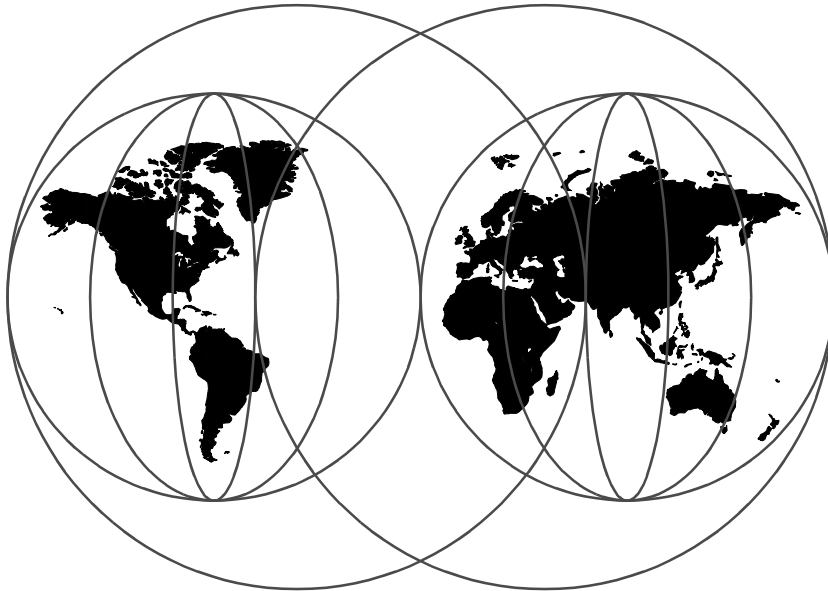


Using Databases with Tivoli Applications and RIM

Jan Marais, Hennie Pienaar, Fred Plassman



International Technical Support Organization

<http://www.redbooks.ibm.com>

SG24-5112-00



International Technical Support Organization

**Using Databases with Tivoli Applications
and RIM**

June 1998

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix C, "Special Notices" on page 177.

First Edition (June 1998)

This edition applies to Tivoli Framework, Tivoli Inventory, and Tivoli Enterprise Console, Version 3.6 pre-GA for use with the AIX 4.3.1 and Windows NT 4.0 SP3 Operating Systems.

Note

This book is based on a pre-GA version of a product and may not apply when the product becomes generally available. We recommend that you consult the product documentation or follow-on versions of this redbook for more current information.

Comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. DHHB Building 045 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

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 1998. 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

Figuresix
Tables	xv
Preface	xvii
The Team That Wrote This Redbook	xvii
Comments Welcome	xviii
Chapter 1. Introduction to RIM	1
1.1 What is RIM?	1
1.1.1 Applications That Use RIM	2
1.1.2 RIM Implementation	3
1.2 RIM Architecture	4
1.2.1 Implications of Architectural Changes	6
1.2.2 Recommended RIM Configuration	8
1.3 The RIM Object	9
1.3.1 RIM Object Methods	10
1.3.2 Adding Appropriate Roles to Tivoli Administrator	11
1.4 RIM Commands	13
1.4.1 Creating a RIM Object	13
1.4.2 Listing Database Information for a RIM Object	14
1.4.3 Changing Information for a RIM Object	14
1.4.4 Changing the Database User Password	15
1.4.5 Deleting a RIM Object	16
1.4.6 Register Resource with Tivoli Name Registry	16
1.4.7 Searching for RIM Object in Tivoli Name Registry	16
1.5 Testing the Connection between RIM Host and Database Server	17
1.5.1 RIM Test Command	17
1.5.2 RIM Trace Command	18
1.5.3 RIM Error Log	19
Chapter 2. Configuring IBM DB2 for Use with RIM	23
2.1 IBM DB2 Overview	23
2.1.1 DB2 Versions Supported	23
2.1.2 DB2 Database Objects	23
2.2 Creating the DB2 System Administrator User	26
2.2.1 Creating a SYSADM Group and User for Windows NT	26
2.2.2 Creating a SYSADM Group and User for AIX	29
2.3 Installing DB2 Common Server Version 2.1.2	33
2.3.1 Installing DB2 Common Server for NT	34
2.3.2 Starting DB2 for NT Server Automatically	38

2.3.3	Start the Security Server and Database Instance for NT	40
2.3.4	Installing DB2 Common Server for AIX	40
2.3.5	Creating Database Instance Using DB2 for AIX	44
2.4	Installing DB2 Common Server CAE Version 2.1.2	46
2.4.1	Installing DB2 CAE for NT	46
2.4.2	Installing DB2 CAE for AIX	48
2.5	Configuring DB2 Version 2.1.2 Connectivity	50
2.5.1	Configuring DB2 Server for NT	50
2.5.2	Configuring DB2 CAE for NT	51
2.5.3	Configuring DB2 Server for AIX	56
2.5.4	Configuring DB2 CAE for AIX	57
2.6	Creating a RIM Object to Access a DB2 Database	58
2.6.1	DB2 RIM Object	59
2.6.2	Creating RIM Object for NT Host	60
2.6.3	Creating a RIM Object for AIX Host	61
2.6.4	Testing the RIM Object	62
2.6.5	Determining the Database Home Directory	63
2.6.6	Determining the Instance Home Directory	65
Chapter 3. Configuring Oracle for Use with RIM		67
3.1	Oracle Overview	67
3.1.1	Oracle Versions Supported	67
3.1.2	SQL*Plus	68
3.1.3	SQL*Net	68
3.2	Installing Oracle 7.3.3 for AIX	68
3.2.1	Installing Oracle for AIX Server	68
3.2.2	Installing Oracle for AIX Client	83
3.3	Installing Oracle 7.3.3 for Windows NT	86
3.3.1	Required Files for RIM from Oracle for NT	86
3.3.2	Installing Oracle for NT Server	86
3.3.3	Installing Oracle for NT Client	91
3.4	SQL*Net Overview	92
3.4.1	Creating TNSNAMES.ORA File Entry	92
3.4.2	Oracle SQL*Net Listener	95
3.4.3	Determining Name of Current Listener	96
3.4.4	Creating LISTENER.ORA File Entry	96
3.4.5	SQL*Net Listener Log File	98
3.5	Configuring Oracle for AIX Connectivity	99
3.5.1	Configuring SQL*Net Server for AIX	99
3.5.2	Configuring SQL*Net Client for AIX	100
3.6	Configuring Oracle for NT Connectivity	102
3.6.1	Configuring SQL*Net Server for NT	102
3.6.2	Configuring SQL*Net Client for NT	103

3.7	Connecting the Oracle Client and Server for AIX	106
3.7.1	Using Oracle for AIX TNS Ping Utility	106
3.7.2	Using Oracle for AIX SQL*Plus Utility	107
3.8	Connecting Oracle Server and Client for NT	108
3.8.1	Using Oracle for NT TNS Ping Utility	108
3.8.2	Using Oracle for NT SQL*Plus Utility	109
3.9	Creating RIM Object to Access Oracle Database	111
3.9.1	Oracle RIM Object	111
3.9.2	Creating RIM Object for AIX RIM Host	112
3.9.3	Creating RIM Object for NT RIM Host	113
3.9.4	Testing RIM Object	114
3.10	Configuring a Connection between NT and AIX	116
3.10.1	Configuring SQL*Net for NT to Connect to AIX Server	116
3.10.2	Configuring SQL*Net for AIX to Connect to NT Server	118
3.10.3	Connecting Oracle AIX Client and NT Server	119
3.10.4	Creating RIM Object	120
3.10.5	Testing RIM Object	121
Chapter 4. Troubleshooting Hints and Tips		123
4.1	Using RIM Test Utility to Resolve Problems	123
4.1.1	ORA-12154 TNS:Could Not Resolve Service Name	123
4.1.2	ORA-03121 Error While Trying to Retrieve Text for Error	124
4.1.3	SQL1032N No Start Database Manager Issued	125
4.1.4	SQL1402N Unable to Authenticate User	126
4.1.5	Authorization Error - Insufficient Authorization	126
4.2	Using RIM Test Utility to Execute SQL Commands	127
4.2.1	List Help - Option ?	127
4.2.2	RIM Retrieve Rows - Option g	128
4.2.3	RIM Execute SQL Statements - Option e	131
4.2.4	RIM Insert Rows - Option i	131
4.2.5	RIM Delete Rows - Option d	133
4.2.6	RIM Update Rows - Option u	134
4.2.7	RIM Commit - Option c	135
4.2.8	RIM Rollback - Option r	135
4.3	Using RIM Trace to Resolve Problem	135
4.3.1	The RDBMS Server Call Has Failed	136
4.4	Oracle Performance Tips	140
4.4.1	Oracle Optimizer	140
4.4.2	Determining Oracle Database Free Space	140
4.4.3	Sizing Database Objects	141
4.4.4	Database Backup	141

Chapter 5. How to Use the Tivoli Framework Query Facility	143
5.1 Tivoli Framework Query Facility Overview	143
5.2 Creating a Query Administrator	143
5.3 Creating a Query Using GUI	145
5.3.1 Section 1 - Query Name, Repository, View/Table Name	147
5.3.2 Section 2 - Columns	149
5.3.3 Section 3 - Where Clause	151
5.3.4 Section 4 - Additional Clauses	153
5.4 Creating a Query Using the Command Line	154
5.5 Creating Views for Non-Tivoli Application	156
5.5.1 Creating Catalog Views for Non-Tivoli Application for DB2	156
5.5.2 How to Execute the SQL Script for DB2	158
5.5.3 RIM Supported DB2 Data Types	158
5.5.4 Creating Catalog Views for Non-Tivoli Applications for Oracle	159
Appendix A. Other Databases Supported by RIM	161
A.1 Sybase Overview	161
A.1.1 Sybase RIM Object	161
A.1.2 Sybase Versions Supported	162
A.1.3 Using ISQL	162
A.1.4 Sybase Interfaces File (UNIX)	162
A.1.5 Sybase Interfaces File (Windows NT)	163
A.1.6 Asynchronous I/O Configuration	165
A.1.7 Sybase Installation Tips	165
A.2 Microsoft SQL Server Overview	165
A.2.1 MS SQL Server RIM Object	165
A.2.2 MS SQL Server Versions Supported	166
A.2.3 MS SQL Server Installation Tips	166
Appendix B. Useful DB2 Commands	167
B.1 How to Find Help	167
B.1.1 Help on Syntax or Commands	167
B.1.2 Help on SQL Statements	169
B.1.3 Help on Error Codes	169
B.1.4 On-line Manuals	170
B.2 Finding the Database Names Known to DB2	170
B.3 Finding Other Servers or Instances Known to DB2	171
B.4 Connecting to a DB2 Database	172
B.5 Listing the Table Names Created in a DB2 Database	173
B.6 Executing SQL Statements	173
B.7 Listing the Column Names in a DB2 Database Table	174
B.8 Listing Applications Connected to DB2 Database	174
B.9 Forcing Applications to Disconnect from a DB2 Database	175

B.10 Deleting a DB2 Database	175
B.11 Listing All DB2 Instances on an Machine	176
B.12 Switching DB2 Instances from the Command Line.	176
B.13 Starting and Stopping an DB2 Instance	176
Appendix C. Special Notices	177
Appendix D. Related Publications.	181
D.1 International Technical Support Organization Publications	181
D.2 Redbooks on CD-ROMs.	181
D.3 Other Publications	182
How to Get ITSO Redbooks	183
How IBM Employees Can Get ITSO Redbooks.	183
How Customers Can Get ITSO Redbooks.	184
IBM Redbook Order Form	185
Glossary	187
List of Abbreviations.	189
Index	193
ITSO Redbook Evaluation.	199

Figures

1. RIM Object Inheritance Hierarchy	3
2. RIM 3.1 Architecture	4
3. RIM 3.2 Architecture for Sybase, IBM DB2, and MS SQL Server	5
4. RIM 3.2 Architecture for Oracle	6
5. Object Dispatcher Status Output	7
6. RIM Trace Output	8
7. Edit Administrators TMR Roles	12
8. Set TMR Roles.	12
9. List Information for RIM Object	14
10. Database User Password Does Not Match RIM Object Password	15
11. List All RIM Objects in TMR.	17
12. Display RIM Test Options	18
13. Set RIM Trace Options	19
14. Set Method Environment for TMR Server from TMR Server	20
15. Set Method Environment for the RIM Host from the TMR Server	21
16. DB2 Objects on AIX.	25
17. DB2 Objects on NT	26
18. Windows NT User Manager - New User	27
19. Windows NT New User	28
20. Windows NT New Local Group	28
21. Windows NT Add Users and Groups.	29
22. SMIT Security & Users.	30
23. SMIT Add a Group for DB2	31
24. SMIT Users	32
25. SMIT Add User for DB2.	33
26. DB2 Installation Error.	34
27. DB2 Installation Selection	35
28. DB2 Licensing	36
29. Setup Options	36
30. DB2 Server Components.	37
31. Windows NT Control Panel Services - DB2.	38
32. Database Instance and Security Server Services	39
33. Database Instance and Security Server Status	39
34. SMIT Software Installation and Maintenance	41
35. SMIT Install and Update from Latest Available Software.	42
36. SMIT Software to Install List for DB2.	44
37. CAE Setup Options - Installation Type	47
38. DB2 Client Components	47
39. SMIT Software to Install List	49
40. Services File - DB2 Server on NT	51

41. Services File - DB2 Client on NT	52
42. Hosts File - DB2 Client on NT	52
43. DB2 Client Setup	53
44. DB2 Node Settings	53
45. DB2 Client Setup - Databases	54
46. DB2 New Database	55
47. IBM DB2 ODBC Driver - Test Connection	56
48. Extract from /etc/service File	57
49. Create RIM Object for NT Host	61
50. Create RIM Object for AIX Host	61
51. RIM Test - DB2 for NT	62
52. RIM Test - DB2 for AIX	62
53. Database Home Directory - DB2 for NT Server	63
54. Database Home Directory - DB2 for NT Client	64
55. Database Home Directory - DB2 for AIX Server	64
56. Database Home Directory - DB2 for AIX Client	65
57. Display Instance Owner User Definition	66
58. Verify bos.adt.base Installed	69
59. SMIT Add Group for Oracle	69
60. SMIT Add User for Oracle	70
61. SMIT Create Oracle File System	71
62. Display Mounted File Systems	72
63. Oracle Profile Script (db_setup.sh)	73
64. SMIT Add CDROM File System	74
65. Oracle Installation Menu	76
66. Oracle Environment Variables - Create DB Object	77
67. Oracle Installation Defaults	78
68. Software Asset Manager - Database Server	79
69. Oracle Table File	81
70. Oracle Database Start Output	82
71. Oracle Database Shutdown Output	82
72. System Shutdown Script	83
73. Oracle Client Installation Options	83
74. Oracle Environment Variables - Do Not Create DB Object	84
75. Software Asset Manager - Database Client	85
76. Oracle Asset Manager Window for Windows NT	88
77. TNSNAMES.ORA File Format	93
78. Oracle SQL*Net Listener Status	94
79. TNSNAMES.ORA File - RIM Host	95
80. Show Current Oracle SQL*Net Listener	96
81. LISTENER.ORA File Format	97
82. TCP Entry in LISTENER.ORA File	97
83. IPC Entry in LISTENER.ORA File	98

