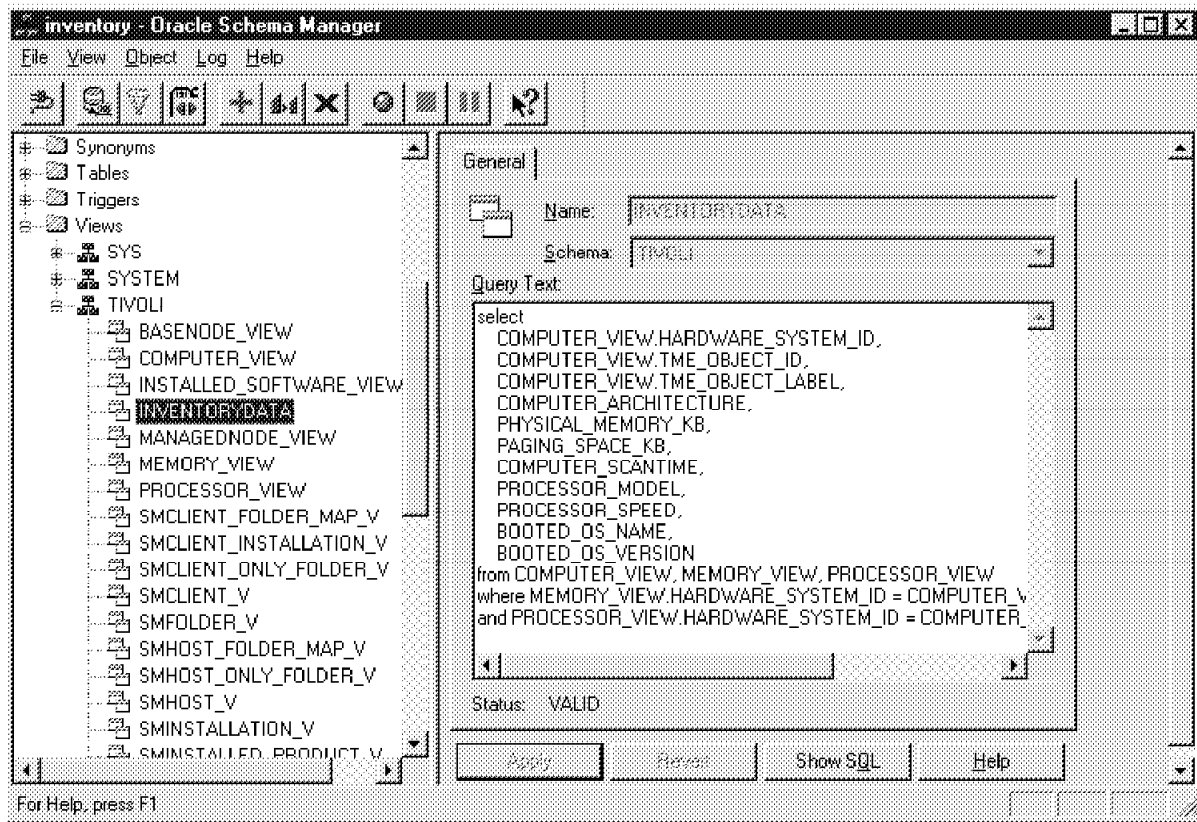


Figure 214. Inventory - Oracle Schema Manager Window

You can use the scroll bar to view areas of the window that are hidden.

On the next figure you will see an example of the structure of the TIVOLI COMPUTER\_SYSTEM table:

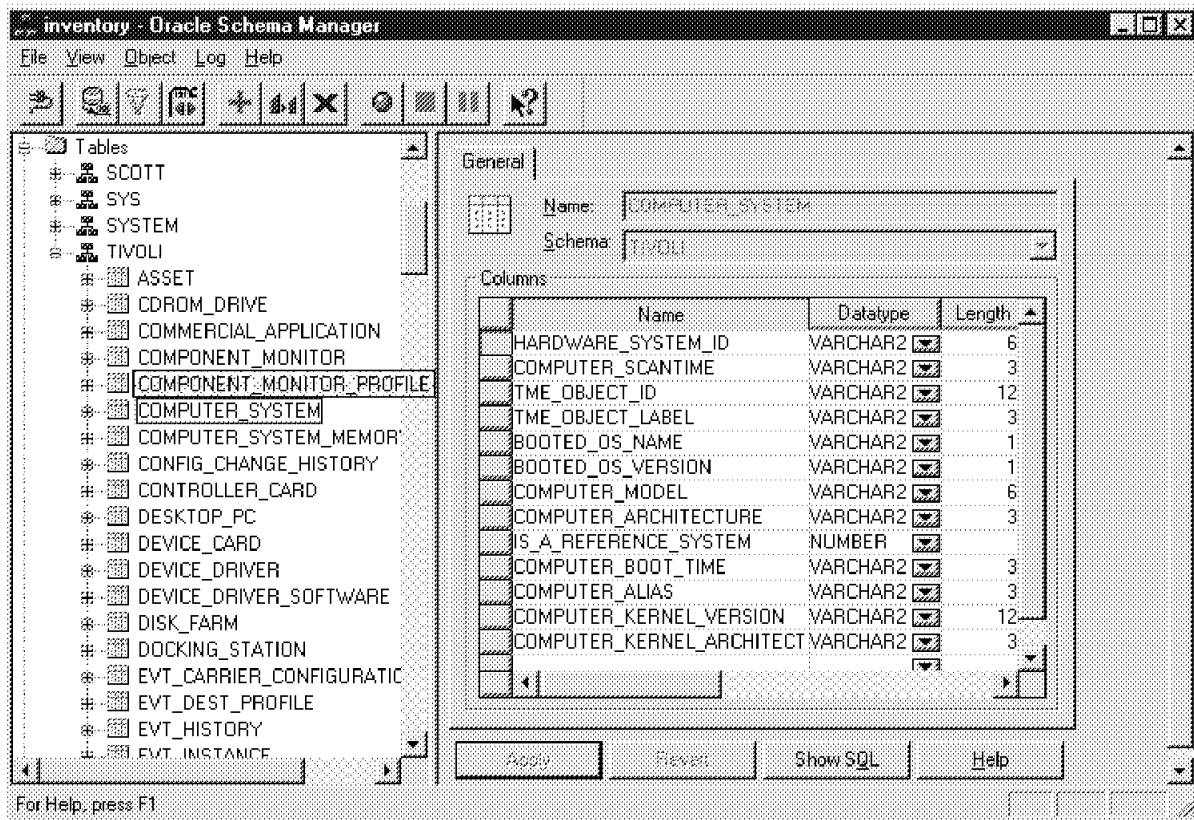


Figure 215. Inventory - Oracle Schema Manager Window

With the left mouse button you can expand the column width if you want to see sections that are hidden.

## 8.3 Working with a Sybase Database on a Windows NT Client

We will not go into detail explaining the tools for Sybase on a Windows NT machine, because this has already been done in other chapters. We will, however, refer you to those chapters when appropriate.

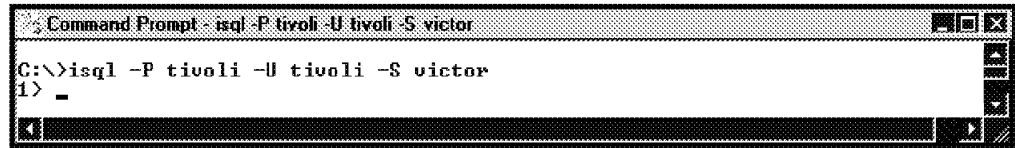
### 8.3.1 Working with the ISQL Interface

The equivalent of the SQL\*PLUS command line interface for Oracle for Windows NT 4.0 is the ISQL command line interface for Sybase Server Version 11 for Windows NT.

To start the ISQL interface, you issue the following command in a Windows command prompt:

```
ISQL -P password -U userid -S server
```

When the connection to your Sybase database is successfully established, you will see the following figure:

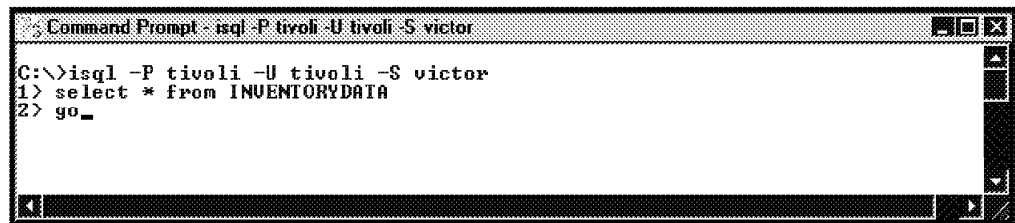


```
Command Prompt - isql -P tivoli -U tivoli -S victor
C:\>isql -P tivoli -U tivoli -S victor
i>
```

Figure 216. Database Connection Established Successfully

Now you can start entering select statements. The select statement can be entered on more than one line. Every time you press the Enter key, a new line is started. When your select statement is complete, type go and then press the Enter key.

An example of the select statement is shown in the following figure:



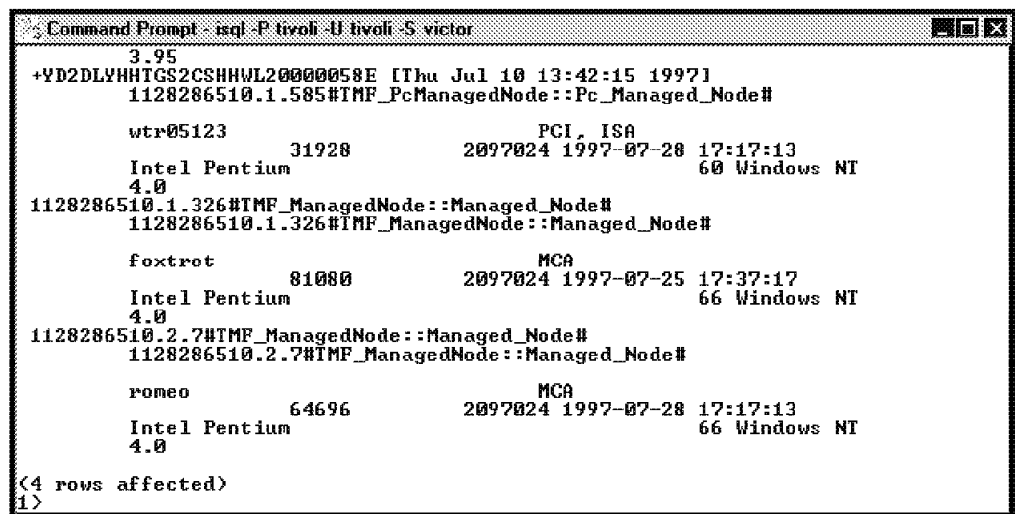
```
Command Prompt - isql -P tivoli -U tivoli -S victor
C:\>isql -P tivoli -U tivoli -S victor
i> select * from INVENTORYDATA
2> go
```

Figure 217. Select Statement for INVENTORYDATA

**Note**

Contrary to the SQL\*PLUS interface in Windows NT, the table name is case sensitive when you work with the ISQL interface.