84. SQL*Net Listener Log File	98
85. Oracle for AIX LISTENER.ORA File - Database Server	99
86. Oracle Start Listener Output	100
87. Oracle for AIX SQLNET.ORA File - Database Client.	101
88. Oracle for AIX TNSNAMES.ORA - Database Client	101
89. Windows NT Control Panel Services - Oracle	103
90. Oracle for NT SQLNET.ORA File	103
91. SQL*Net Easy Configuration - Add Database Alias.	104
92. Choose TCP/IP Host Name and Database Instance	105
93. Confirm Adding Database Alias	105
94. Oracle for NT TNSNAMES.ORA - Database Client.	106
95. Oracle TNS Ping Output - TNS Listener Not Started.	107
96. Oracle TNS Ping Output - TNS Listener Started	107
97. Using SQL*Plus on AIX.	108
98. TNS Ping Output - Oracle for NT Client.	109
99. SQL*Plus Log On.	110
100. SQL*Plus - Oracle for NT Client	111
101. Create Oracle RIM Object for AIX RIM Host	112
102. Display Oracle RIM Object for AIX RIM Host	113
103. Create Oracle RIM Object for NT RIM Host	113
104. Display RIM Object - NT RIM Host	114
105. Oracle for AIX RIM Test Connect Output	115
106. Oracle for NT RIM Test Connect Output	115
107. SQL Net Easy Configuration - Add Database Alias	116
108. Choose Database Alias for AIX Server on NT Client	117
109. Choose Host Name & Database Instance for AIX Server on NT Client	117
110. Confirm Adding Database Alias for AIX server on NT Client	118
111. TNSNAMES.ORA Entry for NT Server on AIX Clients	119
112. TNS Ping From AIX Client To NT Server	120
113. SQL*Plus From AIX Client To NT Server	120
114. Create Oracle RIM Object for AIX RIM Host.	121
115. Display RIM Object for AIX RIM Host	121
116. RIM Test Connect From AIX RIM Host To NT Server.	122
117. TNSNAMES.ORA Default File Permission Setting	123
118. ORA-12154 TNS: Could Not Resolve Service Name	124
119. Server ID Without Domain.	124
120. ORA-03121 Error while Trying to Retrieve Text for Error	125
121. SQL1032N No Start Database Manager Command Was Issued	125
122. SQL1402N Unable to Authenticate User.	126
123. Authorization Error - Insufficient Authorization	127
124. RIM Test Options	128
125. Select DEPTNAME from DEPARTMENT	129
126. Enter Select Statement - Do Not Use Editor	130

127.Enter Select Statement - Use Editor	131
128.Insert Single Row of Data Into Table - Do Not Use Editor	132
129.Insert Single Row of Data Into Table - Use Editor.	133
130.Delete Row	134
131.Update Rows - Do Not Use Editor.	134
132.Update Rows - Use Editor.	135
133.Syntax of <code>wrimtrace</code> Command	136
134.RIM Trace Status	136
135.The RDBMS Server Call Has Failed	136
136.Change RIM Trace Level	137
137.Windows NT Task Manager	138
138.Define RIM Trace File	138
139.SQL1013N Database Alias or Name Could Not Be Found.	139
140.RIM Trace File Output.	139
141.Oracle Free Space Query Output	141
142.Policy Region: Create Query Library.	144
143.Create Query Library: Name	144
144.SQL Statement for Query 1	145
145.Policy Region: Query Library.	145
146.Query Library: Create a Query	146
147.Create Query - All Sections.	147
148.Create Query: Section 1	148
149.Create Query: Table/View Name Dialog	149
150.Create Query: Section 2	149
151.Create Query: Table Description.	150
152.Create Query: Section 3	151
153.Create Query: Available Columns	152
154.Create Query: Column Values List	153
155.Create Query: Section 4	154
156.Run Query - Results	154
157.SQL Statement for Query 2.	156
158.Create Query Command for Query 2	156
159.Create Views for DB2 Database	157
160.Create QUERY_VIEWS for Oracle	159
161.Create USER_TAB_COLUMNS for Oracle.	160
162.Create QUERY_DOMAINS for Oracle	160
163.Sybase Interfaces File (UNIX).	163
164.Sybase SQL.INI File (Windows NT)	164
165.DB2 Command Help	168
166.DB2 List Applications Help Syntax	169
167.DB2 Error Code Help	169
168.DB2 List Database Directory on DB2 Server	170
169.DB2 List Database Directory on DB2 Client	171

170.DB2 List Node Directory on DB2 Client	172
171.DB2 Connect on DB2 Client	172
172.DB2 List Tables	173
173.DB2 List Column Names for Database Table	174
174.DB2 List Applications	175
175.DB2 List Instances	176

Tables

1. RIM Executable Modules	3
2. RIM Attributes	9
3. RIM Methods	10
4. DB2 RIM Attributes	59
5. Oracle Supported Versions	67
6. Oracle RIM Attributes.	111
7. DB2 Data Types Supported by RIM.	158
8. Sybase RIM Attributes	161
9. MS SQL Server RIM Attributes	165

Preface

The Tivoli RDBMS Interface Module (RIM) is the interface that allows Tivoli applications to support different RDBMS products. Without this support, each Tivoli application would be responsible for handling the differences that exist among the RDBMS-support products. The RIM provides the Tivoli applications a common application programming interface to these database products.

This redbook documents the information necessary to understand, support, and troubleshoot RIM with various supported relational database products. Hints and tips are included on how to implement RIM with Oracle and DB2 database products and with the Tivoli Inventory and Tivoli Enterprise Console applications. Tivoli 3.6 pre-GA products are used in this redbook.

This redbook discusses the purpose and architecture of RIM, describes how RIM objects are created and used by application programs, gives guidance on troubleshooting common RIM problems, and discusses the differences and peculiarities of different RDBMSs that RIM supports.

This redbook was written for the technical professional who must use the RIM and supported RDBMS products with various Tivoli applications.

Readers should be familiar with the Tivoli Framework, the Tivoli Inventory and Tivoli Enterprise Console applications, and with RDBMS products. If the reader is not familiar with these topics, the IBM Redbooks provide an excellent way to learn about these latest systems management solutions. To learn more about available IBM Redbooks, visit the IBM Redbook home page on the Internet at URL <http://www.redbooks.ibm.com>. You can order Redbooks or Redbooks Collection CDs directly through this site.

We hope that you find the information presented here informative and helpful. We look forward to receiving any comments or suggestions you have for this or future redbooks.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Austin Center.

Jan Marais is an I/T Specialist working in Product Support Services at IBM South Africa. He also teaches IBM classes on DB2 for Common Server. He holds a Electronic Engineering degree from the University of Pretoria.

Henni Pienaar is a Education Specialist working at the Education and Training Centre in IBM South-Africa. He teaches AIX, Netview/6000, TCP/IP, Korn shell programming courses at this center. His areas of expertise include AIX, TCP/IP and network management.

Fred Plassman is a Senior ITSO Specialist working as a project leader at the ITSO Austin Center. He holds a degree in Applied Mathematics from the University of Missouri at Rolla. His areas of expertise include AIX, Windows NT, network management, and software distribution. He has written extensively and taught classes worldwide on all areas of Deployment Management.

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

Bart Jacob
IBM International Technical Support Organization, Austin Center

Chip Young
Tivoli Systems

Niloufar Lamei
Tivoli Systems

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 199 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

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@us.ibm.com

Chapter 1. Introduction to RIM

This chapter explains the concepts and architecture of the Tivoli RDBMS Interface Module (RIM).

1.1 What is RIM?

RIM (RDBMS Interface Module) is the Tivoli Management Framework service that allows Tivoli applications to store and retrieve data from a variety of popular relational database products. RIM allows these Tivoli applications that collect or generate large amounts of data to store that data in external databases. The goal of RIM is to allow the applications to have a common set of APIs to get data from and store data in various database products. RIM works to hide the differences between the various database products.

The current RIM supported relational database products are:

- Oracle 7.X (Oracle 7.3.3 is the latest supported version)
- Sybase 10 and 11 (Sybase 11.0.3 is latest supported version)
- IBM Database 2 (DB2) Version 2.1.2
- Microsoft SQL Server 6.5

Note

When this redbook was written (April-May 1998), the following database product versions were not supported by RIM:

- Oracle8
- IBM DB2 Universal Database Version 5

RIM consists of three distinct parts:

1. An application program interface (API) written in C that client programs use to request RIM services. This API is database-independent. That is, applications do not have to know which specific database product they are using.
2. The RIM object, which is stored in the Tivoli object database. The RIM object defines the attributes necessary for accessing the database and the methods supported by the API.
3. The RIM daemon server, which is launched by the oserv the first time that a program calls the RIM connect method. This process is what actually

talks to the relational database, executing requests on behalf of a client program and returning the results from the database back to the client.

Applications create their own RIM objects at application installation time. The machine where the RIM server process is running is known as the RIM host. The RIM host can either be the TMR server or any managed node within the TMR. The database server machine does not need to be a managed node, but does need to be able to communicate with the RIM host machine.

The TMR server, RIM host, and database server can run on the same machine. However, this is not recommended except for testing or prototype environments. The recommended configuration is to have the TMR server, RIM host, and the database server all on different machines.

A single database server can be used by multiple RIM hosts in different TMR's. This feature allows data from multiple TMRs to be stored in a single, centralized database.

1.1.1 Applications That Use RIM

The Tivoli applications that store and use large amounts of data are being migrated to use external databases. Two application currently use RIM, but many more applications are to come. The current applications that use RIM are:

1. Tivoli Inventory - Views and records hardware and software installed on remote systems.
2. Tivoli Enterprise Console (TEC) - Collects event information from a wide variety of resources; correlates the information to determine root-cause problems; and performs automated responses.

Applications that may move to RIM are:

1. Tivoli Software Distribution - Automates software distribution, installation, and updates across a network. Tivoli Software Distribution would write information to the database with every file package distribution and use the same configuration repository as Tivoli Inventory.
2. Tivoli User Administration - Performs user, group, and host management. The current number of attributes for a user managed with Tivoli User Administration is over 100.
3. Tivoli Distributed Monitoring - Monitors distribute system resources and services, pro-actively detecting and automatically correcting potential problem situations, while escalating critical situations to the Tivoli Enterprise Console.

4. Tivoli NetView - Enables users to discover TCP/IP networks, display network topologies, correlate and manage events and SNMP traps, monitor network health, and gather performance data.

The exact products and expected delivery dates for Tivoli applications that may move to RIM are not known at this time.

1.1.2 RIM Implementation

RIM is implemented as a Tivoli daemon server process. Each application creates its own RIM object and uses that object to access its database. The RIM object can only talk to one logical database. Each support RDMS (Oracle, Sybase, DB2, MS SQL) has its own RIM class object, as shown in Figure 1.

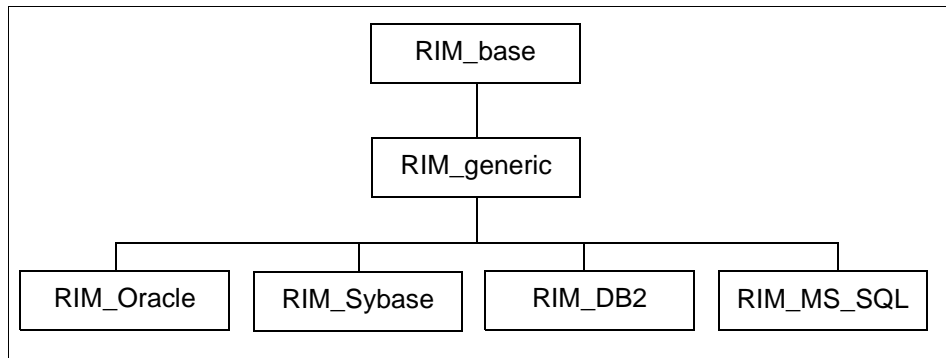


Figure 1. RIM Object Inheritance Hierarchy

Each object instance executes its own RIM server program, which is named for its object. The RIM executable modules are named as shown in Table 1 on page 3.

Table 1. RIM Executable Modules

RDBMS	Object Name	Executable Name
Oracle	RIM_Oracle	RIM_Oracle_prog
Sybase	RIM_Sybase	RIM_Sybase_prog
MS SQL Server	RIM_MS_SQL	RIM_MS_SQL_prog
DB2	RIM_DB2	RIM_DB2_prog

The RIM executable modules are located in \$BINDIR/TAS/RIM.

1.2 RIM Architecture

The original RIM architecture had all communication going through the `oserv` as shown in Figure 2 on page 4. The client program would be linked to the RIM API and all method calls would be passed from the `oserv` to the RIM server, which is the standard architecture for Tivoli daemon servers. This original architecture had two significant problems:

1. The amount of data passed either from the client (such as a `RIM_insert_rows` that has hundreds or even thousands of rows) or from the server (such as a `RIM_retrieve_rows` that returns thousands of rows) can be very large. Parsing all of this data through the `oserv` not only slows down the reading and writing of data to and from the database, but potentially slows down all TMR operations that pass through the `oserv`.
2. Oracle has considerations on certain platforms (HP-UX 10, AIX 4.1.5 and above, Sun UltraSparc) because the Oracle connect function needs to be executed in the main thread. However, in the Tivoli daemon server architecture, this function is executed in a child thread, causing all attempts to connect to the database to fail.

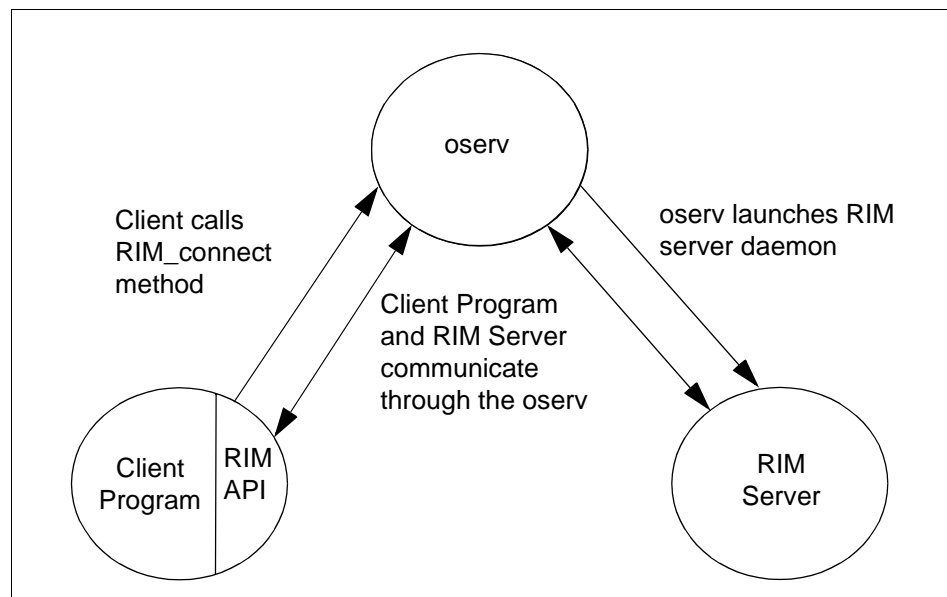


Figure 2. RIM 3.1 Architecture

The first problem is solved by opening an IOM channel between the client program and the RIM server. This RIM architecture is shown in Figure 3 on

page 5. All requests subsequent to the RIM_connect call are sent by the client to the server through the IOM channel and all results are returned from the server to the client through the IOM channel. Setting up and using the IOM channel is entirely transparent to the client and to the oserv.

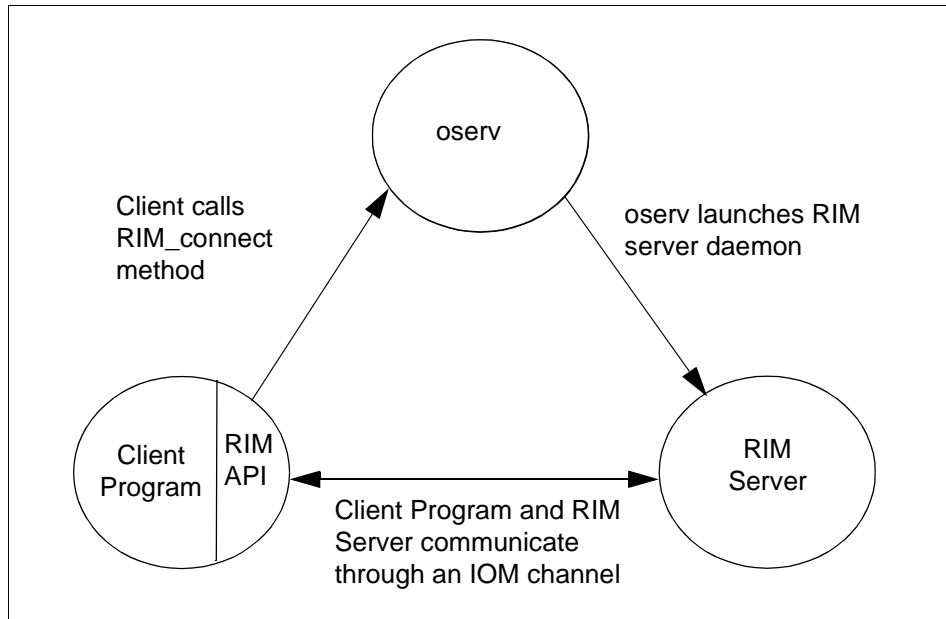


Figure 3. RIM 3.2 Architecture for Sybase, IBM DB2, and MS SQL Server

Solving the second problem - the fact that the Oracle connect function must be done in a main thread - is solved by using an agent process. This RIM architecture is shown in Figure 4 on page 6.

The RIM program creates a new Oracle agent process every time a client program executes the RIM_connect method. The lifetime of the agent is the lifetime of the connection. In other words, the agent program exits when it receives a RIM_release request from the client. The IOM channel is opened between the client and the agent, not between the client and the server.

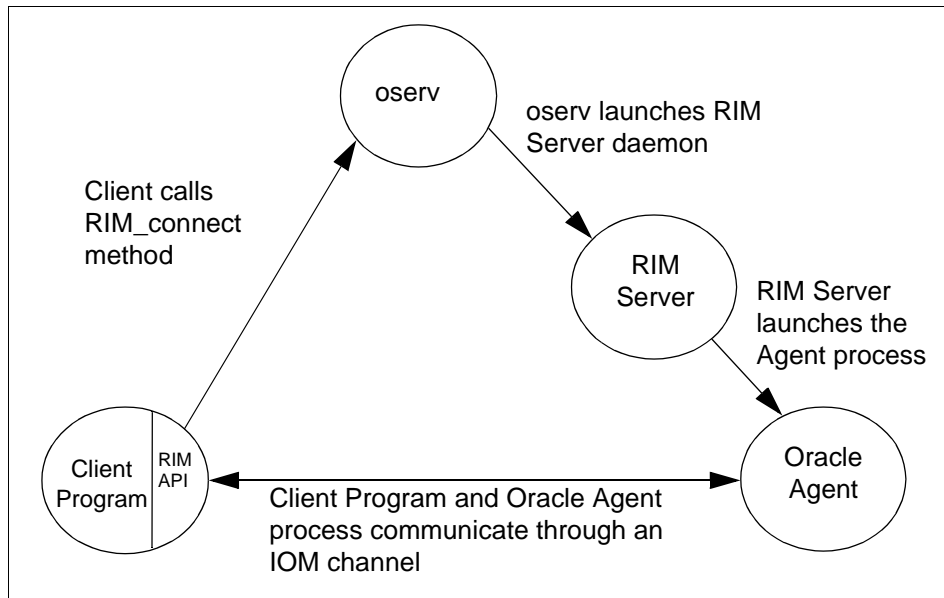


Figure 4. RIM 3.2 Architecture for Oracle

This architecture is not used on Windows NT since Oracle does not exhibit the threading issues that occur on UNIX platforms.

1.2.1 Implications of Architectural Changes

The positive impact of these changes is that RIM runs faster, does not slow down the operation of the oserv, and supports Oracle on all Tier 1 platforms. However, these changes do impact two of the most basic services offered by the platform. First, since all of the RIM method calls and returned results are passing through an IOM channel, they are no longer visible to the oserv. Since they are no longer visible to the oserv, they no longer show up in the output from the odstat and wtrac utilities. The only RIM method call that will show up in odstat is RIM_iom_session (which is what is called by RIM_connect), as shown in Figure 5 on page 7.

```

4170 O+           done    15     0 15:43:45
      0.0.0 get_name_registry
4171 O+hdoq       done   115     0 15:43:45
      1140726473.1.26 lookup
4172 O+hdoq       done    18     0 15:43:45
      1140726473.1.75#TMF_Message::Catalog# get_message_catalog
4173 O+hdoq       done   109     0 15:43:45
      1140726473.1.26 lookup
4174 O            done   294     0 15:43:45
      1140726473.9.139#RIM::RDEMS_Interface# _get_rim_info
4175 O            done    40     0 15:43:45
      1140726473.9.139#RIM::RDEMS_Interface# get_type_name
4176 M           9-1132 done    40     0 15:43:46
      1140726473.1.448#TMF_SysAdmin::InstanceManager# _get_label
4177 O+hdoq       done   653     0 15:43:46
1140726473.1.26 object_get_all
4178 O            done    18     0 15:43:46
      1140726473.9.139#RIM::RDEMS_Interface# RIM_icm_session
4179 O+           done    15     0 15:44:32
      0.0.0 get_oserv

```

Figure 5. Object Dispatcher Status Output

The wrimtrace program is provided as a substitute for odstat and wtrace.

wrimtrace <RIM_object_label> <trace_level>

Valid trace_level values are ERROR, INFORMATION, and TRACE_OFF.

```

00000235 [Fri Apr 10 13:28:32 1998] Trace Message -
      Connection ID:: Connecting to IOM Channel
00000235 [Fri Apr 10 13:28:35 1998] Trace Message -
      Connection ID:: Beginning IOM Loop
00000235 [Fri Apr 10 13:28:51 1998] Trace Message -
      Connection ID::

IOM Command: RELEASE
row_param: <NULL>
rows: <NULL>
number1: 0
number2: 0
string1:
string2:

00000235 [Fri Apr 10 13:28:51 1998] Trace Message -
      Connection ID::
REPLY IOM Command : RELEASE Result : Success
rows: <NULL>

00000235 [Fri Apr 10 13:28:51 1998] Trace Message -
      Connection ID::

Ending IOM Loop

```

Figure 6. RIM Trace Output

For more information about `wrimtrace`, see “RIM Trace Command” on page 18.

The second change is that `idlcalls` no longer work with RIM Oracle and are no longer focally supported in RIM. The reason that `idlcalls` will not work in RIM Oracle is because the `idcall` goes to the RIM server process, which does not have a connection to the database, instead of the agent process, which does.

The `wrimtest` program is provided as a substitute for making `idlcalls`.

```
wrimtest <-l RIM_object_label>
```

for example, the default `RIM_object_label` for Inventory object is `inventory` and for Tivoli Enterprise Console is `tec`.

For more information about `wrimtest`, see “RIM Test Command” on page 17.

1.2.2 Recommended RIM Configuration

There are four configurations for RIM:

1. TMR server, RIM host, and database server all on the same machine.

2. TMR server and RIM host are on the same machine. The database server is on another machine.
3. TMR server is on one machine, RIM host and database server are on another machine.
4. TMR server, RIM host, and database server are all on separate machines.

Configuration 4 is the recommended configuration because it most effectively balances the processing load across the TMR. This is because the performance profile of each component is different.

The TMR server will consume processor cycles and network bandwidth since its primary jobs are to communicate with managed nodes and to launch new application processes. Application processes will, in turn, use up more processor cycles and memory. The database server will consume memory, secondary storage, and I/O bandwidth. Tuning a database server requires giving it as much memory as possible and optimizing disk utilization and I/O. The RIM host will require network bandwidth, since it must pass data to and from client applications and the database server, and memory, particularly when reading or writing large number of rows.

It may appear to be optimal to put the database server and the RIM host on the same machine, but this configuration makes it difficult to tune the database server since not only is the RIM server process running, but also an oserv process (since the RIM hosts must be a managed node). This configuration is particularly sub-optimal when RIM Oracle is being used since a new process is launched every time a connection is made to the database.

1.3 The RIM Object

The Tivoli database can contain multiple RIM objects, but there will be at least one RIM object for each application that uses RIM. By convention, the RIM object label will be the name of the application, such as "tec" or "inventory". The RIM object contains the attributes shown in Table 2 on page 9.

Table 2. RIM Attributes

Attribute	Description
Database Vendor	The database vendor. The value of this attribute will be Oracle, Sybase, DB2, or MS_SQL.
RIM Host	The name of the machine where the RIM server runs.
Database ID	The name of the database.

Attribute	Description
Database Home	The directory on the RIM host where the database software is installed.
Server ID	Database server name.
User Name	ID of the user who is authorized to access the database.
Instance Home	(DB2 only) Directory on the RIM host where DB2 instance was created.

The exact values that you specify for Database ID, Database ID, Server ID, and Instance Home are dependent on the specific database vendor product requirements.

The RIM performs the following actions using the values provided.

1. The client application looks up the object ID for the RIM host.
2. The RIM host is contacted and given an action.
3. The RIM host uses the directory value for database home to locate the database configuration file. For example, the interfaces file for Sybase, or the tnsnames.ora file for Oracle.
4. The RIM host looks up how to contact the database server matching the name of the server ID to an entry in the file.
5. The RIM host contacts the database server and performs the action on the database specified in the database ID field.
6. Once the request is completed by the database server, the RIM host passes the information back to the application.

1.3.1 RIM Object Methods

The RIM object supports the methods shown in Table 3 on page 10.

Table 3. RIM Methods

Method	Description
RIM_connect	Connect to the database. In RIM Oracle, start a new agent process.
RIM_release	Release the connection to the database. In RIM Oracle, terminates the agent.
RIM_retrieve_rows	Retrieve rows from the database.
RIM_insert_rows	Insert new rows into the database.

Method	Description
RIM_update_rows	Update existing rows in the database.
RIM_delete_rows	Delete rows from the database.
RIM_commit	Commit an outstanding transaction - makes all changes to database permanent.
RIM_rollback	Rollback an outstanding transaction - undo all effects of database changes since last commit.
RIM_database	Deprecated method. Changes database ID for Oracle and Sybase.
RIM_create_table*	Creates a new table in the database. Supports the creation of primary keys, foreign keys, and column constraints.
RIM_delete_table*	Deletes an existing table from the database.
RIM_describe_table*	Returns a sequence of rows in which each row describes the meta-data information for each column.
RIM_retrieve_rows2*	Improved version of RIM_retrieve_rows that separates input parameters from output parameters.
RIM_update_rows2*	Improved version of RIM_update_rows.

The methods shown with an asterisk (*) are methods that are new TMF 3.2 functions.

1.3.2 Adding Appropriate Roles to Tivoli Administrator

Before creating a RIM object, a Tivoli TMR administrator must exist with the required authorization granted to create and maintain the RIM object. By default, a TMR root administrator does not have the authority required to create and administer RIM objects or queries. This authorization is required regardless of the operating system.

We followed these steps to grant the authorization:

1. Log on as a TMR administrator with authority to change administrator roles.
2. Select Administrators from the TME desktop.
3. Right click on the administrator who will manage the RIM objects.
4. Select TMR Roles from the selection list as shown in Figure 7 on page 12.

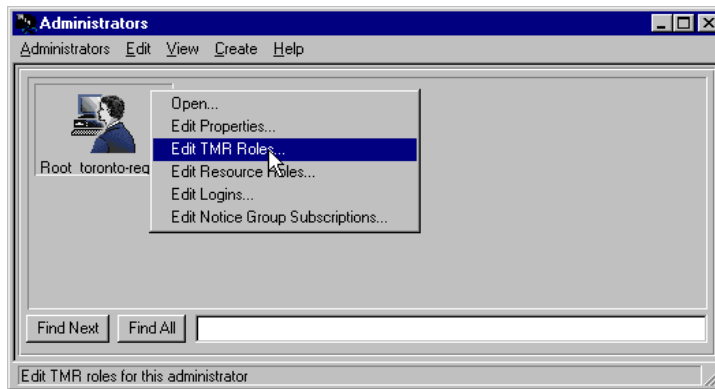


Figure 7. Edit Administrators TMR Roles

5. Select the following from the Available Resources as shown in Figure 8 on page 12 and click on the arrow button pointing left.

- Query_view
- Query_execute
- Query_edit
- RIM_view
- RIM_update

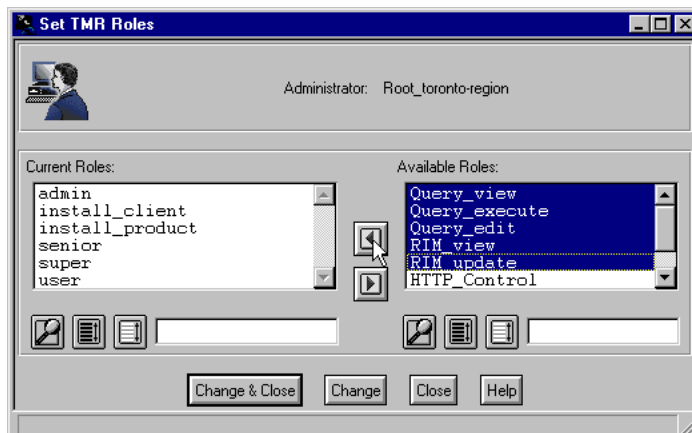


Figure 8. Set TMR Roles

6. Click on the Change & Close button.

7. Close the Tivoli desktop and start the desktop again to make the changes effective.

1.4 RIM Commands

The RIM commands are:

- `wcrtrim` - Creates a RIM object.
- `wgetrim` - Lists information about a RIM object.
- `wsetrim` - Changes the properties of a RIM object. To change the RIM host or the vendor, you must delete the object using the `wdel` command and re-create the RIM object with the `wcrtrim` command.
- `wsetrimpw` - Sets the database password for a RIM object.
- `wrimtest` - Verifies that the RIM host can connect to database server.
- `wrimtrace` - Sets the RIM trace level.

Miscellaneous Framework commands are:

- `wdel` - Deletes an object from the Tivoli database.
- `wregister` - Registers a resource with the name registry.
- `wlookup` - Searches for a resource instance in the Tivoli name registry.