A sample output of the query you just started is shown in the following figure:



```
Command Prompt - isql -P tivoli -U tivoli -S victor
3.95
+YD2DLYHHTGS2CSHHWL20000058E [Thu Jul 10 13:42:15 1997]
1128286510.1.585#IMF_PcManagedNode::Pc_Managed_Node#

wtr05123 PCI, ISA
 Intel Pentium 31928 2097024 1997-07-28 17:17:13
 4.0 60 Windows NT

1128286510.1.326#IMF_ManagedNode::Managed_Node#
1128286510.1.326#IMF_ManagedNode::Managed_Node#

foxtrot MCA
 Intel Pentium 81080 2097024 1997-07-25 17:37:17
 4.0 66 Windows NT

1128286510.2.7#IMF_ManagedNode::Managed_Node#
1128286510.2.7#IMF_ManagedNode::Managed_Node#

romeo MCA
 Intel Pentium 64696 2097024 1997-07-28 17:17:13
 4.0 66 Windows NT

<4 rows affected>
i>
```

Figure 218. Output from Select Statement for INVENTORYDATA

To end an ISQL session, enter quit and then press the Enter key.

### 8.3.2 Writing SQL Scripts

Writing an SQL script for a Sybase database is about the same as an SQL script for an Oracle database. Therefore, refer to 8.2.4.2, "Writing SQL Scripts to Generate Reports" on page 214. You should, however, remember that an SQL statement in a Sybase script is terminated by the command go. In an Oracle script, you used a semicolon instead.

### 8.3.3 Working with the WISQL32 Interface

The equivalent function of the Oracle for Windows NT 4.0 SQL Worksheet is the WISQL32 interface for Sybase Server Version 11 for Windows NT.

To start WISQL32 on a Windows NT client, run the following command from a Windows command prompt:

```
WISQL32
```

WISQL32 will start the WISQL32 interface. Before you can execute any SQL commands, you have to connect to your Sybase database. For instructions on how to connect to a Sybase database, refer to Chapter 3, "Using TME 10 Inventory with Sybase on AIX" on page 47. When you have established a successful connection, you may start queries on your Sybase database.

An example for the select statement is shown in the following figure:

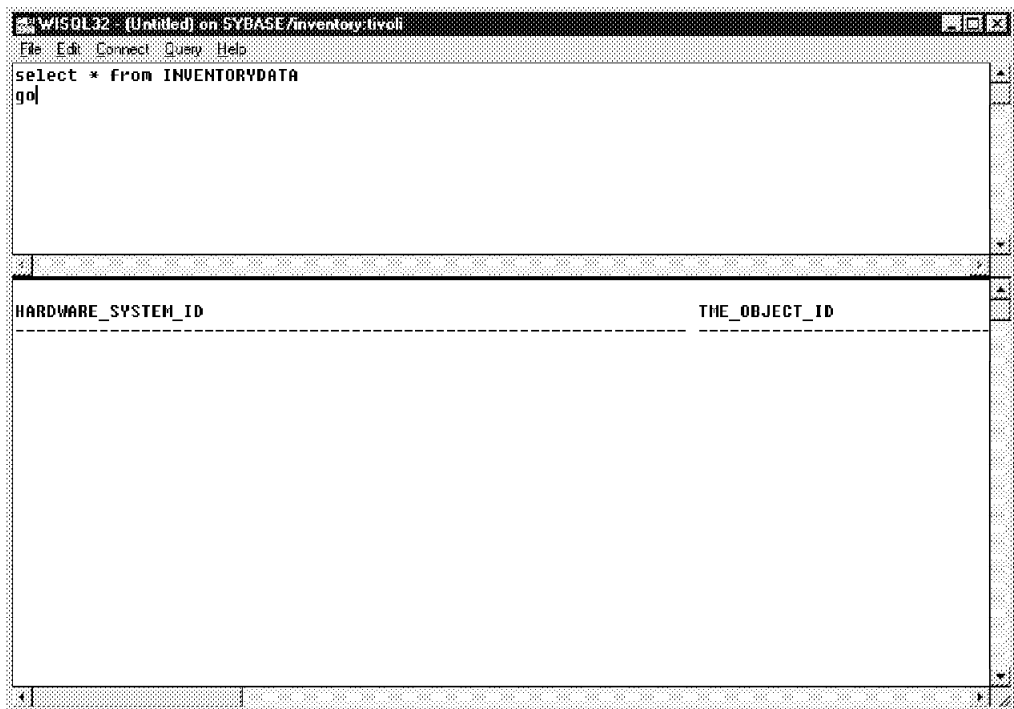


Figure 219. WISQL32 Window After Successful Connection

**Note**

The SQL statement is not terminated by a semicolon as in Oracle for Windows NT 4.0. You have to enter go after you have entered the SQL command.

### 8.3.4 Using the Sybase SQL Central

The equivalent of the Oracle for Windows NT 4.0 Schema Manager is the SQL Central interface for Sybase Server Version 11 for Windows NT. Refer to 8.2.4.4, "Using the Oracle Schema Manager" on page 220.

To start SQL Central, click on the **Start** icon of your Windows NT desktop and select SQL Central as shown in the following figure:

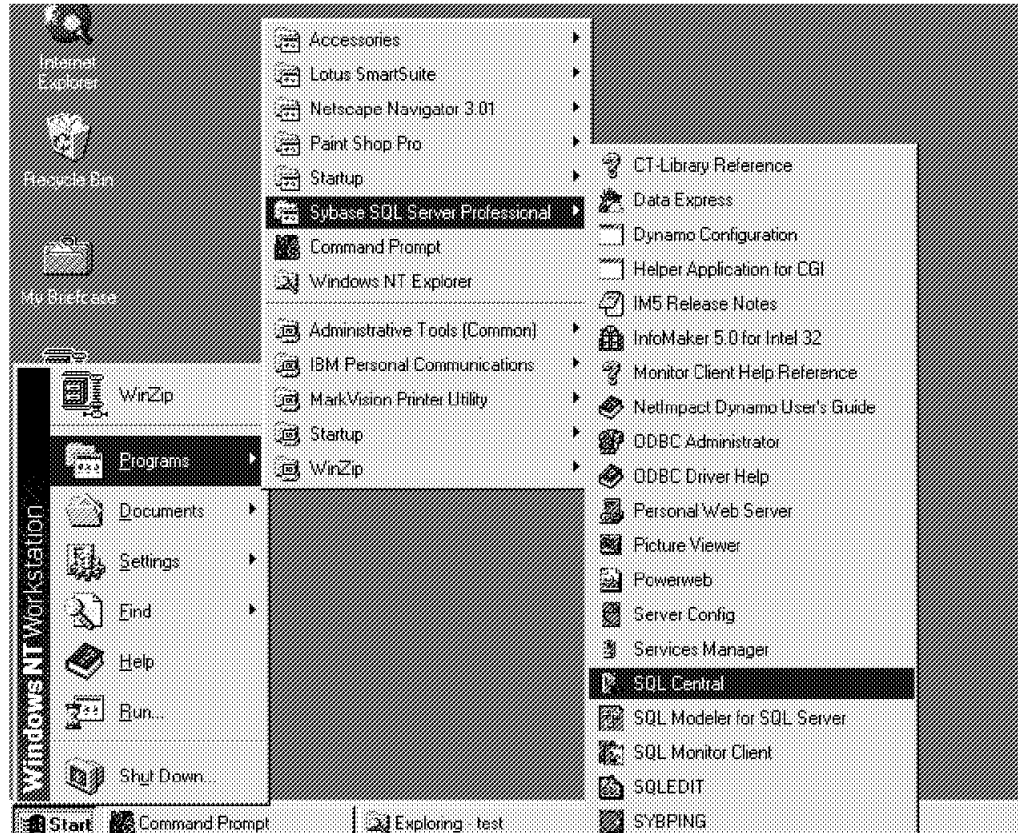


Figure 220. Start/Programs/Sybase SQL Server Professional\SQL Central Window

When SQL Central is started, the SQL Central application window is opened as shown in the following figure:

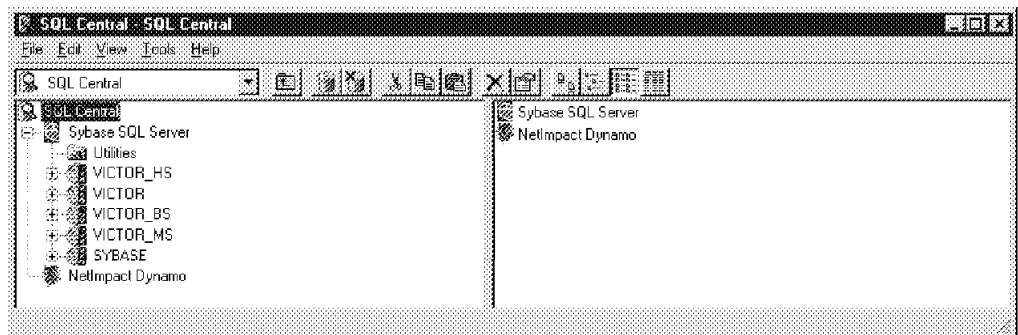


Figure 221. SQL Central Window

We want to learn more about the inventory database on victor. When you try to expand the victor object as shown in Figure 221, you will be requested to enter the required connect information as shown in the following figure:

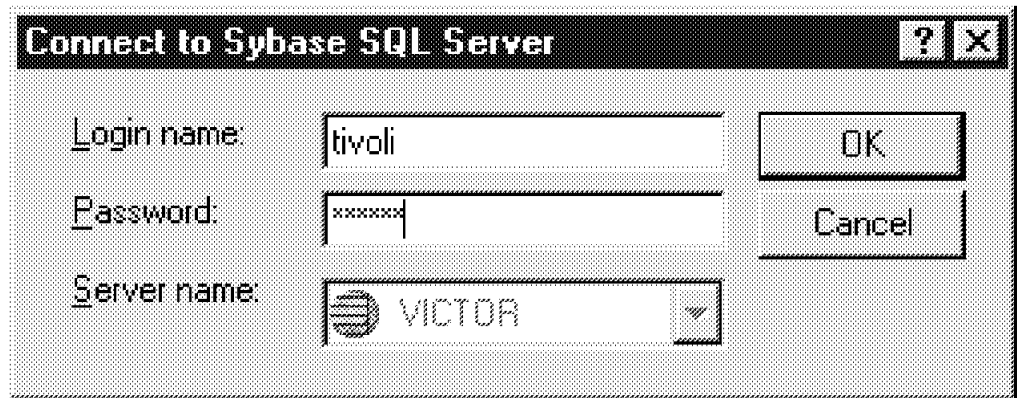


Figure 222. Connect to Sybase SQL Server Window

For information on how to get the correct connection information, refer to Chapter 3, "Using TME 10 Inventory with Sybase on AIX" on page 47.

When you are finished, click on **OK**, which will display Figure 223.

When you expand the inventory database tree, you will see an output of a complete list of all tables that are defined in your database as shown in the following figure:

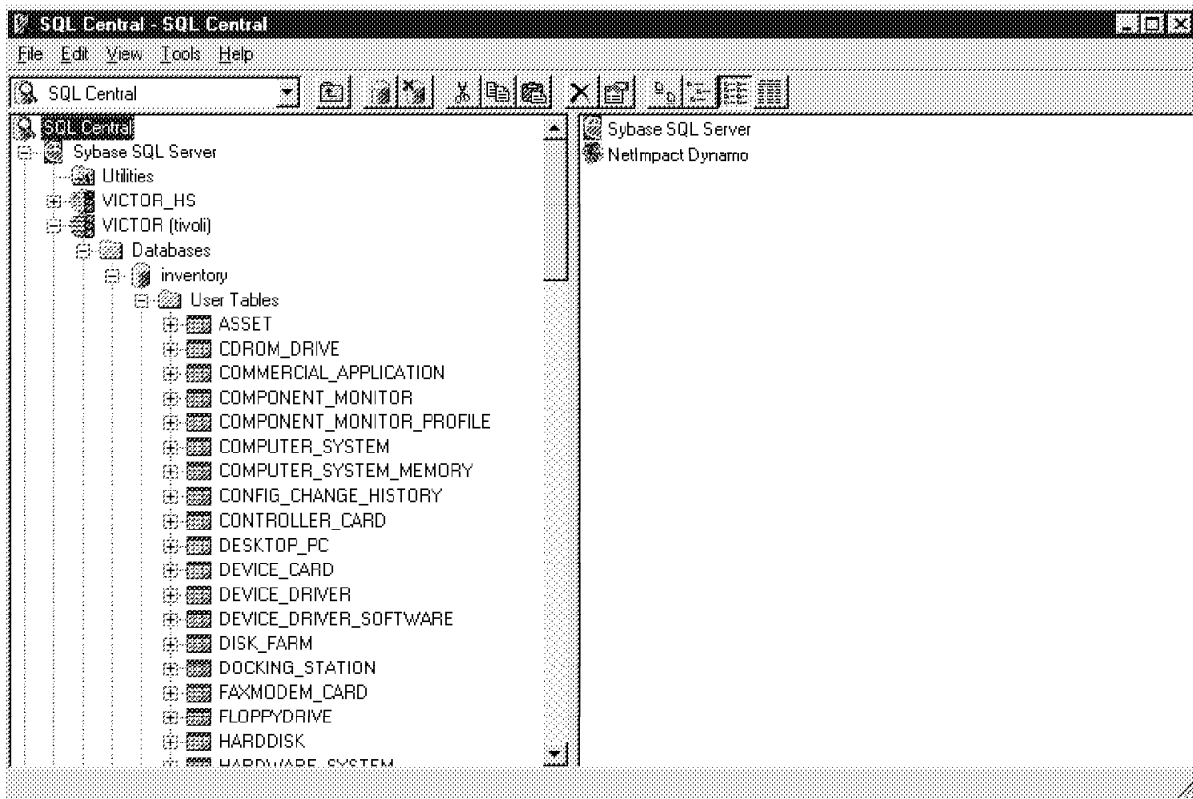


Figure 223. SQL Central Window - Inventory Database Tables

If you need to know more about the details of a certain database table, you have to double-click on the User Tables of the inventory database you want to expand. See also the left side of Figure 223.



We did this for the database table COMPUTER\_SYSTEM, as shown in the following figure:

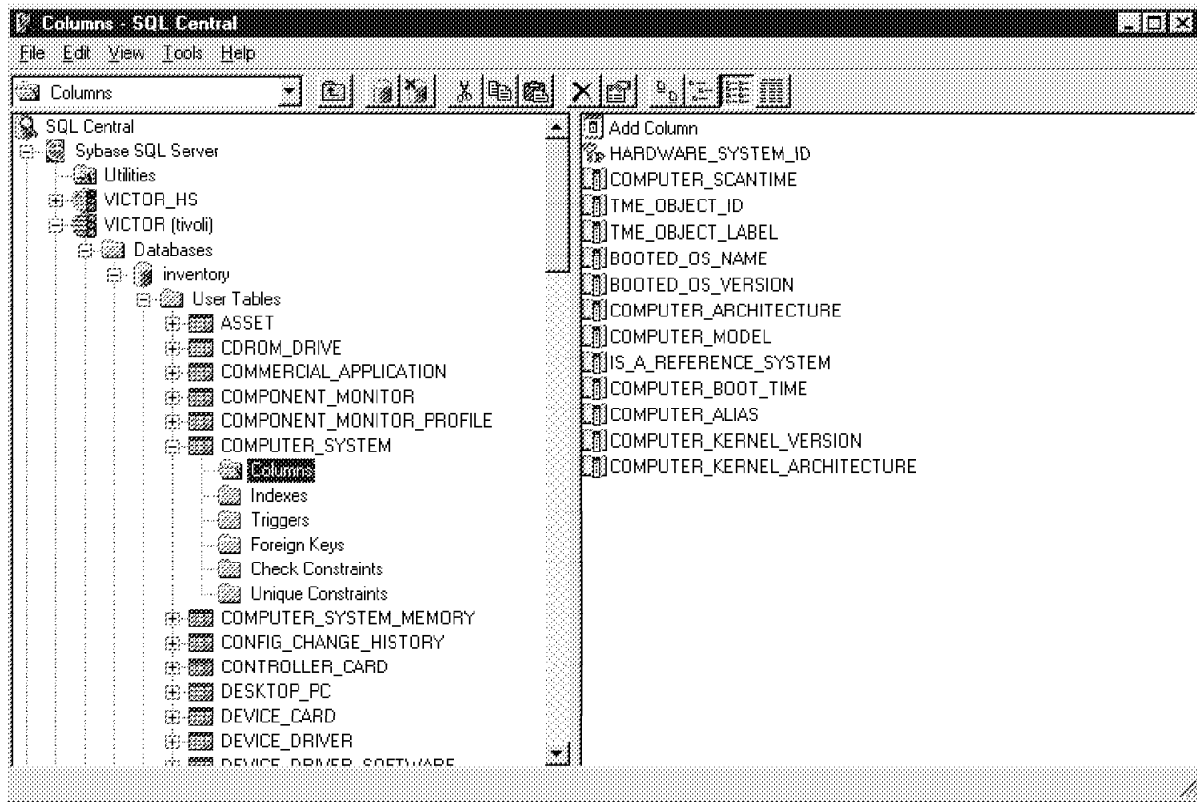


Figure 224. SQL Central Window - COMPUTER\_SYSTEM Details

## 8.4 Working with a Sybase Database on an AIX Client

We will not go into deep details, explaining the tools on an AIX machine for Sybase when this has already been done in other chapters. We will, however, refer you to those chapters when needed.

### 8.4.1 Working with the ISQL interface

The equivalent function to the SQL\*PLUS command line interface for Oracle for Windows NT 4.0 is the ISQL command line interface for Sybase for AIX. Usage and functionality of ISQL is the same as in Sybase Server Version 11 for Windows NT. For more details, refer to 8.3.1, "Working with the ISQL Interface" on page 224.

### 8.4.2 Writing SQL Scripts

AIX has the Korn Shell available as a built-in function. To start a Korn Shell script, you run the script from a command prompt.

As shown in Figure 225 on page 230, the first line in the script identifies the script as a Korn shell script and must be `#!/bin/ksh`.

```
#!/bin/ksh

This script creates the new table SW_LICENSES,
retrieves data from table INSTALLED_SOFTWARE_VERSION,
and inserts this data in the table SW_LICENSES

SYB_USER=tivoli
SYB_PASSWD=tivoli
.
.
.
go
```

Figure 225. Sample of a Korn Shell Script File

For more information on how to write scripts for Sybase for AIX, refer to Chapter 5, “Examples Using Asset Management with TME 10 Inventory” on page 107. There are several extensive examples in that chapter that will provide some help to you, if you want to write your own scripts.

### 8.4.3 Using a GUI

There are no equivalent functions to the WISQL32 and SQL Central GUIs available in Sybase for AIX. If you want to perform those functions, we recommend you do these functions from a Windows NT client.

---

## 8.5 Working with an Oracle Database on an AIX Client

We will not go into details, explaining the tools on an AIX machine for Oracle when this was already done in other chapters. We will, however, refer you to those chapters when needed.

### 8.5.1 Using SQL\*PLUS

Usage and functionality is the same as in Windows NT. For more details, refer to 8.2.4.1, “Using SQL\*PLUS” on page 210.

### 8.5.2 Writing SQL Scripts

For more information on how to write scripts for Oracle for AIX, refer to Chapter 4, “Using TME 10 Inventory with Oracle for AIX” on page 63. There are several extensive examples in that chapter that will provide some help to you, if you want to write your own scripts.

### 8.5.3 Using a GUI

There are no equivalent functions to the SQL Worksheet or Schema Manager GUIs available the Oracle for AIX If you want to perform those functions, we recommend you do these functions from a Windows NT client.

---

## Chapter 9. Summary

In this chapter we summarize some of the experiences from the scenarios performed in this redbook and give some advice on common matters involving TME 10 Inventory.

We also include experiences that are documented in the redbook *TME 10 Inventory 3.2 New Features and Database Support*, SG24-2135, which was released at the same time as this redbook.

You should notice that some of the features discussed in this summary, such as DB2 and MS SQL support, are only available in TME 10 Inventory 3.2 and later.

---

### 9.1 Selecting the TME 10 Inventory Server Platform

The most notable thing about selecting the TME 10 Inventory platform is that the TME 10 Inventory platform is not necessarily the RDBMS server platform.

For example, your TME 10 Inventory server may be on Windows NT whereas your RDBMS server is Sybase on AIX. In that case, you only need to make sure that there is a Sybase client for Windows NT.

What platform TME 10 Inventory runs on will mostly depend on the kind of systems you are using in your enterprise. For example, if you are using mainly Windows NT systems, your TME 10 Inventory server will most likely be running on Windows NT.

However, if you also have UNIX systems, you will have a choice of running the TME 10 Inventory server on Windows NT or UNIX. The selection of the server platform in that case will depend on factors such as stability, scalability and performance.

#### Note

You should notice that with TME 10 Inventory 3.1 and TME 10 Inventory 3.2 the only supported server platforms are UNIX and Windows NT. Again, you should also notice that this does not limit your choice of the RDBMS platform. This could be, for example, DB2 on an MVS system using a DDCS connection.

---

### 9.2 Selecting the Database Server Platform

There are a number of factors to consider when selecting which RDBMS system should be used to store data used by TME 10 Inventory.