The RIM and miscellaneous Framework commands are useful commands during the initial installation process and later for troubleshooting.

1.4.1 Creating a RIM Object

The command line program to create a new RIM object is `wcrtrim`.

```
Usage: wcrtrim [-i] [-v vendor] [-h host_name | -o host_oid] [-d database]
[-u user] [-H rdbms_home] [-s server_id] [-I instance_home] rim_name
```

Each of the arguments corresponds to an attribute in the RIM object. The password will be prompted unless the `-i` option is used, in which case the password is read from stdin (or a redirected from a file). The `-I` option is only valid when the Vendor is DB2.

There is a prototype RIM object in the Tivoli object database for each vendor. In other words, there is a `RIM_Oracle` prototype object, a `RIM_Sybase` prototype object, a `RIM_DB2` prototype object, and a `RIM_MS_SQL` prototype object. This prototype object is cloned to create a new RIM object when `wcrtrim` is executed and its attributes receive the values specified in the arguments to `wcrtrim`.

You can create a RIM object for the Oracle Starter Database, ORCL, created during the Oracle installation process.

```
wcrtrim -v Oracle -h polar -d ORCL -u sys -H C:\ORANT orange
```

where polar is the RIM host and C:\ORANT is the Oracle home directory (default directory for the Oracle installation on Windows NT).

1.4.2 Listing Database Information for a RIM Object

The command to list the database information for a RIM object is `wgetrim`.

```
Usage: wgetrim rim_name
```

After you create a RIM object for the Oracle Starter Database, ORCL, you can list the current attributes of the RIM object. See Figure 9 on page 14.

```
[root@panda]/> wgetrim orange

RIM Host:      polar
RDBMS User:    sys
RDBMS Vendor:  Oracle
Database ID:   ORCL
Database Home: C:\ORANT
Server ID:
Instance Home:

[root@panda]/>
```

Figure 9. List Information for RIM Object

Since the RDBMS server and the RIM host are on the same machine, the Server ID can be blank. Actually, if the RDBMS server and RIM host are on the same machine, the Server ID is not used and the contents could be specified incorrectly and the RIM would still connect to the RDBMS server.

1.4.3 Changing Information for a RIM Object

The command to change some, but not all, the information stored in a RIM object is `wsetrim`.

```
Usage: wsetrim [-n new_name] [-d database] [-u user] [-H rdbms_home] [-s
server_id] [-I instance_home] rim_name
```

You can change the RIM label from orange to purple.

```
wsetrim -n purple orange
```

You can change the database ID, database user, database home, database server ID, and instance home. To change the RIM host or vendor for a RIM object, you must delete the RIM object using the `wdel` command and recreate the RIM object using the `wcrtrim` command.

1.4.4 Changing the Database User Password

If you can change the database password for a database user associated with a RIM object, you must change the password for that user stored in the RIM object. The command to change the database user password in a RIM object is `wsetrimpw`.

```
Usage: wsetrimpw rim_name [old-password new-password]
```

The command prompts you for the passwords unless you specify the `old-password` and `new-password` arguments.

If you change the password for the Oracle database user `sys` from `change_on_install` to `armadillo`, you must change the password for `sys` in the RIM object.

```
wsetrimpw orange change_on_install armadillo
```

If the password for the database user in the RIM object does not match the database password for that user, you get the ORA-01017 error message shown in Figure 10 on page 15 when you attempt a `wrimtest`.

```
[root@panda]/> wrimtest -l orange

Resource Type : RIM
Resource Label : orange
Host Name : polar
User Name : sys
Vendor : Oracle
Database : ORCL
Database Home : C:\ORANT
Server ID :
Instance Home :
Opening Regular Session...
    Could not connect to RDBMS server to access database ORCL
The RDBMS server call's return code was: 1017
The RDBMS server call's error was:
    ORA-01017: invalid username/password; logon denied

[root@panda]/>
```

Figure 10. Database User Password Does Not Match RIM Object Password

The default database user for the Tivoli Inventory database is `tivoli` with password of `tivoli` for Oracle and Sybase. For DB2, you define the database user that owns the database instance.

1.4.5 Deleting a RIM Object

The command to delete a RIM object from the Tivoli database is `wdel`.

```
Usage: wdel [-I] label ...
```

where `-I` ignores all failed sub-operations, allowing the command to continue and `label` is one or more objects to be deleted.

You can delete the RIM object created for the Oracle Strater Database you created earlier.

```
wdel @RIM:orange
```

where `@RIM` is the object type and `orange` is the RIM object label.

1.4.6 Register Resource with Tivoli Name Registry.

The command to register a resource with the Tivoli name registry is `wregister`.

```
Usage: wregister -i [-f flags] -r resource_type
```

where `-i` initializes the resource cache with a new resource type (`-r` option) and `-f` specifies that the resource created should be non-exchangeable. Non-exchangeable resource types cannot be updated between connected Tivoli Management Regions (TMRs).

```
Usage: wregister -u [-r resource_type] name
```

where `-u` removes a resource from a resource type and `-r` specifies the resource type.

You can delete a RIM object for the Oracle Starter Database you created earlier:

```
wregister -ur RIM orange
```

where `RIM` is the resource type and `orange` is the resource name.

1.4.7 Searching for RIM Object in Tivoli Name Registry

The command to search for a RIM object in the Tivoli name registry is `wlookup`.

Usage: wlookup [-l] -R

where -R displays a list of resource types that are registered.

Usage: wlookup [-r resource_type] [-n resource_name] -a [-L | -o] | name

where -a lists all instances of the specified resource type in the Tivoli name registry and -r specifies the type of the resource for which the command searches. If omitted, the default resource type is distinguished.

```
[root@panda]/> wlookup -r RIM -a
inventory      1140726473.1.1242#RIM::RDBMS_Interface#
orange 1140726473.9.139#RIM::RDBMS_Interface#
tec           1140726473.1.1020#RIM::RDBMS_Interface#

[root@panda]/>
```

Figure 11. List All RIM Objects in TMR

The RIM object named `inventory` is created when you install the Tivoli Inventory product. The RIM object named `tec` is created when you install the Tivoli Enterprise Console product. We created the RIM object named `orange` to allow RIM access to the Oracle Starter Database.

The RIM objects `inventory` and `tec` do not have to use the same RIM host, but in this example, the RIM host for RIM objects `inventory` and `tec` is the same managed node (object dispatcher 1). The RIM host for RIM object `orange` is a different managed node (object dispatcher 9). Both RIM hosts are in the same TMR (region `1140726473`).

1.5 Testing the Connection between RIM Host and Database Server

The commands `wrimtest` and `wrimtrace` are used to test the connection between the RIM host and the database server. These commands can be used before you install a Tivoli application that uses the RIM. This technique allows you to verify that the RIM host can connect to the database server.

The `wrimtest` and `wrimtrace` commands can be entered from any managed node, but the trace is run and the RIM database log is stored on the RIM host.

1.5.1 RIM Test Command

The `wrimtest` command is not intended for the casual user, but can be used to verify that the RIM host can connect to the database server.

Usage: wrimtest [-l rim_object_name]

where the default rim_object_name is inventory.

```
[root@panda]/> wrimtest -l orange

Resource Type   : RIM
Resource Label  : orange
Host Name       : polar
User Name       : sys
Vendor          : Oracle
Database        : ORCL
Database Home   : C:\ORANT
Server ID       :
Instance Home   :
Opening Regular Session...Session Opened
RIM : Enter Option >?
Options -
  c : RIM_commit
  d : RIM_delete
  e : RIM_execute_sql
  g : RIM_retrieve_rows
  i : RIM_insert_rows
  r : RIM_rollback
  u : RIM_update_rows
  ? : Help List
  x : Exit
RIM : Enter Option >x
Releasing session

[root@panda]/>
```

Figure 12. Display RIM Test Options

The wrimtest options can be used by an experienced database user to manipulate data in database.

1.5.2 RIM Trace Command

The command to set the RIM trace level is wrimtrace:

Usage: wrimtrace rim_object_name [new_trace_level]

The valid entries for new_trace_level are ERROR, INFORMATION, and TRACE_OFF. If you want both ERROR and INFORMATION messages, you must enclose the new trace levels in double quotes. If you enter the wrimtrace command without a new trace level, the command displays the current trace setting.

```

[root@panda]# wrimtrace orange

RIM Tracing is turned off

[root@panda]# wrimtrace orange "error information"

[root@panda]# wrimtrace orange

RIM Trace Level:  ERROR|INFORMATION

[root@panda]# wrimtrace orange trace_off

[root@panda]# wrimtrace orange

RIM Tracing is turned off

[root@panda]#

```

Figure 13. Set RIM Trace Options

The new trace level entries are not case sensitive.

1.5.3 RIM Error Log

When an exception occurs, any RDBMS-specific error messages are included in the exception. Error messages are also logged to a file.

Exceptions contain:

- RIM error message
- Database function that caused the error
- Return code from the database function
- Database error message

The default RIM error log file location is `/tmp/rim_db_log`. The directory, such as `/tmp`, that contains the RIM error log must exist before you run the trace. You can set the location by creating a `RIM_DB_LOG` variable in the `oserv` environment.

```
odadmin environ get [od ... |clients|all]
```

The `get` option retrieves the method environment. The default is the current method environment for the `oserv` on the managed node where you run the `odadmin` command.

For the `set` option, the method environment is read from standard input (`stdin`). The command line interface is terminated, pressing `Ctrl-D`.

```
odadmin environ set [od ... |clients|all]
```

Since you must enter all the method environment from the command line, you may find it easier to put the method environment variables in a file and redirect standard input to use the file instead of the keyboard.

```
[root@panda]> cat /u/plassman/envIRON.set

PATH=/bin:/usr/bin
SHLIB_PATH=
TZ=CST6CDT
NLSPATH=/Tivoli/msg_cat/%L/%N.cat:/usr/lib/nls/msg/%L/%N:/usr/lib/nls/msg/%L/%N.cat:/usr/dt/nls/msg/%L/%N.cat:/usr/dt/lib/nls/msg/%L/%N.cat:/usr/dt/lib/nls/msg%1/%t/%c/%N.cat:/usr/dt/lib/nls/msg/%1/%c/%N.cat
RIM_DB_LOG=/tmp/rim_db_log

[root@panda]> odadmin environ set < /u/plassman/envIRON.set

[root@panda]>
```

Figure 14. Set Method Environment for TMR Server from TMR Server

If you want to set the method environment for the RIM host from a managed node that is not the RIM host, you must specify the object dispatcher (od) for the RIM host.


```

[root@panda]# odadmin odlist

Region      Disp  Flags  Port          IPaddr      Hostname(s)
1140726473  1     ct-    94            9.3.1.47    panda
              8     ct-    94            9.3.1.134   grizzly
              9     ct-    94            9.3.1.35    polar
              84    ct-    94            9.3.1.133   kodiak

[root@panda]# odadmin environ get 9

NLSPATH=C:/Tivoli/msg_cat/%L/%N.cat
RIM_DB_LOG=/tmp/rim_db_log

[root@panda]# cat /u/plassman/environ.polar

NLSPATH=C:/Tivoli/msg_cat/%L/%N.cat
RIM_DB_LOG=/var/rim_db_log

[root@panda]# odadmin environ set 9 < /u/plassman/environ.polar

[root@panda]# odadmin environ get 9

NLSPATH=C:/Tivoli/msg_cat/%L/%N.cat
RIM_DB_LOG=/var/rim_db_log

[root@panda]#

```

Figure 15. Set Method Environment for the RIM Host from the TMR Server

For the new trace level to be in effect, you must terminate the RIM process on the RIM host that talks to the RDBMS system. You can use the UNIX kill command or the Windows NT Task Manager (applications option), as appropriate. The name of the process is RIM_<vendor>_prog for UNIX and RIM_<vendor>_prog.exe for Windows NT where <vendor> is Sybase, Oracle, DB2, or MS_SQL. This RIM process will re-start automatically when a RIM_connect is requested. For example, the `wrimtest` command will cause the RIM process to start on the RIM host.

If the trace level is set to ERROR, the trace output is written only if an error occurs. You can create an error by modifying your RIM object to contain an attribute that prevents the RIM from connecting to the RDBMS server.

Chapter 2. Configuring IBM DB2 for Use with RIM

This chapter guides you through the steps necessary to install the IBM Database 2 (DB2) database product and access the DB2 database via the Tivoli RIM.

2.1 IBM DB2 Overview

The DB2 Product Family offers open, industrial-strength database management for business intelligence, transaction processing, and a broad range of applications for all types of businesses.

DB2 Common Server includes an industry-leading optimizer that selects the fastest route to your data. It also includes powerful built-in functions to develop applications with audio, video, images, and documents. Further, DB2, the relational database that manages more than 70 percent of the world's business data, is now available on Windows NT.

2.1.1 DB2 Versions Supported

RIM from the Tivoli Framework pre-GA version 3.6 supports:

- IBM Database 2 (DB2) for Windows (Version 2.1.2)
- IBM Database 2 (DB2) for AIX (Version 2.1.2)

RIM is expected to support the following DB2 versions at Tivoli Framework Version 3.6 GA or soon after GA:

- DB2 Universal Database Version 5 for AIX
- DB2 Universal Database Version 5.0 for the Windows NT
- DB2 Connect Version 5.0 for the Windows NT

The exact delivery date is not know at this time.

2.1.2 DB2 Database Objects

DB2 is a Relational Database Management System (RDBMS) that is organized in different multiple objects. The following are a few of these.

2.1.2.1 Instance

There can be multiple instances on a single server, each of which can be stopped or started without influencing other instances on the server. Each instance has an owner and an associated home directory.

On AIX and other UNIX systems, the instance name and the instance owner's name is the same. Therefore, before an instance can be created, a user with the same name must exist. The instance home directory is the same as the instance owner's home directory. Most of the instance code is stored in this directory.

On NT and other PC operating systems, the instance owner and the instance does not necessarily have the same name. A default instance with the name DB2 is created when the product code is installed. In this case the instance home directory refers to the directory with the same name as the instance name.

The instance owner or System Administrator (SYSADM) has the authority to stop, start and configure the specific instance. To give a user SYSADM authority on an instance, the user must belong to a group that is defined to the instance as the SYSADM group. We'll see how that is done in a later paragraph.

Note

An example of where multiple instances could be used on a server could be in a production and development environment. If the SYSADM of the development instance stops the development instance, the production instance will not be stopped or hindered from normal execution.

2.1.2.2 Database

A database is a grouping of related data. There can be multiple databases defined in a single instance. It is also important to note that a single database can not exist in more than one instance, but a single instance can have more than one database defined.

A SYSADM user of an instance can create a database.

2.1.2.3 Tables

In a database, there can be multiple tables that store the user data. Each table can be referenced by the user ID that created the table and its name. An example of a table reference is:

db2admin.tablename

The user ID in this case is commonly referred to as the schema. The predefined schemas that can not be used to create user tables are:

SYSIBM

SYSCAT

SYSSTAT

All the tables that have these three schema names are created and used by DB2 to store information related to the database.

Figure 16 on page 25 is a graphical representations of the DB2 objects on a AIX server.