First of all, you will need to use an RDBMS system that TME 10 Inventory actually supports. At the moment, the following RDBMS servers are supported:

- Sybase
- Oracle
- MS SQL Server
- DB2 for AIX

**Note**

Support for DB2 and MS SQL is only available in TME 10 Inventory 3.2 and later versions.

There is, however, another implication to consider which is connected to which platform you choose for your TME 10 Inventory server to run on: this platform must be supported by the database system you choose. To make things a little more complicated, your TME 10 Inventory server can be either a database server or a database client. Thus, the database system you choose must support the platform your TME 10 Inventory server is installed on either as a server, or as a client.

To understand this principle, let's have a closer look at how TME 10 Inventory (actually it's RIM) "talks" to the database. The following figure gives an overview:

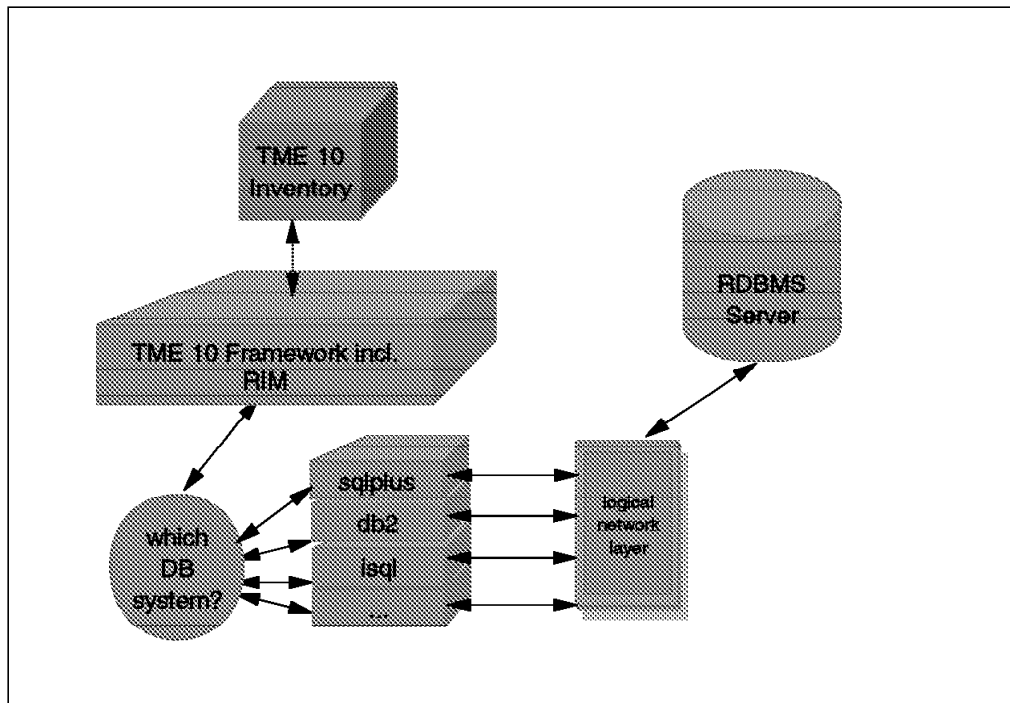


Figure 226. RIM Accessing an RDBMS System

All of the supported database systems provide a simple command line interface, that is usually used to perform operations in batch mode, such as creating tables, updating values or performing queries.

These command line interfaces have different names, depending on the database system; the above figure shows some of them:

- For Oracle the interface is called `sqlplus`.
- For DB2 for AIX the interface is called `db2`.
- For Sybase the interface is called `isql`.

These interfaces (in fact they are commands) are used no matter if you work on the database server or on a database client. The interface then determines the

server by looking up some kind of configuration record and then tries to establish a connection with the server.

This is usually done through some kind of logical network interface. For example, if you are on the database server itself, probably some interprocess communication method is used for the connection, whereas if you are on a client, some network protocol, such as TCP/IP is used.

For a user trying to access the RDBMS server, this means that the user does not have to worry about where the database server physically resides, or if he is on the server or a client.

TME 10 Inventory is such a user. Therefore, when selecting the database system it must be considered that the database interface - either the server or a client - is available on the platform where the TME 10 Inventory server is installed. For example, TME 10 Inventory may be installed on a Windows NT managed node, whereas the Sybase database server is installed on AIX 4.1.

In this case the requirement is that a client for Windows NT be available to provide the command line interface to access the RDBMS server (in this case isql) and that the Windows NT client can be connected to the AIX database server.

There are other factors that are usually predominant when it comes to the selection of the RDBMS system to use for TME 10 Inventory. As most customers already use RDBMS systems for other tasks, the decision of which RDBMS system to use has perhaps already been decided. Naturally, the first choice would be to use the RDBMS system already in use.

However, there are many environments, where more than one of the supported RDBMS systems are used. In that case, some time should be spent considering which of them to use for TME 10 Inventory. Since with all of the supported RDBMS systems you can have clients, having an RDBMS server on the platform where TME 10 Inventory is used does not necessarily mean that this is the RDBMS in which to store TME 10 Inventory data.

For example, a customer may have a Windows NT environment, where MS SQL Server is used as the preferred RDBMS system, while at the same time the customer is using Sybase and DB2 for AIX. Now, the customer decides to use TME 10 Inventory on Windows NT.

Naturally, the first choice would be to use MS SQL Server on Windows NT as the RDBMS server, right? Maybe not. While using MS SQL Server is a valid choice, there may be reasons that make it more desirable to use Sybase or DB2 for AIX. For example, one reason could be that the customer has a Distributed Database Connection Services (DDCS) connection already established between DB2 for AIX and DB2 for MVS and wants to store all data on the mainframe. This would be one reason to store TME 10 Inventory data in DB2 for AIX. In that case, a Windows NT client for DB2 for AIX needs to be installed on the system where the TME 10 Inventory server resides.

### 9.2.1 Can the RDBMS System Be Shared?

There might be the situation that one does not want to have a dedicated RDBMS server that is solely used to store TME 10 Inventory information. In that case, an existing RDBMS server can be shared with other applications.

Technically, this is usually not a problem, as the RDBMS system can handle multiple databases. However, there are other areas to consider when making this decision, such as:

- Performance
- Security
- Scalability
- Availability

As with any application, if you share the RDBMS system between TME 10 Inventory and another application, perhaps another TME 10 application, such as TME 10 Enterprise Console or any application that uses an RDBMS system, the performance will decrease because you are sharing resources.

From a security perspective you need to consider if the data that TME 10 Inventory stores must not be seen by user applications or users. The main security risk is probably the "tivoli" RDBMS user that TME 10 Inventory creates and uses to create the TME 10 Inventory tables and to access data.

The problem with that user is that its password (also set to "tivoli") is known and therefore almost anybody can access the TME 10 Inventory data. If you intend to share the RDBMS with another application you should at least not create any other tables or databases owned by the "tivoli" user.

From a scalability perspective, sharing the RDBMS might limit the scalability options for the RDBMS you have. This is caused by the fact, that the database topology might be different for TME 10 Inventory and the application it is sharing the RDBMS with.

Hence, you may be only able to fit the RDBMS topology to one of the applications. For example, you might want to link some Sybase NT RDBMS servers to a central Sybase AIX server for TME 10 Inventory purposes; however, the other application needs to connect to a central Sybase NT system.

Probably the most important issue in trying to share the RDBMS system is the availability of the RDBMS server. For example, for TME 10 Inventory purposes a 99% availability of the RDBMS system might not be necessary but for another application it might be. Also, concurrent applications sharing the same RDBMS system are likely to influence the degree of availability. For example, the sharing application might request a restart of the RDBMS system, but this action affects the availability of the RDBMS for TME 10 Inventory.

### 9.2.2 RDBMS Scalability

RDBMS scalability depends on the RDBMS system you choose as your TME 10 Inventory database server. However, the RDBMS systems supported by TME 10 Inventory (actually by RIM) should have similar scalability options.

There are different scenarios when talking about RDBMS scalability. In the first scenario, for example, the RDBMS system is DB2 for AIX.

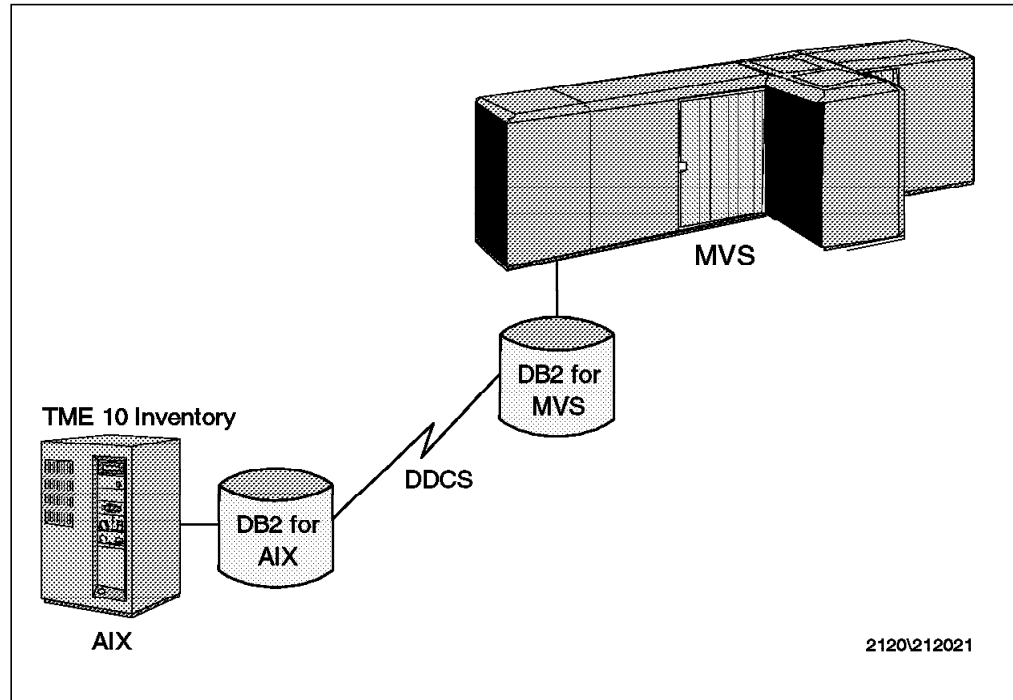


Figure 227. DB2 for AIX Using a DDCS Connection

In the above example, we want to store our data in DB2 for MVS eventually (for example, for security or performance reasons). As we cannot store data in DB2 for MVS from TME 10 Inventory directly, we can establish a DDCS connection between a DB2 for AIX server and the DB2 for MVS server, thus eventually storing all data in DB2 for MVS. In environments where DB2 for MVS is an established platform that might have certain advantages, such as using, for example, backup procedures on MVS.

An example for scalability with more than one TME 10 Inventory server is shown in the following figure:

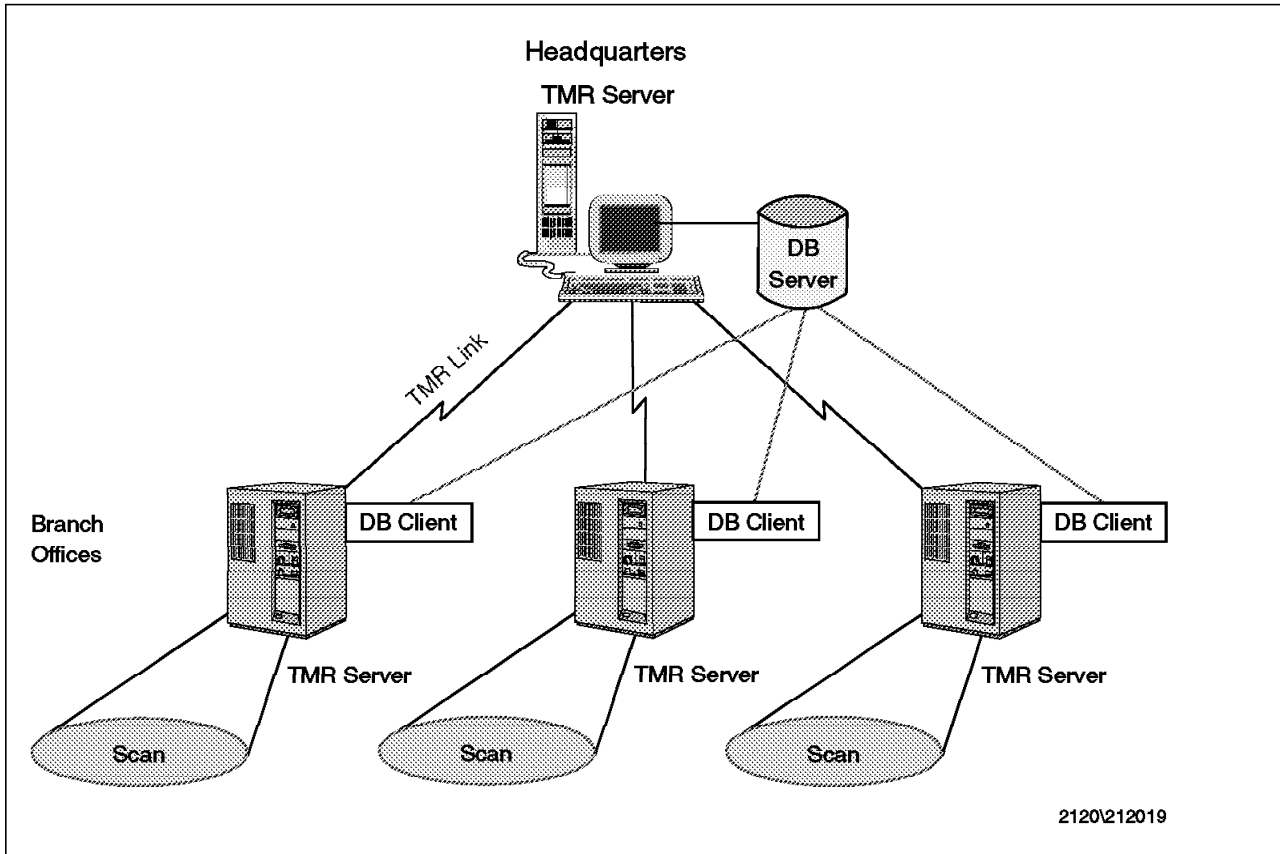


Figure 228. TME 10 Inventory Scalability with Multiple TMR Servers

In the above example there are distributed TMR servers that also serve as TME 10 Inventory servers in branch offices. All branch office TMR servers are linked through TMR connections with the headquarters central TMR server.

In this scenario, there is only one central RDBMS server in the headquarters and all distributed TME 10 Inventory servers access this RDBMS system by using an RDBMS client.

This is the easiest way of linking distributed TME 10 Inventory servers; however, this method has certain implications. For example, the TME 10 Inventory database tables and views must be created only once, as the clients use the same database. By creating separate database instances for each TME 10 Inventory server you can avoid this problem; however, this may conflict with the goal of storing inventory data from all machines in a central database.

Also, if the RDBMS server in the headquarters is down, none of the distributed TME 10 Inventory servers can do any scanning, as the database needs to be available.

As TME 10 Inventory uses TME object labels and object IDs as keys in the SQL database, storing data from different TMRs in one central RDBMS should usually not be a problem, as the object labels and IDs should be unique.

The figure also shows TMR connections between the central and the distributed TMR servers. This is not necessary to store data in the RDBMS; however, it will usually be the case when using TME 10 as the enterprise management solution. When using Lightweight Client Framework (LCF) this structure will look different.



Another scenario would be to use distributed RDBMS servers for each TME 10 Inventory server as shown in the following figure:

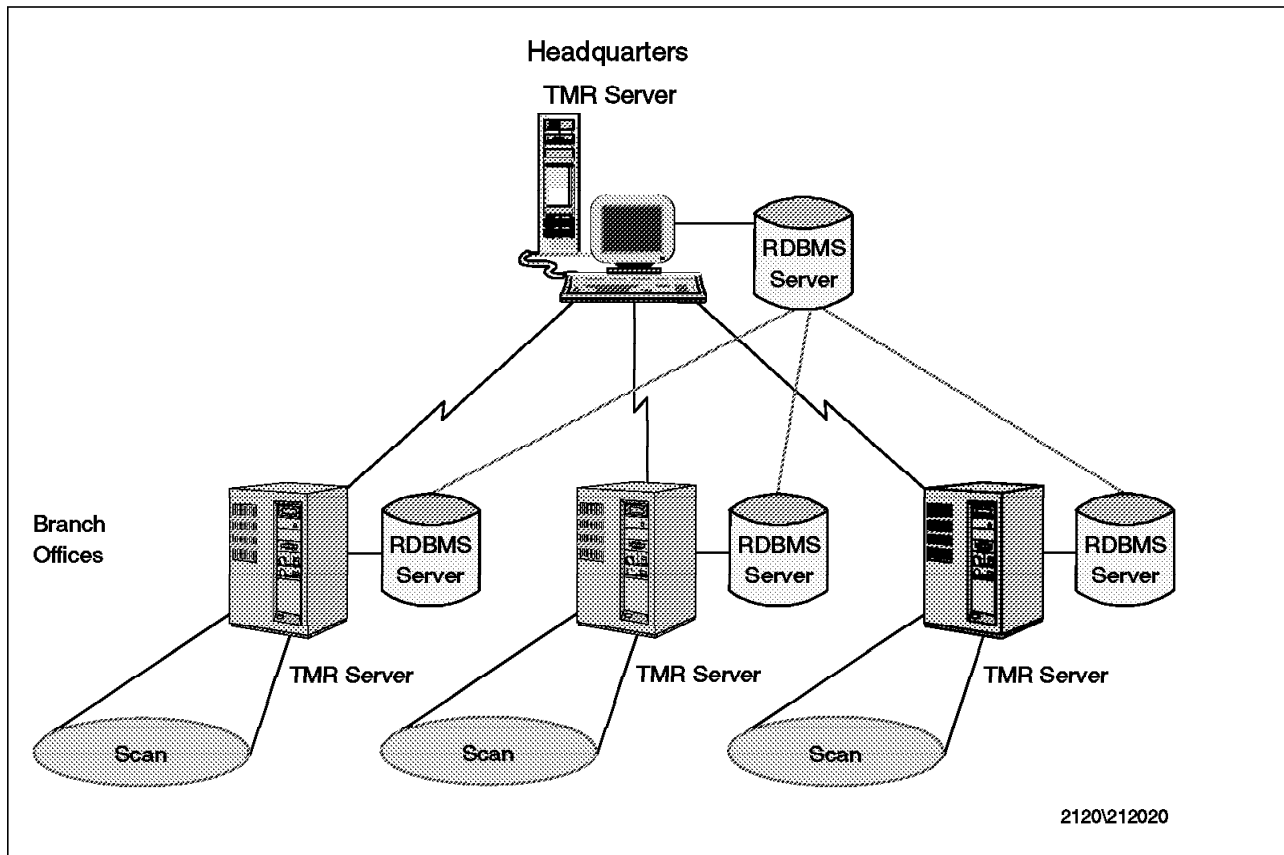


Figure 229. TME 10 Inventory with Distributed RDBMS Servers

The above scenario differs from the previous scenario in that you have a dedicated RDBMS server for each branch office. While this probably creates more work in configuring and maintaining the RDBMS servers, it will increase availability and performance, as the branch offices are not dependent on the headquarter's central RDBMS server.

This scenario requires that there be a mechanism in the RDBMS that allows you to synchronize data from a number of distributed RDBMS servers and consolidate it in the headquarters server, so that the headquarters server contains all data.

The same requirements, as far as unique object identifiers are concerned, apply as they did in the previous scenario.

The main thing to think about when setting up this kind of environment is to make sure that the RDBMS system you choose supports RDBMS server-to-server links.

---

## 9.3 Integrating Existing Inventory Solutions

In an environment where you plan to deploy TME 10 and TME 10 Inventory, there may be some inventory applications already in use. For example, you might be already using Microsoft SMS to gather inventory information on Windows platforms, NetFinity to do the same on Windows and OS/2 or maybe use the inventory function in TME 10 Software Distribution 3.1.3 (the former NetView DM products).

If you are using a desktop management suite that is supported by TME 10 LAN Access, such as Microsoft SMS, you can easily integrate existing inventory solution into an enterprise systems management solutions provided by TME 10, including TME 10 Inventory as the enterprise inventory solution.

If you are using a desktop management suite that is not supported by TME 10 LAN Access or you are using a custom inventory tool you might want to exploit the MPM API to create your own interface to TME 10 Inventory.

### 9.3.1 Integrating Inventory Information from NetView Distribution Manager

The TME 10 Software Distribution 3.1.3 products (formerly known as NetView DM) provide their own hardware and software inventory.

The inventory scanning mechanisms are the same as those used by IBM NetFinity; however, the output format is slightly different and tailored for use with NetView DM.

NetFinity allows importing inventory files from NetView DM. Therefore, if you are currently using the inventory feature in NetView DM and want to migrate this inventory to TME 10 Inventory easily, you can read the existing inventory files into NetFinity and then propagate them through TME 10 LAN Access to TME 10 Inventory.

---

## 9.4 Signature Scanning vs Custom Scanners

TME 10 Inventory software scanning is based on signature files. The principle behind this is that all files on the target node are scanned and then their attributes are compared to a list of attributes known about certain software products.

For example, if you know that application X always contains a file called APPLX.EXE with a size of 103443 bytes, you would compare all files on the scanned node to see if you can find a file matching the criteria. If the file is found, the scanner will assume that the software is installed.