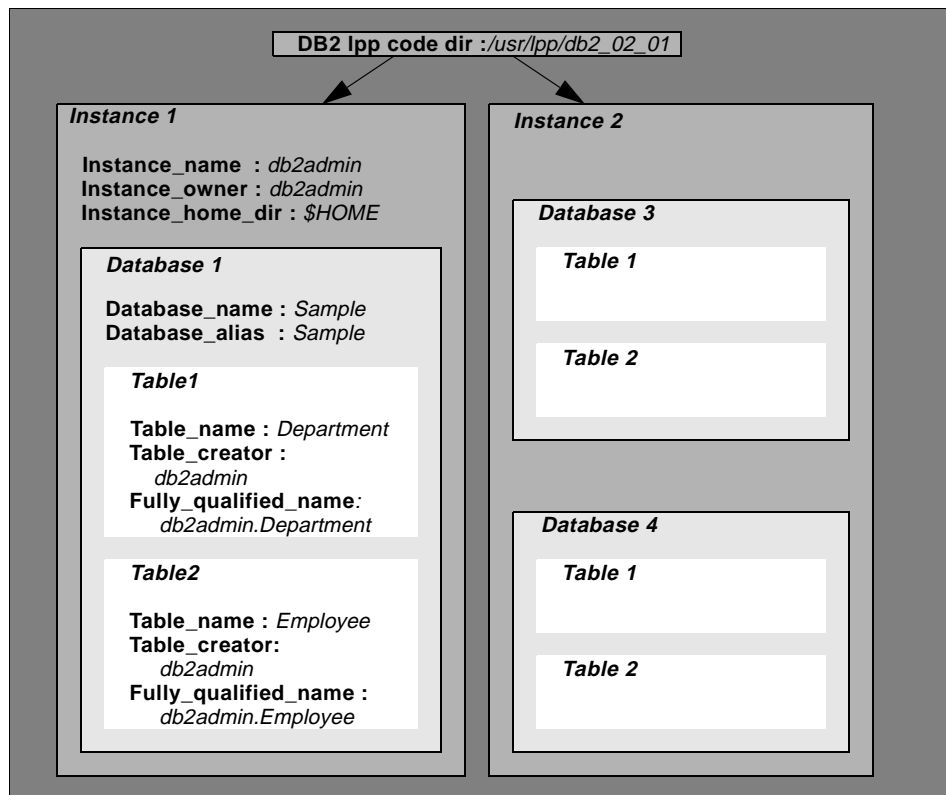


Figure 16. DB2 Objects on AIX

Figure 17 on page 26 is a graphical representations of the DB2 objects on a Windows NT server.

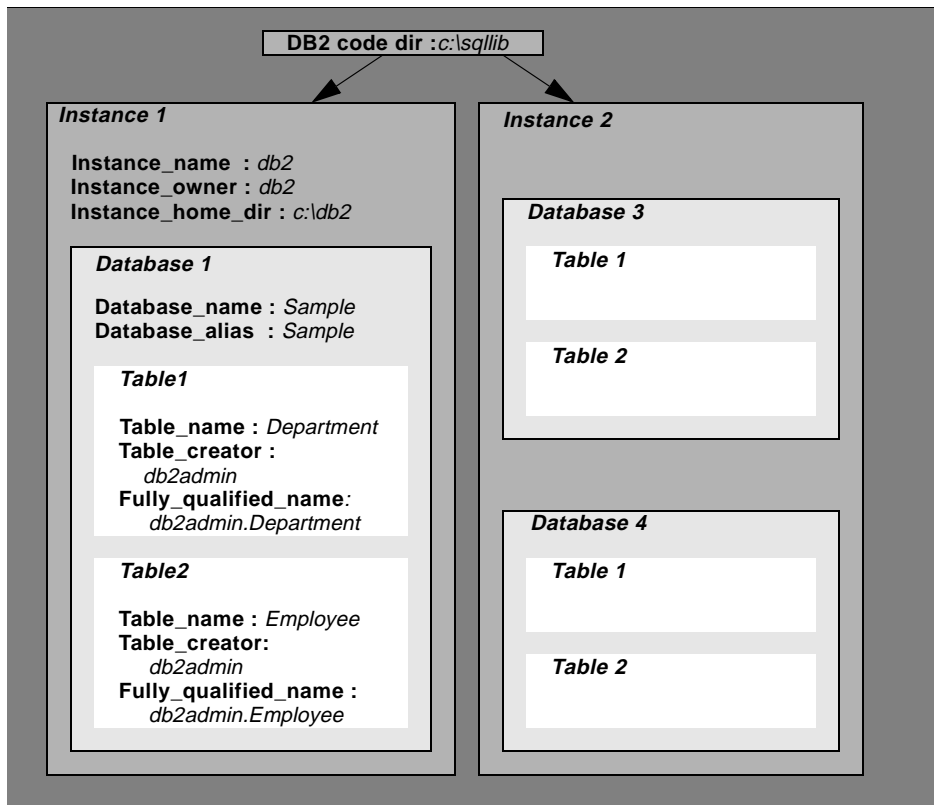


Figure 17. DB2 Objects on NT

2.2 Creating the DB2 System Administrator User

Before installing DB2, a DB2 System Administrator user must be defined. If, on AIX, an instance owner user ID is not created prior to the creation of an instance, the instance creation will fail. The length of the user ID is limited by DB2 to a maximum length of eight characters. This restriction prevents the default Windows NT administrative user (Administrator) from being used as an instance owner.

2.2.1 Creating a SYSADM Group and User for Windows NT

The following steps should be taken to create the user and its group. In our case, we created a user db2 and group sysadm.

Note

The user ID of the instance owner and the instance name should be the same. RIM will use this user ID to connect and execute SQL statements. If a user ID is not the same as the instance name, RIM will not be able to establish a connection to the database.

1. Click on the Start button select the Programs => Administrative Tools => User Manager. The following window shown in Figure 18 on page 27 should open.
2. Select User => New User

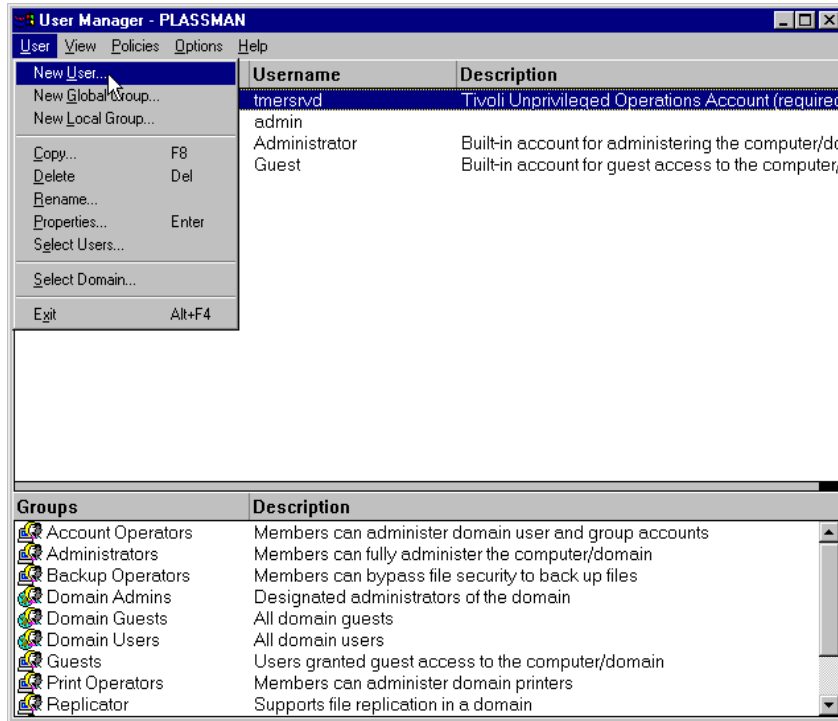


Figure 18. Windows NT User Manager - New User

3. Fill the particulars for the DB2 system administrator user in the New User dialog box and click on add as shown in Figure 19 on page 28.

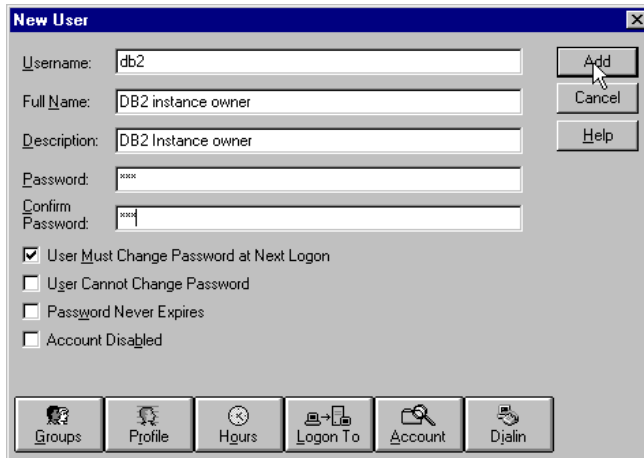


Figure 19. Windows NT New User

To define the SYSADM group:

1. Select New Local Group in the User Manager window, Figure 18 on page 27. Fill in the required information in the dialog box. The name of the SYSADM group can be any name. We chose sysadm to keep things simple.
2. Click on Add as shown in Figure 20 on page 28 to add the user ID of the instance owner to this group.

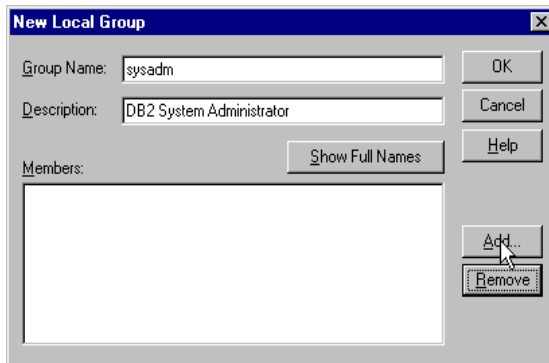


Figure 20. Windows NT New Local Group

3. Select the user or users you want to give SYSADM authority from the Add Users and Groups panel as shown in Figure 21 on page 29. We selected db2admin, our instance owner. Click on the OK button when done.

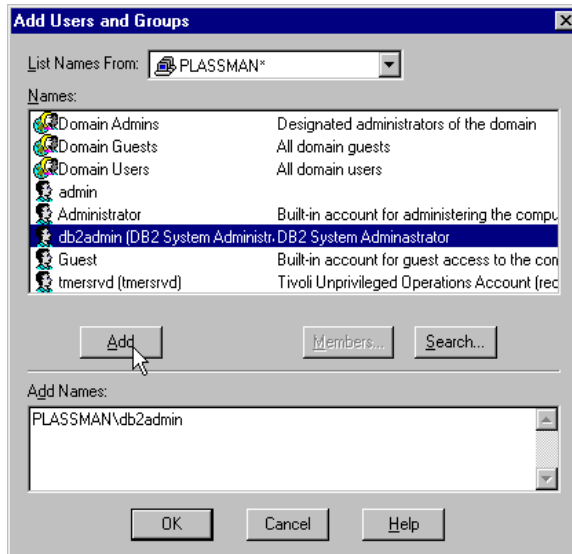


Figure 21. Windows NT Add Users and Groups

The instance owner's user ID and the group the user ID belongs to are defined now. In our setup, the instance owner is db2admin and the SYSADM group is sysadm.

2.2.2 Creating a SYSADM Group and User for AIX

To create the required SYSADM group, follow these steps.

1. Type in `smit` on a xterm window. Click on **Security & Users => Groups**. A window similar to Figure 22 on page 30 should be displayed.

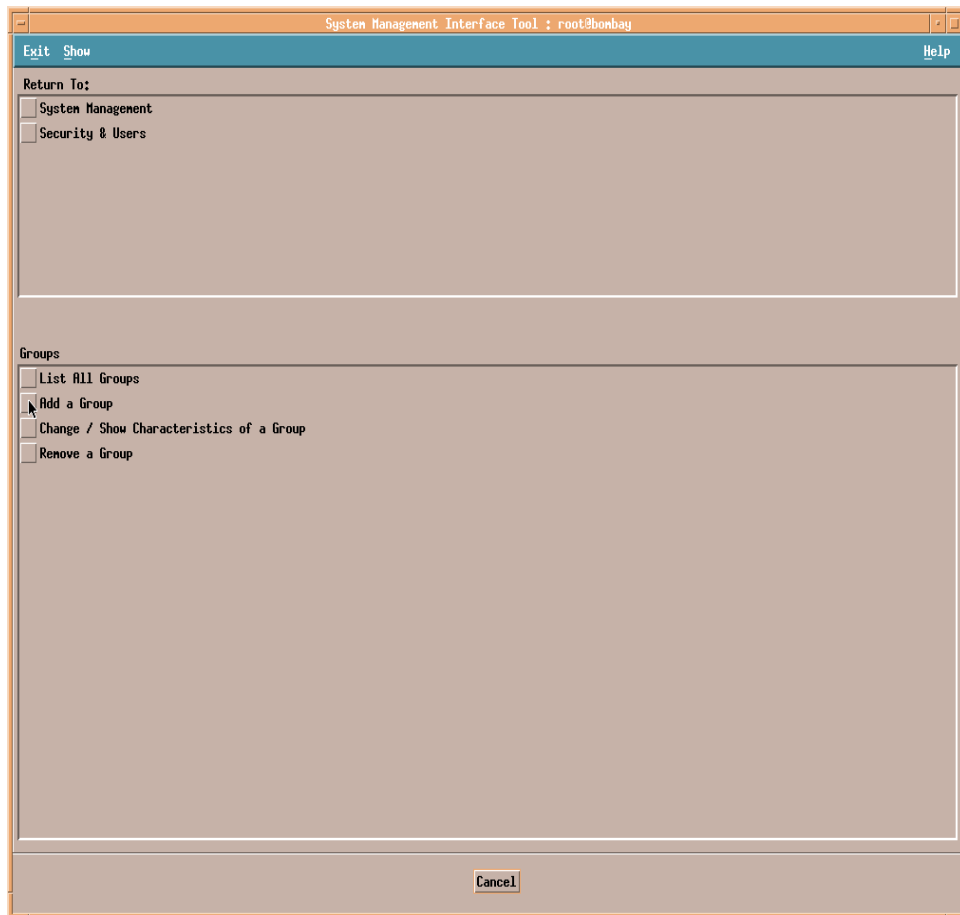


Figure 22. SMIT Security & Users

2. Click on **Add Group** and fill in the required information as shown in Figure 23 on page 31. We chose the SYSADM group to be named SYSADM, but any name could be used.
3. Click on the **OK** button to create the group.

The SYSADM group should now be defined with no members belonging to it.



Figure 23. SMIT Add a Group for DB2

To create the instance owners user ID do the following.

1. Select **Security & Users => Users** in SMIT. The SMIT window should now look similar to the one in Figure 24 on page 32.



Figure 24. SMIT Users

2. When selecting Add User, the Add User panel should appear prompting for information about the new user as shown in Figure 25 on page 33.
3. Fill in the required information. We chose the user ID of the instance owner to be db2admin.
4. Make sure the primary group of the instance owner is SYSADM group previously created.
5. The home directory of the instance owner will house the instance and possibly multiple databases, so make sure the home directory is chosen in a file system big enough to hold the instance executables and databases created with default parameters.