The advantage of this procedure is that you can immediately start scanning and can easily add new signatures to match additional software.

However, this approach will always follow the principle that the scanners can only find what they are looking for.

On the other hand, a lot of operating systems, especially UNIX flavors have their own internal mechanisms to administer the installed software. For example, software on AIX is usually installed using the `installp` command which results in the software installation status being stored in an AIX internal database.

The contents of that database can be queried and a list of installed software products and their versions can be determined. This approach is based on the principle of showing everything that is there as opposed to only searching for what you know.

As a result, under certain circumstances it may be worthwhile to write a custom scanner that takes advantage of the operating system facilities and then transforms the output from the operating system command into a MIF file.

You can also combine signature and custom scanning, which is perhaps the best solution for a thorough scanning.

---

## **9.5 TME 10 Inventory and TME 10 Software Distribution**

When you look at the data model for TME 10 Inventory 3.2 you will see that there are some tables that are supposed to be used by TME 10 Software Distribution, for example, `INSTALLED_SOFTWARE_COMPONENT`. TME 10 Software Distribution 3.1 does not use these tables in TME 10 Inventory; however, it is expected that the next version of TME 10 Software Distribution (3.2) will use these fields in the configuration repository.



## Appendix A. Oracle Network Manager

Oracle Network Manager creates two files, `tnsname.ora` and `sqlnet.ora`, for the RIM host and three files, `tnsname.ora`, `sqlnet.ora`, and `listener.ora`, for the RDBMS server.

```

Filename.....: tnsnames.ora
Name.....: LOCAL_REGION.world
Date.....: 30-JUL-97 16:49:12

sid1.world =
 (DESCRIPTION =
 (ADDRESS_LIST =
 (ADDRESS =
 (COMMUNITY = sid1.world)
 (PROTOCOL = TCP)
 (Host = rs600024)
 (Port = 1526)
)
)
 (CONNECT_DATA =
 (SID = sid1)
 (GLOBAL_NAME = sid1.world)
)
)
```

Figure 230. `tnsname.ora`

```

Filename.....: sqlnet.ora
Name.....: sid1.world
Date.....: 30-JUL-97 16:49:12

AUTOMATIC_IPC = ON
TRACE_LEVEL_CLIENT = OFF
SQLNET.EXPIRE_TIME = 0
NAMES.DEFAULT_DOMAIN = world
NAME.DEFAULT_ZONE = world
SQLNET.CRYPTO_SEED = "-7508645-7367622"
```

Figure 231. `sqlnet.ora`



---

## Appendix B. Special Notices

This publication is intended to help IBMers, customers, and business partners use the TME 10 products successfully. The information in this publication is not intended as the specification of any programming interfaces that are provided by the TME 10 products. See the PUBLICATIONS section of the IBM Programming Announcement for the TME 10 products for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

The following document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these

names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

|             |                  |
|-------------|------------------|
| AIX         | DB2              |
| IBM         | NetFinity        |
| OS/2        | PowerPC 601      |
| PowerPC 604 | PowerPC          |
| PS/2        | RISC System/6000 |
| RS/6000     | ThinkPad         |

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.



---

## Appendix C. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

---

### C.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How to Get ITSO Redbooks" on page 247.

- *The TME 10 Deployment Cookbook: Courier and Friends*, SG24-4976
- *TME 10 Cookbook for AIX: Systems Management and Networking Applications*, SG24-4867
- *Systems Management from an NT Server Point of View*, SG24-4723
- *Software Distribution for AIX: A Solution for Installation and Configuration of Pristine AIX Environments*, SG24-4508.
- *NetView DM/6000 Release 1.2 Agents and Advanced Scenarios*, GG24-4490
- *TME 10 Inventory 3.2 New Features and Database Support*, SG24-2135

---

### C.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

| CD-ROM Title                                          | Subscription Number | Collection Kit Number |
|-------------------------------------------------------|---------------------|-----------------------|
| System/390 Redbooks Collection                        | SBOF-7201           | SK2T-2177             |
| Networking and Systems Management Redbooks Collection | SBOF-7370           | SK2T-6022             |
| Transaction Processing and Data Management Redbook    | SBOF-7240           | SK2T-8038             |
| AS/400 Redbooks Collection                            | SBOF-7270           | SK2T-2849             |
| RS/6000 Redbooks Collection (HTML, BkMgr)             | SBOF-7230           | SK2T-8040             |
| RS/6000 Redbooks Collection (PostScript)              | SBOF-7205           | SK2T-8041             |
| Application Development Redbooks Collection           | SBOF-7290           | SK2T-8037             |
| Personal Systems Redbooks Collection                  | SBOF-7250           | SK2T-8042             |

---

### C.3 Other Publications

The following publications are also relevant as further information sources. They are available from their respective manufacturers, in this case Oracle and Sybase, respectively:

- *Oracle 7 Installation Guide*, Document ID A53258
- *Oracle 7 Administrator's Guide*, Document ID A42808
- *Oracle 7 Network Manager Administrator's Guide*, Document ID A16264
- *Configuring Sybase SQL Server for IBM RISC System/6000 AIX*, Document ID 35840-01-1102-01
- *Release Bulletin Sybase SQL Server Release 11.0.2 for IBM RISC System/6000 AIX*, Document ID 72410-01-1102-01
- *Installing Sybase Products on UNIX Platforms*, Document ID 34258-01-0101-01



---

## How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at <http://www.redbooks.ibm.com>.

---

## How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

To get LIST3820s of redbooks, type one of the following commands:

```
TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
```

To get BookManager BOOKs of redbooks, type the following command:

```
TOOLCAT REDBOOKS
```

To get lists of redbooks, type one of the following commands:

```
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
```

To register for information on workshops, residencies, and redbooks, type the following command:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
```

For a list of product area specialists in the ITSO: type the following command:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
```

- **Redbooks Web Site on the World Wide Web**

<http://w3.itso.ibm.com/redbooks>

- **IBM Direct Publications Catalog on the World Wide Web**

<http://www.elink.ibm.link.ibm.com/pb1/pb1>

IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to [announce@webster.ibm.link.ibm.com](mailto:announce@webster.ibm.link.ibm.com) with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

### Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (<http://www.redbooks.ibm.com/redpieces.htm>). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

---

## How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** — send orders to:

|                        |                                       |                                         |
|------------------------|---------------------------------------|-----------------------------------------|
| In United States:      | <b>IBMMAIL</b><br>usib6fpl at ibmmail | <b>Internet</b><br>usib6fpl@ibmmail.com |
| In Canada:             | caibmbkz at ibmmail                   | lmannix@vnet.ibm.com                    |
| Outside North America: | dkibmbsh at ibmmail                   | bookshop@dk.ibm.com                     |

- **Telephone orders**

|                           |                               |
|---------------------------|-------------------------------|
| United States (toll free) | 1-800-879-2755                |
| Canada (toll free)        | 1-800-IBM-4YOU                |
| Outside North America     | (long distance charges apply) |
| (+45) 4810-1320 - Danish  | (+45) 4810-1020 - German      |
| (+45) 4810-1420 - Dutch   | (+45) 4810-1620 - Italian     |
| (+45) 4810-1540 - English | (+45) 4810-1270 - Norwegian   |
| (+45) 4810-1670 - Finnish | (+45) 4810-1120 - Spanish     |
| (+45) 4810-1220 - French  | (+45) 4810-1170 - Swedish     |

- **Mail Orders** — send orders to:

|                                                                                                      |                                                                                |                                                                      |
|------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|----------------------------------------------------------------------|
| IBM Publications<br>Publications Customer Support<br>P.O. Box 29570<br>Raleigh, NC 27626-0570<br>USA | IBM Publications<br>144-4th Avenue, S.W.<br>Calgary, Alberta T2P 3N5<br>Canada | IBM Direct Services<br>Sortemosevej 21<br>DK-3450 Allerød<br>Denmark |
|------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|----------------------------------------------------------------------|

- **Fax** — send orders to:

|                           |                                         |
|---------------------------|-----------------------------------------|
| United States (toll free) | 1-800-445-9269                          |
| Canada                    | 1-403-267-4455                          |
| Outside North America     | (+45) 48 14 2207 (long distance charge) |

- **1-800-IBM-4FAX (United States) or (+1)001-408-256-5422 (Outside USA)** — ask for:

Index # 4421 Abstracts of new redbooks  
Index # 4422 IBM redbooks  
Index # 4420 Redbooks for last six months

- **Direct Services** - send note to [softwareshop@vnet.ibm.com](mailto:softwareshop@vnet.ibm.com)

- **On the World Wide Web**

|                                 |                                                                                 |
|---------------------------------|---------------------------------------------------------------------------------|
| Redbooks Web Site               | <a href="http://www.redbooks.ibm.com">http://www.redbooks.ibm.com</a>           |
| IBM Direct Publications Catalog | <a href="http://www.elink.ibm.com/pbl/pbl">http://www.elink.ibm.com/pbl/pbl</a> |

- **Internet Listserver**

With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserv. To initiate the service, send an e-mail note to [announce@webster.ibm.com](mailto:announce@webster.ibm.com) with the keyword `subscribe` in the body of the note (leave the subject line blank).

---

### Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (<http://www.redbooks.ibm.com/redpieces.htm>). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

---

## IBM Redbook Order Form

Please send me the following:

| Title | Order Number | Quantity |
|-------|--------------|----------|
|-------|--------------|----------|

|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

---

|            |           |
|------------|-----------|
| First name | Last name |
|------------|-----------|

---

Company

---

Address

---

|      |             |         |
|------|-------------|---------|
| City | Postal code | Country |
|------|-------------|---------|

---

|                  |                |            |
|------------------|----------------|------------|
| Telephone number | Telefax number | VAT number |
|------------------|----------------|------------|

• Invoice to customer number \_\_\_\_\_

• Credit card number \_\_\_\_\_

---

Credit card expiration date

Card issued to

Signature

**We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.**



---

## Glossary

The IBM Software Glossary that contains terminology for software, the Internet, and TME 10 products from Tivoli, is available at the Web site <http://www.networking.ibm.com/nsg/nsgmain.htm>. You can use this Web site to search for the definition of a term or retrieve the entire glossary in PDF or PostScript format.

This glossary is an ever-evolving document that defines technical terms used in the documentation for many IBM software products. It includes IBM product terminology as well as selected terms from other sources.





## List of Abbreviations

|              |                                                             |              |                                                                   |
|--------------|-------------------------------------------------------------|--------------|-------------------------------------------------------------------|
| <b>ACF</b>   | T/C Adapter Configuration Facility                          | <b>JMAPI</b> | Java Management Application Programming Interface                 |
| <b>ADE</b>   | Application Development Environment                         | <b>KB</b>    | kilobytes                                                         |
| <b>AEF</b>   | Application Extension Facility                              | <b>MB</b>    | megabytes                                                         |
| <b>aka</b>   | also known as                                               | <b>MDist</b> | multiplexed distribution                                          |
| <b>AMS</b>   | Application Management Specification                        | <b>MIF</b>   | Management Information File                                       |
| <b>API</b>   | Application Programming Interface                           | <b>MLM</b>   | Mid-Level Manager                                                 |
| <b>BARC</b>  | before, after, removal, and commit configuration program(s) | <b>MSB</b>   | Multiple Services Broker                                          |
| <b>BAROC</b> | BAsic Recorder of Objects in C                              | <b>MTS</b>   | Multi-thread Server (Oracle)                                      |
| <b>BDT</b>   | Bulk Data Transfer                                          | <b>NFS</b>   | Network File System                                               |
| <b>BOA</b>   | Basic Object Adapter                                        | <b>NIS</b>   | Network Information Services (formerly called "NFS yellow pages") |
| <b>CDF</b>   | component description file                                  | <b>OID</b>   | object identifier                                                 |
| <b>CORBA</b> | Common Object Request Broker Architecture                   | <b>OMG</b>   | Object Management Group                                           |
| <b>DES</b>   | Data Encryption Standard                                    | <b>ORB</b>   | object request broker                                             |
| <b>DHCP</b>  | Dynamic Host Configuration Protocol                         | <b>PDF</b>   | Portable Document File or Product Description File                |
| <b>DM</b>    | Distribution Manager (or Management)                        | <b>RCS</b>   | Revision Control System                                           |
| <b>DMI</b>   | Desktop Management Interface                                | <b>RDBMS</b> | relational database management system                             |
| <b>DMTF</b>  | Desktop Management Task Force                               | <b>RIM</b>   | RDBMS interface module                                            |
| <b>DSL</b>   | Dialog Specification Language                               | <b>SID</b>   | Oracle System Identifier                                          |
| <b>EIF</b>   | Event Integration Facility                                  | <b>SMP</b>   | Symmetrical Multiprocessor                                        |
| <b>FFST</b>  | First Failure Support Technology                            | <b>TEC</b>   | TME 10 Enterprise Console                                         |
| <b>IBM</b>   | International Business Machines Corporation                 | <b>TLL</b>   | Task Library Language                                             |
| <b>IDL</b>   | Interface Definition Language                               | <b>TME</b>   | Tivoli Management Environment                                     |
| <b>IIS</b>   | Microsoft Internet Information Server                       | <b>TMF</b>   | Tivoli Management Framework                                       |
| <b>IMS</b>   | Internet Management Specification                           | <b>TMP</b>   | Tivoli Management Platform (obsolete)                             |
| <b>ITSO</b>  | International Technical Support Organization                | <b>TMR</b>   | Tivoli Management Region                                          |
|              |                                                             | <b>TNS</b>   | Transparent Network Substrate (Oracle)                            |
|              |                                                             | <b>TNWR</b>  | Tivoli NetWare repeater                                           |
|              |                                                             | <b>TRIP</b>  | Tivoli Remote Execution Service                                   |
|              |                                                             | <b>UNC</b>   | Universal Naming Convention                                       |



---

## Index

### Special Characters

/.rhosts file 69  
/etc/inittab 87  
/etc/oratab 72, 86  
/home/oracle 70, 71  
/oracle\_link 71  
/usr/sbin 70, 86  
/usr/lib/X11 87  
/usr/local/oracle 86, 95  
/usr/local/oracle/bin/dbshut 73  
/usr/local/oracle/bin/dbstart 72  
/usr/local/sybase 30, 49, 168  
/usr/local/sybase/bin/isql 168  
/usr/local/sybase/interfaces 52  
/usr/local/Tivoli 86  
/usr/local/Tivoli/bin/aix4-r1/TME/TEC/sql 62  
/var/spool/Tivoli 86  
./orainst 71  
./root.sh 89  
./rootpre.sh 71  
./profile 71  
@SUM 119  
\$BINDIR/TAS/RIM/SQL/scripts 76  
<PUBLIC> 182

### Numerics

32-bit ODBC TEST 209

### A

abbreviations 253  
accounting process 3  
acronyms 253  
Add monitor 161  
admin 80, 150  
AIX 4.1 233  
AIX Base Application Development Toolkit 70  
AIX root user 48  
AIX Version 4.2 63  
alert management 11  
all\_catalog 213  
AMS 2  
AMS-enabled 2  
API layer 7  
Application Management Specification (AMS) 2  
APPLX.EXE 238  
asset database 121, 123  
asset management xv, 107  
assets 120  
asterisk 143  
Asynchronous I/O 48  
auto-detect 192

Availability 234  
availability of the RDBMS server 234  
Available Targets 41

### B

backup server 25, 52  
backup the database 73  
BAROC file 132  
bibliography 245  
book value 4, 120  
bos.adt.base 70  
boundary of the TMR 64  
business applications 6

### C

C library calls 7  
c:Sybase 15  
C:\ORANT 201  
c:\Sybase 27  
c:\sybase\bin\SQL.INI 24  
calculations 115  
cell 118  
central RDBMS server 236  
change control policies 45  
character set representation 82  
character-mode terminal 87  
class definitions 137  
classes 135  
client machine 82  
Collect Software Inventory 178  
Command History 219  
command line interface 17, 66, 232  
common syntax 67  
communication error 103  
Community 91  
COMPUTER\_SYSTEM 223  
concurrent applications 234  
configuration files 90  
configuration option 13  
configuration record 233  
configuration repository 239  
connect 226  
connection 82, 89  
connection information 24  
Connection Service Entry 25  
cooperative approach 8  
CORBA objects 4, 10, 174  
cr\_tec\_db.sh 62  
criteria 238  
current database information 57  
custom scanners 238  
customization options 40

## D

data 107  
 data model 5, 175  
 database 2  
 database administration 202  
 database administrator 70  
 Database Alias 205  
 database backup 29  
 database client 232  
 Database Connection 204  
 Database Directory 178  
 database environment variables 70  
 Database Home 57  
 database home directory 57  
 Database ID 57  
 Database Listener 94  
 database monitors 163  
 database query 114  
 database schema 34, 54, 55  
 database server 47, 63, 232  
 database tables 5, 196  
 database user ID 57  
 DB\_MON\_PRO 164  
 DB\_MONITOR 162  
 db\_setup.sh 71  
 DB2 7, 231, 232  
 DB2 for AIX 231, 233  
 DB2 for MVS 233  
 DDCS connection 231  
 dedicated RDBMS server 234  
 Default rule base 135  
 degree of availability 234  
 deployment management 1  
 describe 213  
 desktop management suite 9, 173, 238  
 detector components 2  
 device capabilities 11  
 devices 11  
 DISC\_SYSTEM\_TCP\_HOSTNAME 175  
 discovery 11  
 DISPLAY 90  
 Distributed Database Connection Services (DDCS) 233  
 distributed RDBMS servers 237  
 distributed TMR servers 236  
 domain 186  
 drop existing tables 34  
 DSQUERY 168

## E

Edit monitor 161  
 enabled state 170  
 end-user problems 3  
 enterprise management 8  
 enterprise-wide inventory 8  
 environment 72

equipment 4  
 event classes 131  
 event group 140  
 events coming in 146  
 execution options 155  
 external database reference 109

## F

file name extension 122  
 file system 49, 73  
 filter 143  
 filtering 140  
 financial information 4  
 formula 118  
 free capacity of any logical drive 214  
 full life-cycle management 1  
 function 118

## G

glossary 251  
 go 226  
 graphical reports 120  
 graphical user interface 73

## H

hardware information 44  
 hardware inventory 44  
 hardware profile scan 78  
 HDW 178  
 headquarters 236  
 heterogeneous LANs 8  
 homogeneous view 8

## I

IBM NetFinity 8  
 IF function 118  
 Informix 7  
 inside the TMR 76  
 install clients remotely 69  
 Install\_product 81  
 installation prerequisites 67  
 installation progress 203  
 installation status 2  
 INSTALLED 124  
 INSTALLED\_SOFTWARE\_COMPONENT 239  
 INSTALLED\_SOFTWARE\_VERSION 123  
 Installer 71  
 Intel LANDesk 8  
 interface code 6  
 Interfaces file 18  
 interprocess communication method 233  
 INVENTORY 214  
 inventory account 103  
 inventory database 34, 61

- inventory database tree 228
- inventory information 173
- inventory management 1
- inventory scan 38
- inventory server 63
- inventory user account 76
- Inventory\_edit 80
- Inventory\_view 80
- INVENTORYDATA 37, 56, 213, 222
- INVENTORYDATA view 114
- InventoryProfile 30
- ISQL 17, 159, 232
- isql command 30
- ISQL path 168

## K

- Korn Shell 229

## L

- label of the RIM object 57
- LAN 76
- LAN management tool 175
- LAN tools 175
- LanAccessSite 174
- language 201
- LCF 236
- LCF Patch B 160
- LD\_LIBRRY\_PATH variable 87
- lic\_check 150
- lic\_job 155
- lic.baroc 131
- LICENCES\_USED 128
- License Checking Application 121
- license key 50
- license management 107
- License\_Exceeded 131, 147
- License\_Missing 131, 147
- licenses 4
- LICENSES\_ALLOWED 124
- LICENSES\_USED 124
- Lightweight Client Framework 236
- links 71
- listener.ora 96
- Log File Path 191
- logical network interface 233
- LOGICALDRIVE 214
- Lotus 1-2-3 107
- Lotus 1-2-3 worksheet 114
- Lotus Approach 108
- Isnrctl 96

## M

- macro 118
- mainframe 233
- managed resource 30

- management cycle 2
- management domain 9
- manager of managers 175
- mapping mechanism 175
- master 25, 52, 54
- master database 32
- Master Device File 19, 51
- Microsoft SMS 238
- Microsoft Systems Management Server (SMS) 8
- MINOR 146
- missing or exceeded licenses 146
- monitor arguments 168
- monitoring 140
- monitoring collection 167
- Motif 87
- Motif libraries 87
- Motif V2.1 74
- MPM providers 174
- MS SQL 7, 231
- MS SQL Server 231
- MSB object 10
- MSB processing engine 174
- Multi-Platform Management application 174
- multiple databases 234
- Multiple Services Broker (MSB) 9
- MVS system 231
- MYOWN.EXE 122

## N

- name of the logical drive 214
- named pipes 25
- nested Lotus 1-2-3 function 118
- NetBIOS 187
- NETFBASE.exe 180
- NetFinity 5.0 173
- NetFinity 5.0 Configuration program 179
- NetFinity 5.0 Server Manager 176
- Netscape Navigator Version 3.01 148
- NetView DM 238
- network 200
- network connection 24
- network drive 26
- network driver 187
- Network Driver Configuration 186
- network protocol 63
- Network Time-Out 187
- NLWNSCK 28, 54
- notification bulletin board 80