We chose the directory /db2 which is a file system of 200 MB.

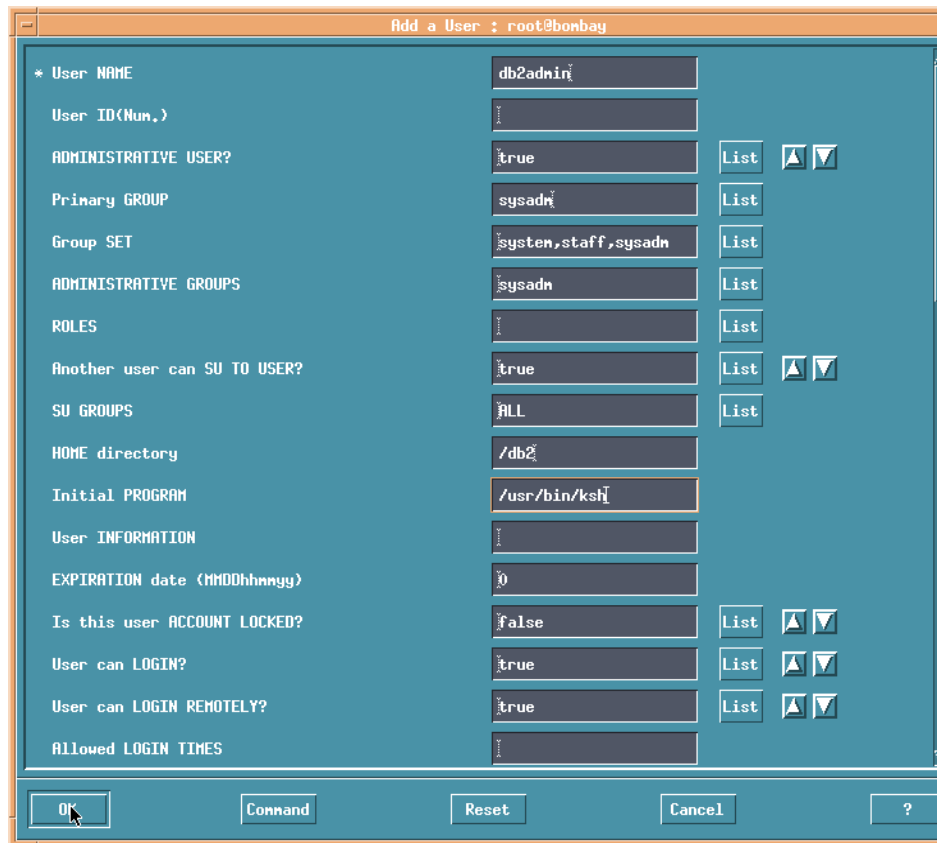


Figure 25. SMIT Add User for DB2

- Change the instance owner's password by issuing the following command as root.

```
passwd db2admin
```

The instance owner's user ID and the SYSADM group are defined.

2.3 Installing DB2 Common Server Version 2.1.2

DB2 Version 2 extends the industrial-strength technology of DB2 on MVS, VM, VSE, and OS/400 platforms to the client/server environment, giving you equivalent server function on OS/2, Windows NT, AIX, Solaris, HP-UX, and other UNIX-based platforms. That's why IBM calls it DB2 Common Server.

2.3.1 Installing DB2 Common Server for NT

In our DB2 for NT setup, the TMR server is toronto, the RIM host is sydney, and the DB2 database server is london. In our setup, we installed the database server on a NT machine outside our TMR.

Below are the steps we used to install DB2 in our setup:

1. Log on to the NT server using the SYSADM user's ID and password.
2. Insert the DB2 Common Server Version 2.1.2. or the DB2 Database Server Version 4 installation CD in the CD-ROM. Execute the setup.exe executable that can be found in the d:\i386\en\disk1 subdirectory. If an error window appears as shown in Figure 26 on page 34, you are not logged on as the SYSADM user or the SYSADM user's ID is more than 8 characters in length.

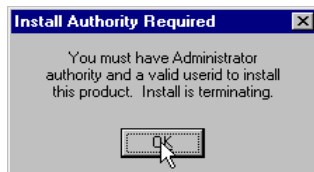


Figure 26. DB2 Installation Error

Note

The two letter subdirectory in the i386 directory indicates the language code of a particular installation image. In our setup, we are using the EN_US installation code. More information on this code can be found in the READ.ME file on the installation CD.

3. Select the DB2 product to install from the Install selection panel, Figure 27 on page 35. In our setup, we want to install the Server version.

The Software Developer's Kit (SDK) need only be installed if application development will be done on the database server. If application development will be done on clients only, then there is no need to install it on the server.

The DB2 World Wide Web Connection provides the ability to access data stored in DB2 from a web browser.

The DB2 DDCS Multi-User Gateway provides the ability to connect from a DB2 Common Server Client to DB2 on MVS or any other DRDA compliant relational database. This product is not needed if the only connections to

and from the DB2 server will be from clients running DB2 Common Server Client Application Enablers commonly referred to as CAE's or other DB2 Common Server Servers.

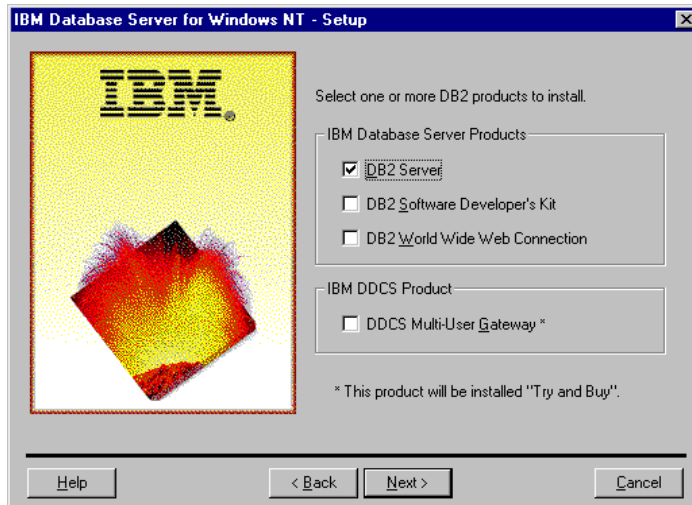


Figure 27. DB2 Installation Selection

4. Enter the licence key supplied with the installation media in the dialog box on the Licensing panel as shown in Figure 28 on page 36. Click on the **Next** button.

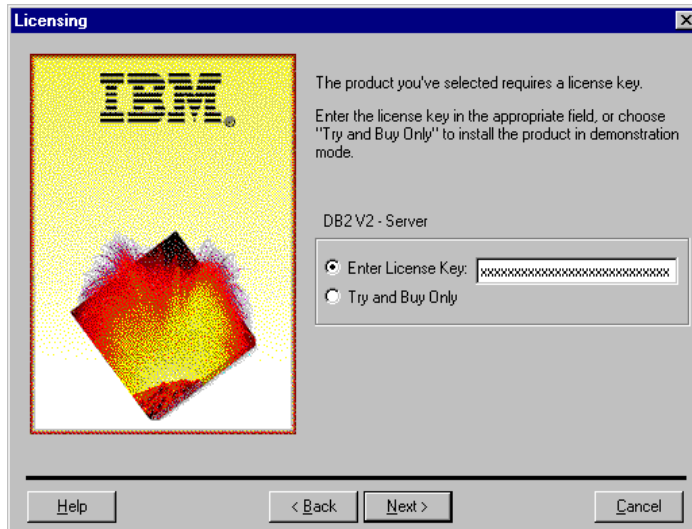


Figure 28. DB2 Licensing

5. Select the type of installation from the Setup options panel as shown in Figure 29 on page 36. We chose the custom installation option.

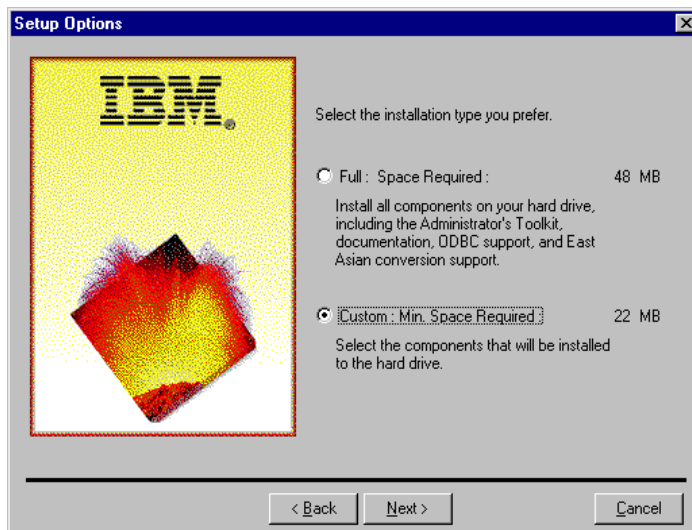


Figure 29. Setup Options

6. The DB2 Components panel shows a list of optional components that can be installed. Information about the components can be found by clicking

on the Description button. We chose the Administrator's Toolkit, Documentation and the IBM DB2 ODBC Driver, as shown in Figure 30 on page 37. By clicking on the Next button the installation of selected components will begin.

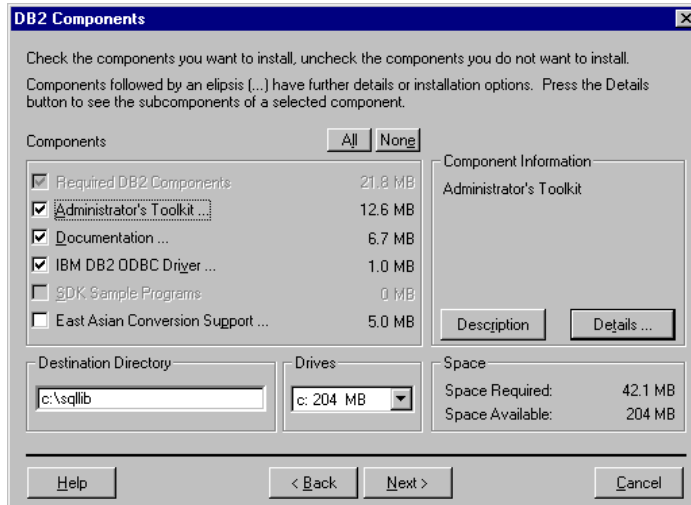


Figure 30. DB2 Server Components

7. Reboot the database server and make sure to log on as the instance owner.
8. Select **Start => Programs => DB2 for Windows NT => Command Window**. In the command window, execute the command:

```
c:\sqlib\bin> db2start
```

This command will start the DB2 database server. We now want to create a sample database.

9. On the command line in the command window, execute the following command to create a sample database and load the sample data.

```
c:\sqlib\bin> db2sampl
```

The name of the sample database is sample and the alias of the database is also sample.

10. Connect to the database using the following command:

```
c:\sqlib\bin> db2 connect to sample
```

11. Verify that the tables are created by issuing the command:

```
c:\sqlib\bin> db2 list tables for all
```

A list of all the table created in the database is returned, if the sample database was created successfully and the DB2 database server installed correctly.

12. Disconnect from the database and stop the instance by issuing the following commands:

```
db2 connect reset  
db2stop
```

The DB2 database server is now installed and a local database with an alias sample is created and populated with data.

2.3.2 Starting DB2 for NT Server Automatically

By default the process associated with the DB2 for NT server does not start at boot time. To change the default behavior, we used the following steps:

1. Open the Windows NT Services window from the Windows NT Control Panel by double clicking on the Services icon.
2. Select the database instance you want to start automatically on the Windows NT Services window. In our setup, we selected DB2-DB2 on Figure 31 on page 38.

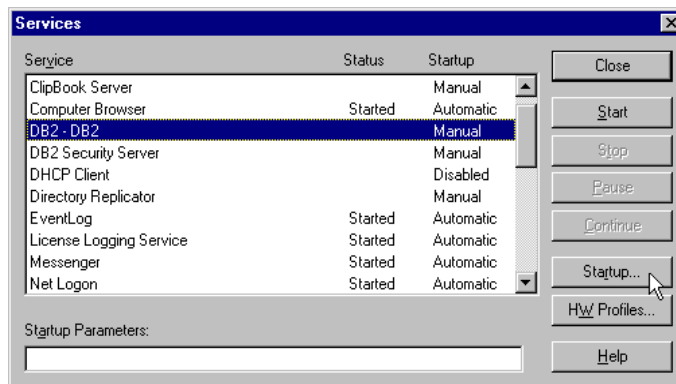


Figure 31. Windows NT Control Panel Services - DB2

3. After clicking on the Startup button, the Service window appears. See Figure 32 on page 39.
4. Select **Automatic** and **This Account**. Click on the '...' button and select the user ID of the instance owner in the list provided. Enter the password of the instance owner and confirm it then click on the OK button. The word Automatic should now be displayed next to the instance name.

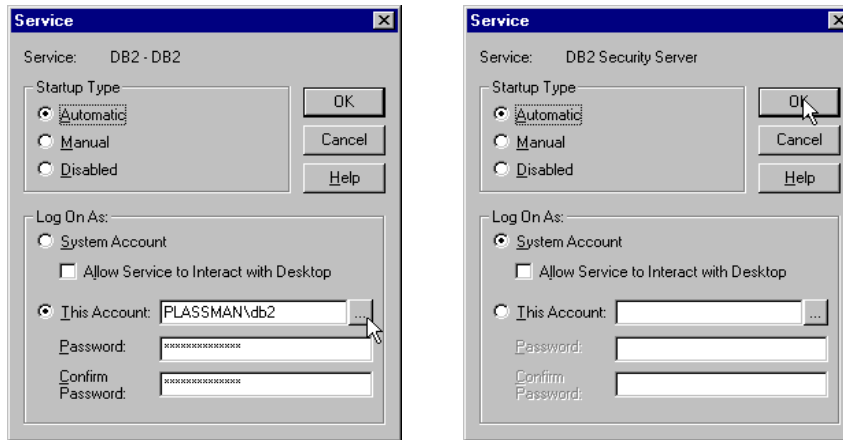


Figure 32. Database Instance and Security Server Services

5. Select the Security Server from the Windows NT Services window. In our setup, we selected DB2 Security Server as shown in Figure 31 on page 38.
6. After you click on the Startup button, the Service window appears. See Figure 32 on page 39.
7. Select **Automatic** and **System Account**. Click on the **OK** button. The word Automatic now appears next to both the Instance and Security Server entries as shown in Figure 33 on page 39.

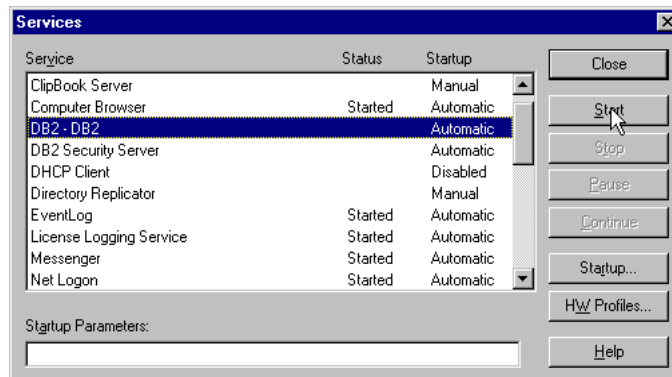


Figure 33. Database Instance and Security Server Status

2.3.3 Start the Security Server and Database Instance for NT

By default, the DB2 database instance or security server are not started. You can start these services manually. We used the following steps:

1. Select the DB2 database instance from the list in the Window NT Services window and click on the **Start** button as shown in Figure 33 on page 39.
2. Select the **Security Server** from the list and click on the **Start** button.

2.3.4 Installing DB2 Common Server for AIX

DB2 Common Server Version 2.1.2. for AIX can be installed using the `installp` command or SMIT. In this section, we discuss the install procedure using SMIT.

Note

The installation must be done by the root user and not the database instance owner defined previously.

We used the following steps:

1. Once in SMIT, select **Install** and **Update** from LATEST Available Software as shown in Figure 34 on page 41.

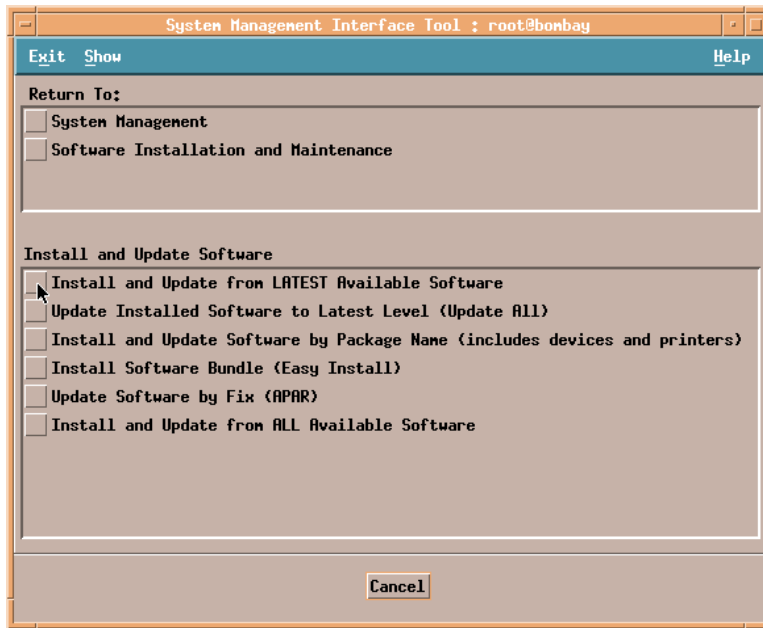


Figure 34. SMIT Software Installation and Maintenance

2. Specify the location of the installation image. It is normally `cd0`, but, in our setup, the installation image is in the `/db2/db2img` directory.

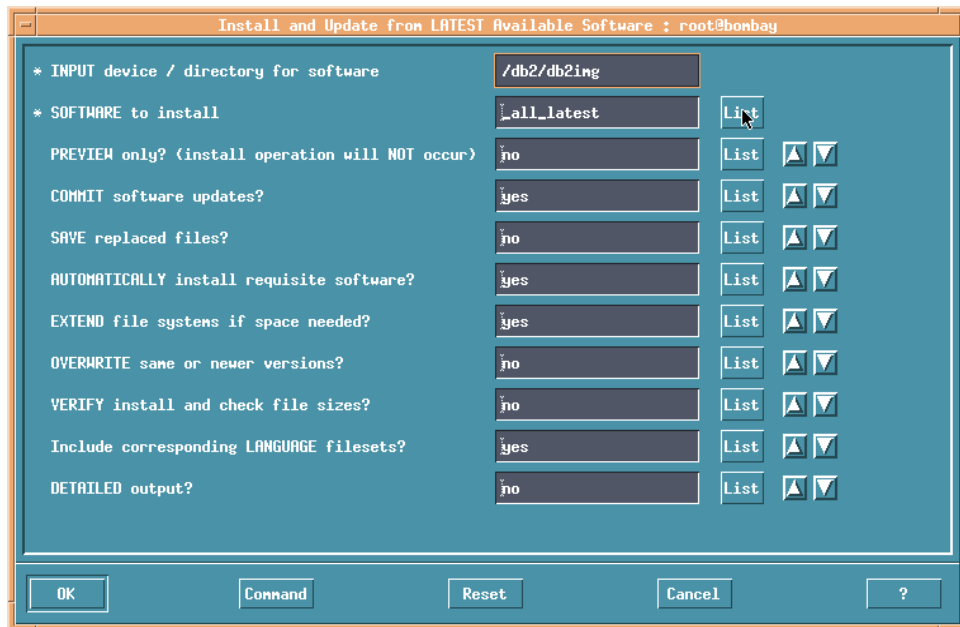


Figure 35. SMIT Install and Update from Latest Available Software

3. Click on the **List** button next to the SOFTWARE to install dialog box to specify file sets to install. Figure 36 on page 44 shows the selection we made:
 - DB2 Client Application Enabler provides the interface for an application to connect and communicate with the server.
 - DB2 Code Page Conversions - If a client and its server has different code pages this module converts data sent and received to a matching code page on the server or client.
 - DB2 Command Line Processor provides the ability for a administrator to issue `db2` commands on the command line interactively.
 - DB2 Communications Support - Base with TCPIP provides TCP/IP connectivity to the server.
 - DB2 Database Director is the administrative GUI interface of DB2.
 - DB2 Executables are the executables necessary to run DB2
 - DB2 Performance Monitor provides a method for the DBA to monitor performance and provides information necessary for tuning database and or instance variables.

- DB2 Product Library - INF - XX_XX is the online documentation for a specific code page designated by the XX_XX. In our setup, the code page is En_US.

Note

To insure that the AIX component, bos.loc.pc_compat.XX_XX, is installed so that the XX_XX online manuals can be read, you enter the following AIX command:

```
locale -a
```

If XX_XX is not returned, install bos.loc.pc_compat.XX_XX from the AIX installation media.

- DB2 Product Library - Postscript - XX_XX provides the documentation in Postscript format that can be printed out and kept as a hard copy. This is an optional component and is not required by the DB2 server to execute.
- DB2 SDK Utilities and Samples are only required if application development is planned, and then they're only required on the client or server where the applications will be written. This is an optional component.
- DB2 Utilities and Samples provide the DBA with utilities to move and manage data and some samples.



Figure 36. SMIT Software to Install List for DB2

2.3.5 Creating Database Instance Using DB2 for AIX

The next step is to create an instance of DB2 where we will create future databases. Already defined is a system administrator or instance owner. In our setup, it is db2admin. We now define an instance with the same name.

Note

With DB2 for AIX, the database instance name and the user ID of the instance owner must be the same.

To create the DB2 instance, follow these steps:

1. As the root user, execute the following command:

```
/usr/lpp/db2_02_01/instance/db2icrt db2admin
```

where `db2admin` is the user ID of the instance owner and the name of the new instance.

2. Check to see if it was created by using the following command:

```
/usr/lpp/db2_02_01/instance/db2ilist
```

If the name of the new instance is returned, the instance was created. A new directory was created in the instance owner's home directory named `sqllib`. In this directory and its subdirectories, all the executables, samples, and so on, for the instance are stored.

3. Switch user to the instance owner's user ID:

```
su - db2admin
```

4. The DB2 environment variables must be setup for the instance owner. A sample script file named `db2profile` can be found in the `$HOME/sqllib` directory. Make sure `DB2COMM` variable is set to the protocol being used to connect to the server. In our setup, the protocol is `TCPIP`.

```
DB2COMM=TCPIP
```

```
export DB2COMM
```

5. Insert to following line in the `.profile` for the instance owner:

```
. $HOME/sqllib/db2profile
```

Be sure to enter the dot and space in front of the path.

6. Save and execute the `.profile`

```
. $HOME/.profile
```

Be sure to enter the dot and space in front of the path.

7. You can test to see if the environment is set correctly by starting the database manager.

```
db2start
```

Note

SQL5043N Start database manager processing was successful.
However, support for one or more communication protocols failed to start successfully.
This error message is displayed because the instance and protocol stack has not yet been configured, which we will do in a later section.

8. Create a sample database by executing the following command:

```
$HOME/sql1lib/misc/db2sampl
```

This command creates a sample database named SAMPLE and populates the database with data.

9. By connecting to the database and selecting a list of all the tables, we can verify if the database was created successfully. We used the following commands:

```
db2 connect to sample  
db2 list tables for all
```

An instance is now created and a sample database created in the instance. The instance owner or system administrator has access to both the instance and the sample database.

The next step is to configure an installed client to communicate with the database server.

2.4 Installing DB2 Common Server CAE Version 2.1.2

The CAE (Client Application Enabler) provides an interface for applications executing on client machines to connect and manipulate data located in a database on a DB2 database server. The CAE software is independent of the operating system on the server. Any DB2 CAE can connect to any of the DB2 Common Servers.

2.4.1 Installing DB2 CAE for NT

The DB2 Client Application Enabler Version 2.1.2 for Windows NT can be found on the CAE CD-ROM provided with the DB2 Version 2.1.2 Server for Windows NT. We used the following steps to install DB2 CAE for NT:

1. Insert the installation CD. From a command window, execute the installation program `d:\win32\i386\en\disk1\setup.exe`.

2. Follow installation prompts until asked for type of install. We selected custom install as shown in Figure 37 on page 47.

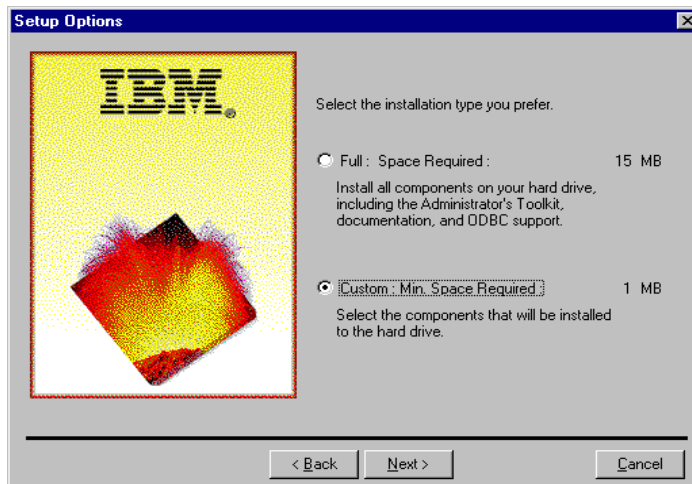


Figure 37. CAE Setup Options - Installation Type

3. Select the optional software to install from the DB2 components panel as shown in Figure 38 on page 47. Descriptions of the software packages are read by clicking on the Description button. We chose the Administrator's Toolkit which provides a GUI interface to perform database administration tasks.

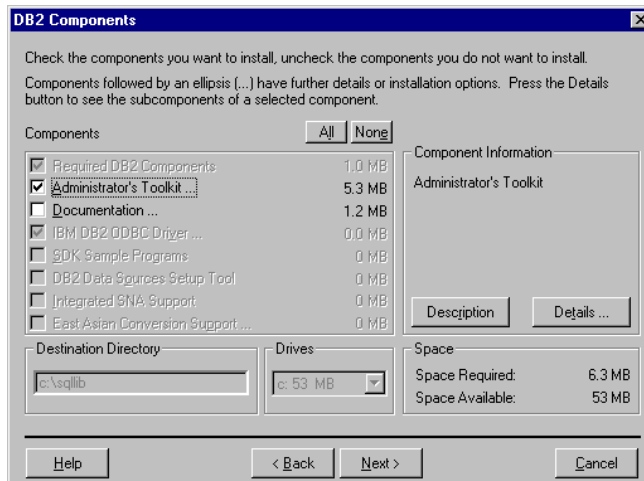


Figure 38. DB2 Client Components

4. Click on the **Next** button, after the selection is made, to start the installation of the required and selected optional software.

The installation of the DB2 Common Server Version 2.1.2 CAE is now complete.

The next step is to configure both the client and the server so that the client can connect to the database on the server.

2.4.2 Installing DB2 CAE for AIX

The DB2 CAE is installed on each machine that will run applications to access a DB2 Server for NT, OS/2, AIX or even MVS. In this section, we describe how to install the DB2 CAE Version 2.1.2 for AIX and how to configure the CAE to communicate with a DB2 server using TCPIP protocol.

On the client AIX machine, we used the following steps:

1. Create an instance owner as described in Part 2.2.2, "Creating a SYSADM Group and User for AIX" on page 29.
2. In SMIT, select the Install and Update from LATEST Available Software as shown in Figure 34 on page 41.
3. Specify the device or directory where the install image of DB2 can be found.
4. Click on the List button next to the Software to install dialog box. See Figure 35 on page 42.
5. Select DB2 Client Application Enabler from the list displayed. See Figure 39 on page 49.

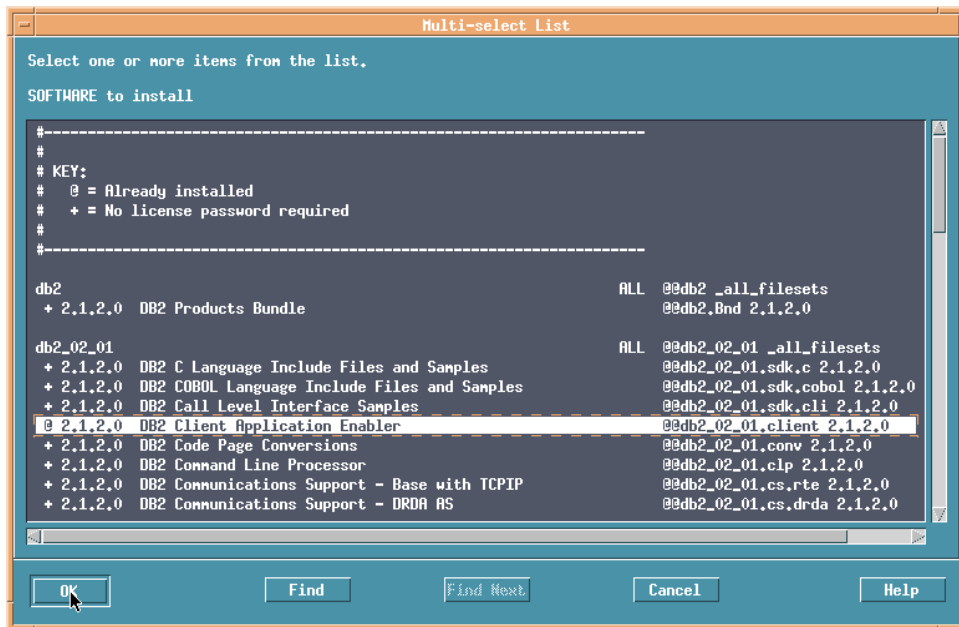


Figure 39. SMIT Software to Install List

- After installing, change to directory `/usr/lpp/db2_02_01/instance` and execute the command:

```
db2icrt <instance_name>
```

Replace `<instance_name>` with the user ID of the instance owner. In our setup, we used `db2admin` for the instance owner user ID.

- Log on as the instance owner.
- Edit the `.profile` file of the instance owner and add the following line after the `set PATH` statement.

```
. $HOME/sqlllib/db2profile
```

Be sure to enter the dot and space before the path.

- Execute the `.profile` file.

```
..profile
```

Be sure to enter the dot and space before the path.

The DB2 Client Application Enabler is now installed.