## O

- object representation 8
- online documentation 203
- ora\_sp1\_patch.sql 77
- Oracle 6, 70, 75, 231
- Oracle client 200, 204
- Oracle client installation 209

Oracle Data Query 72, 88  
 Oracle database 199, 218  
 Oracle Easy\*SQL 72, 88  
 Oracle for AIX 63  
 Oracle for AIX directory 86  
 Oracle Installer 89, 200  
 Oracle Network Manager 87, 88  
 Oracle Schema Manager 220  
 Oracle SQL Worksheet 218  
 Oracle Unix Installer and Documentation 72, 88  
 Oracle Version 7.1 63  
 ORACLE\_HOME 71, 95  
 ORACLE\_HOMENETWORKADMIN 204  
 oracle\_link directory 71  
 ORACLE\_OWNER 71  
 ORACLE\_SID 71  
 Oracle7 Client Products 202  
 ORACLE7 Server (RDBMS) 72, 88  
 ordering process 3  
 ordering system 4  
 OS/2 238  
 OS/2 machines 192  
 OS2.INI 180  
 Output Destination 152  
 outside the TMR 76

## P

password 234  
 patch installation index file 68  
 PC administration tools 174  
 PC agent code 13  
 PC user 3  
 performance 5, 234  
 Perl script 101  
 policy region 30, 37, 161, 174, 189  
 pop-up 169  
 port 5000 25  
 positive filter 143  
 prerequisites 2  
 privileged groups 72  
 privileges 221  
 Processing Engine 10  
 PROCESSOR table 117  
 processor types 117  
 PROCESSOR\_MODEL 114  
 PROCESSOR\_MODEL column 117  
 profile 161  
 profile manager 30, 37, 79, 161  
 progress indicator 50  
 protocol-independent process 89

## Q

query 25, 52, 54  
 Query Table 1 115  
 Query\_execute 80  
 Query\_view 80

QUICK CONFIGURE SQL SERVER 50

## R

RDBMS 5  
 RDBMS client 13, 199, 236  
 RDBMS host 13  
 RDBMS Interface Module (RIM) 6  
 RDBMS scalability 234  
 RDBMS server 53, 63, 81, 96  
 RDBMS server located outside the TMR 64  
 RDBMS server platform 231  
 RDBMS server-to-server links 237  
 RDBMS support 7  
 RDBMS system 107  
 RDBMS topology 234  
 RDBMS User 57  
 RDBMS Vendor 57  
 RDBMS\_DB\_Home 75  
 RDBMS\_DB\_Name 75  
 RDBMS\_Host 75  
 RDBMS\_Vendor 75  
 reference configuration 2  
 Refresh Data 117  
 Refresh Period 178, 198  
 region administrator 80  
 Relational DataBase Management System (RDBMS) 5  
 relational database system 4  
 relink the executables 89  
 remote Oracle database 82  
 remote program execution 11  
 remote RDBMS server 13  
 report 214  
 Request Handler 10  
 return code 59, 153  
 Return Data Results 42  
 RIM 232  
 RIM component 5  
 RIM host 13, 24, 29, 43, 63, 81, 89, 96, 100  
 RIM host label 103  
 RIM object 57  
 RIM\_view 80  
 roles 189  
 Roles Required 150  
 root 67, 90  
 rule base 131  
 rule engine 3  
 rules 135

## S

sa 32, 168  
 sa user 55  
 Save monitor 161  
 scalability 5, 234  
 scalability options 234  
 scalability perspective 234

- scan process 81
- scanned hardware data 81
- scanners 2
- scanning process 81, 196
- scheduler 155
- Security 234
- security perspective 234
- select 37, 56, 213
- select statement 226
- Selected Targets 41
- selection criteria 194
- senior 80
- serial operation 43
- Server ID 57, 58
- server platform 231
- Services Manager tool 21
- Set monitoring schedule 161
- Setup.exe 27, 200
- SFT 178
- share the RDBMS system 234
- shell script 66
- SID 95
- sid1 95
- signature file 122
- signature files 238
- size of the signature 122
- SmartSuite 97 108
- SMS 175
- software configuration 2
- software distribution 3
- software package 2
- SOFTWARE\_ID 124
- SOFTWARE\_VERSION\_ID 124
- source definition 140
- spool 213
- spreadsheet application 107
- SQL 5
- SQL Central 22, 35, 59, 227
- SQL connectivity 35
- SQL queries 218
- SQL script 34, 61
- SQL scripts 33
- SQL Server name 19, 51
- SQL Server status 21
- SQL upgrade scripts 76
- SQL Worksheet GUI 218
- SQL.INI file 18, 25, 32, 52
- SQL\*Net 82
- SQL\*Net Easy Configuration 204
- SQL\*Net Listener 89
- SQL\*Net TCP 89
- SQL\*Net terminology 92
- SQL\*Net V2 89
- SQL\*Net Version 2.1 89
- SQL\*PLUS 77, 210
- SQLEDIT tool 24, 53
- sqlnet.ora 96, 204

- SQLPING 32
- sqlplus 77, 98, 232
- Staging Area Path 191
- Standard Error Output 153
- Standard Output 153
- standard SQL string 8
- subscribers 38, 161
- subset 175
- successful connection 226
- super 81
- superset 175
- SW\_LICENCES 123
- swlic\_cmp.sh 128, 146
- swlic\_cre.sh 123
- swlic\_upd.sh 125
- syb\_sp1\_schema.sql 56
- Sybase 6, 51, 75, 168, 231
- Sybase 11 13
- Sybase client 13, 24, 29, 53, 109
- Sybase CPU an I/O usage 159
- Sybase database 199
- SYBASE directory 49
- Sybase disk reads and writes 159
- Sybase engine status 159
- SYBASE environment variable 49
- Sybase for AIX 47
- Sybase installation images 26
- Sybase locks usage 159
- Sybase message traffic 159
- Sybase Monitoring collection 159
- Sybase port 51
- Sybase RDBMS server 15, 160
- Sybase script 226
- Sybase session usage 159
- Sybase SQL Server Professional 15
- Sybase System 11 111
- Sybase table 110
- Sybase table space usage 159
- Sybase tables 53
- sybsetup 48
- System Administrator Password 22
- System Information 183
- System Procedure Device Size 19
- systems management 6, 8

## T

- T/EC console 121
- T/EC event 128
- tablespace 30
- TCP Port Number 51
- TCP/IP 24, 63, 92, 187, 233
- TCP/IP adapter 100
- TCP/IP connection 25, 54
- TCP/IP hostname 25, 54
- TCP/IP Protocol Adapter (V2) 89
- TCP/IP socket connection 24
- tec database 61

The Management Services Broker (MSB) 174  
 threshold 159  
 threshold values 168  
 time zones 175  
 Tivoli commands 66, 73  
 Tivoli end-user commands 67  
 Tivoli Manager for Applications 5  
 tivoli password 104  
 Tivoli terminology 66  
 tivoli user 34  
 tivoli user ID 34, 55  
 tivoli userid 59  
 Tivoli\_Admin\_Privileges group 186  
 tivoli\_ora\_admin.sql 76  
 tivoli\_ora\_schema.sql 76  
 tivoli\_syb\_admin.sql 33, 54, 62  
 tivoli\_syb\_schema.sql 33, 35, 54  
 Tivoli/Inventory 3.0 Revision A 74  
 tivoli.INVENTORYDATA 113  
 TME 10 database 73  
 TME 10 Deployment Management 1  
 TME 10 Desktop 37, 187  
 TME 10 Distributed Monitoring 3.5 159  
 TME 10 Enterprise Console 51, 107, 121, 131, 234  
 TME 10 Framework 4, 33, 34, 63  
 TME 10 Framework 3.1 54  
 TME 10 Framework 3.2 7, 160  
 TME 10 Framework Version 3.1 Revision C 63  
 TME 10 Framework Version 3.1.1 Service Pack 1 63  
 TME 10 Inventory 1, 5, 34, 47, 63, 107  
 TME 10 Inventory 3.0 35, 56, 74  
 TME 10 Inventory 3.1 35, 56, 64, 74, 231  
 TME 10 Inventory 3.1 for NT Only 29  
 TME 10 Inventory 3.1 Release Notes 76  
 TME 10 Inventory 3.2 231  
 TME 10 Inventory data model 6  
 TME 10 Inventory database 30, 34, 121, 158, 173  
 TME 10 Inventory platform 231  
 TME 10 Inventory profile 37  
 TME 10 Inventory server 24, 232  
 TME 10 Inventory Version 3.1 Revision A 63  
 TME 10 job 149  
 TME 10 LAN Access 8, 173, 238  
 TME 10 LAN Access Provider 174  
 TME 10 patch 68  
 TME 10 Software Distribution 1  
 TME 10 Software Distribution (3.2 239  
 TME 10 Software Distribution 3.1.3 238  
 TME 10 task 149  
 TME 10 task library 149  
 TME Desktop 37  
 TME object labels 236  
 TME RDBMS Access Host 29, 65  
 TME\_OBJECT\_LABEL 114  
 TMR 9, 13, 43  
 TMR connections 236  
 TMR RDBMS access host 63

TMR server 63, 69  
 TNS-based software 92  
 TNS\_ADMIN 96  
 tnsnames.ora 96, 204  
 tools 199  
 total size of any logical drive 214  
 total space used 169  
 translation layer 8  
 Transparent Network Substrate (TNS) 89  
 transport protocol 92  
 trusted host permissions 69

## U

Universal times 175  
 UNIX 231  
 UNIX flavors 238  
 UNIX systems 231  
 UNLOAD SYBASE PRODUCTS 49  
 user 80  
 user help-desk 3  
 user ID 70, 168  
 utilization 216

## W

Windows 238  
 Windows 95 30  
 Windows ISQL for 32 bit 17  
 Windows NT 30, 231, 233  
 Windows NT 3.51 30  
 Windows NT 4.0 30  
 Windows NT command 204  
 Windows NT Server 4.0 27  
 WINSOCK 54  
 WINSOCK connection 25  
 WISQL32 30, 35, 54, 56, 226  
 workgroup management tool 8  
 World Wide Web 101



---

## ITSO Redbook Evaluation

TME 10 Deployment Cookbook: Inventory and Company  
SG24-2120-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at <http://www.redbooks.com>
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to [redbook@vnet.ibm.com](mailto:redbook@vnet.ibm.com)

**Please rate your overall satisfaction** with this book using the scale:  
**(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)**

**Overall Satisfaction** \_\_\_\_\_

**Please answer the following questions:**

Was this redbook published in time for your needs? Yes\_\_\_\_ No\_\_\_\_

If no, please explain:

---

---

---

---

What other redbooks would you like to see published?

---

---

---

**Comments/Suggestions:**      **( THANK YOU FOR YOUR FEEDBACK! )**

---

---

---

---

---



This soft copy for use by IBM employees only.

Printed in U.S.A.

SG24-2120-00