Next we can configure both the client and the server so applications running on the client can work with the database on the server.

2.5 Configuring DB2 Version 2.1.2 Connectivity

This section describes how the DB2 CAE Version 2.1.2. and the DB2 Server Version 2.1.2. can be configured using the TCP/IP protocol.

At this point, we assume that the DB2 Version 2.1.2 Server and Client are installed on either NT or AIX. We also assume a database named sample was created on the database server.

2.5.1 Configuring DB2 Server for NT

To get the client to communicate to the database server, you configure the server to accept and service requests from the client. We used the following steps:

1. Log on to the database server with the SYSADM user ID. See “Creating a SYSADM Group and User for Windows NT” on page 26.
2. Update the SVCENAME parameter using the following command to update the parameter:

```
db2 update dbm cfg using SVCENAME db2
```

or

```
db2 update database manager configuration using SVCENAME db2
```

This parameter matches a name of a port reserved in the c:\winnt\system32\drivers\etc\services file for the instance in which it is defined. We chose the name to be the same as the instance name db2.

3. You reserve two unused ports for the instance by inserting the following two lines into the c:\winnt\system32\drivers\etc\services file. We chose port 5500 and 5501.

```
db2          5500/tcp
```

```
db2i        5501/tcp
```

The first port is used as the primary port for a client to send and receive data from the server. The second port is used by DB2 Version 1.x clients as an interrupt port. See Figure 40 on page 51.

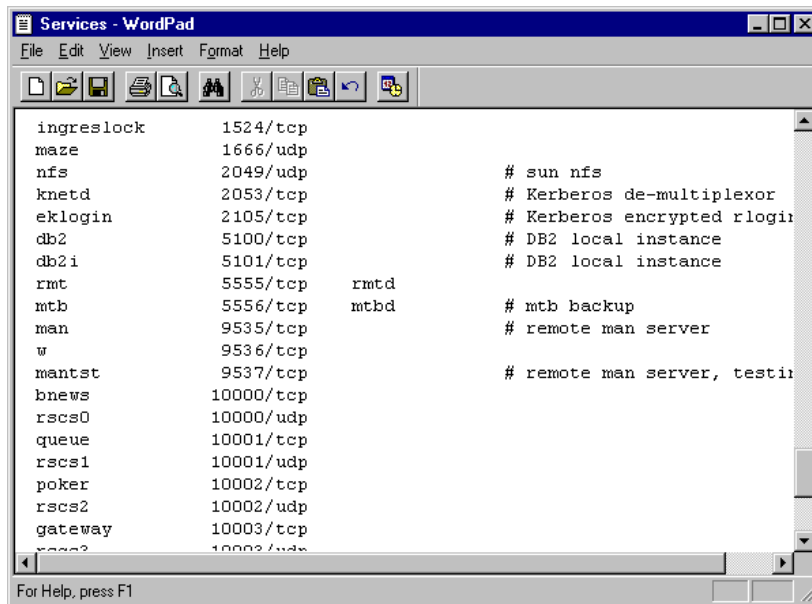


Figure 40. Services File - DB2 Server on NT

4. To make the change effective, stop and then start the instance by issuing the following commands in a DB2 Command Line Processor window, which can be found by selecting **Start => Programs => DB2 for Windows NT => Command Line Processor**.

```

c:\sqllib\bin> db2stop
c:\sqllib\bin> db2start

```

The server is now configured to service requests from clients.

2.5.2 Configuring DB2 CAE for NT

We configure the client to send service requests to the database server.

1. Edit the `c:\winnt\system32\drivers\etc\services` file using the port numbers reserved for the instance in the services file on the server, and a name chosen for the client node. See Figure 41 on page 52. We chose the port name `db2tiv`.

```

erlogin      888/tcp          # Login and environment passing
kpop         1109/tcp         # Pop with Kerberos
phone        1167/udp
ingreslock   1524/tcp
maze         1666/udp
nfs          2049/udp         # sun nfs
knetd        2053/tcp         # Kerberos de-multiplexor
eklogin      2105/tcp         # Kerberos encrypted rlogin
db2tiv       5100/tcp         # DB2 remote instance for Tivoli
db2tivi      5101/tcp         # DB2 remote instance for Tivoli
mtb          5556/tcp         # mtb backup
man          9535/tcp         # remote man server
w            9536/tcp
mantst       9537/tcp         # remote man server, testing
bnews        10000/tcp
rscs0        10000/udp
queue        10001/tcp
rscs1        10001/udp

```

Figure 41. Services File - DB2 Client on NT

2. If you are not using DNS, insert the host name of the server machine and its IP address in the c:\winnt\system32\drivers\etc\hosts file. In our setup, the host name is london and its IP address is 9.3.1.115. See Figure 42 on page 52.

```

# Copyright (c) 1993-1995 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows NT.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97      rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

127.0.0.1      localhost
9.3.1.115     london

```

Figure 42. Hosts File - DB2 Client on NT

- Execute the client configuration utility by selecting **Start => Programs => DB2 for Windows NT => DB2 Client Setup**. The DB2 Client Setup window appears. See Figure 43 on page 53.

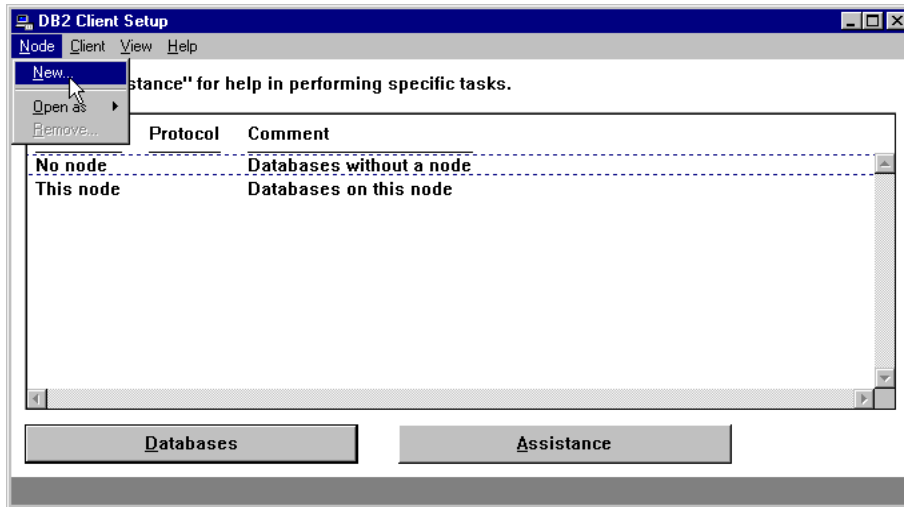


Figure 43. DB2 Client Setup

- Select New Node from the menu and fill the required information in the Node Settings panel. See Figure 44 on page 53.

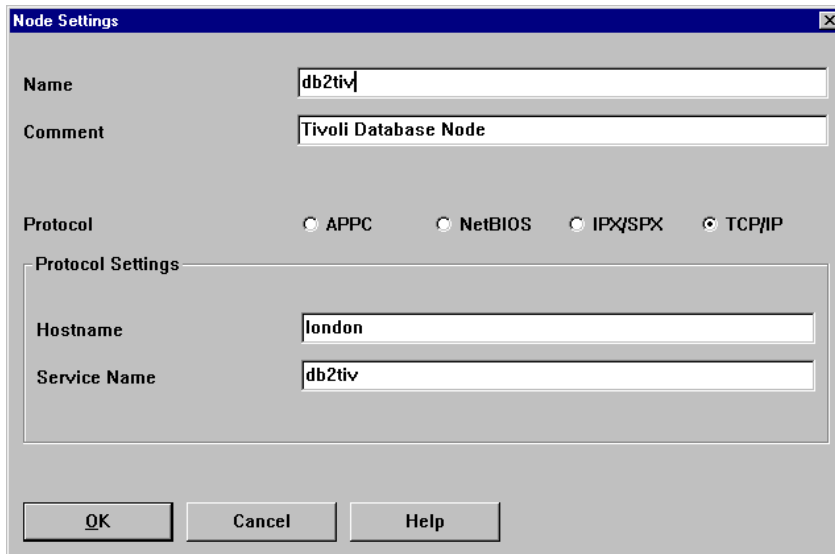


Figure 44. DB2 Node Settings

The node name is an identifier DB2 gives the configured connection to an instance. Remember, there can be more than one databases in an instance, and this node can have more than one database associated with the node. We chose db2tiv, the same name as the service name, but the node name can be different from the service name.

The host name is the DB2 server in the DB2 client's hosts file. In our setup, the host name is london.

The service name refers to the name associated with first port number reserved in the \etc\services file of the DB2 client for communication with the DB2 server. In our setup, the service name is db2tiv.

5. Click on **Database** when the newly defined node is highlighted. In the DB2 Client Setup - Databases window (see Figure 45 on page 54), select the **Database => New** from the menu bar.

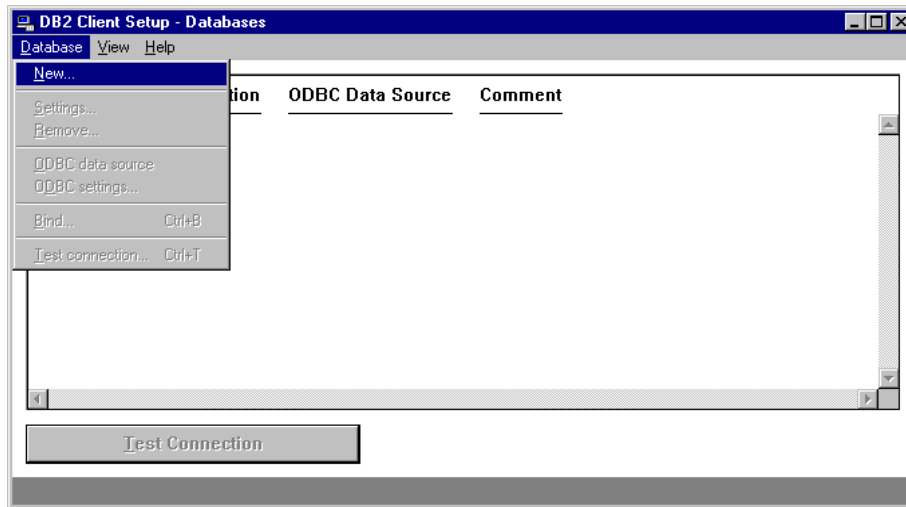


Figure 45. DB2 Client Setup - Databases

6. Complete the **New Database** window.

Name is the database alias name on the server.

Alias field provides a method of giving the database an alias on the local machine. In our setup, the sample database on the DB2 server is called rimsmp1 on the DB2 client.

If the alias field is not filled in, a default alias is created with the same name as the name of the database.

7. Click on the **Test Database Connection** button in the New Database window (see Figure 46 on page 55) to verify if the DB2 client is configured correctly.

The screenshot shows a 'New Database' dialog box with the following fields and options:

- Name:** sample
- Alias:** rimssmpl
- Comment:** Sample database on London
- Location:** At Node: DB2TIV
- Authentication Type (DB2 1.x Database)
- Buttons:** Test Database Connection, OK, Cancel, Help

Figure 46. DB2 New Database

8. Supply the user ID and password of the instance owner on the DB2 server user and click on the **OK** button in the IBM DB2 ODBC Diver window as shown in Figure 47 on page 56.

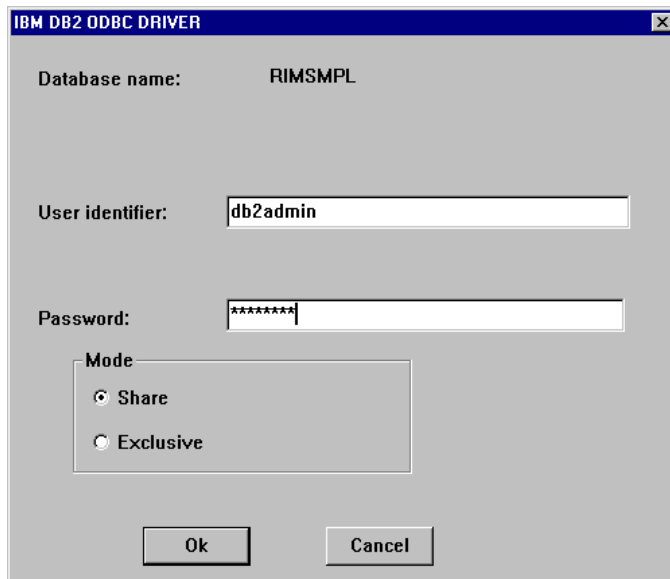


Figure 47. IBM DB2 ODBC Driver - Test Connection

If a message is received stating that the connection established successfully, the configuration of the DB2 Client and the DB2 Server is complete.

2.5.3 Configuring DB2 Server for AIX

In this section, we describe how to configure the DB2 Server Version 2.1.2 for AIX to communicate with a DB2 CAE using TCPIP for the protocol.

On the DB2 server, we used the following steps:

1. Log on as the instance owner.
2. Update the SVCENAME parameter.

This parameter matches a name of a port reserved in the `/etc/services` file for the instance in which it is defined. We chose the name to be the same as the instance name `db2admin`. Use the following command to update the parameter:

```
db2 update dbm cfg using SVCENAME db2admin
```

or

```
db2 update database manager configuration using SVCENAME db2admin
```

3. Log on as the root user.

4. Reserve two unused ports for the instance. We chose port 5500 and 5501. Insert the following lines into the /etc/services file:

```
db2admin      5500/tcp
db2admini     5501/tcp
```

The first port will be used as the primary port for a client to send and receive data from the server. The second port is used by DB2 Version 1.x clients as an interrupt port.

```
rfa           4672/udp          # Remote File Access Server
complex-main  5000/tcp
complex-main  5000/udp
complex-link  5001/tcp
complex-link  5001/udp
rfe           5002/tcp          # Radio Free Ethernet
rfe           5002/udp          # Radio Free Ethernet
rmonitor_secure 5145/tcp
rmonitor_secure 5145/udp
padl2sim      5236/tcp
padl2sim      5236/udp
db2admin      5500/tcp          # DB2 - db2admin instance port
db2admini     5501/tcp          # DB2 - db2admin instance interrupt
sub-process   6111/tcp          # HP SoftBench Sub-Process Cntl.
sub-process   6111/udp          # HP SoftBench Sub-Process Cntl.
xdsxdm       6558/udp
```

Figure 48. Extract from /etc/service File

5. Log on as the instance owner db2admin.
6. To make the change effective, stop and then start the instance by issuing the following commands on the command line:

```
db2stop
db2start
```

The server is now configured to service requests from clients.

2.5.4 Configuring DB2 CAE for AIX

In this section, we describe how to configure the DB2 CAE Version 2.1.2 for AIX to communicate with a DB2 Server using TCP/IP for the protocol.

On the client, we used the following steps:

1. Log on as the instance owner db2admin.
2. Catalog the (server) node on the client.

```
db2 catalog tcpip node db2tiv remote bombay server db2tiv
```

The node directory is a list of remote instances that DB2 can connect to. The node name refers to a port number in the /etc/services file and a host name. In our setup, the port number (referred to by name) is db2tiv and the host name of the server is bombay.

3. As the root user, edit the services file (/etc/services on Unix and c:\winnt\system32\drivers\etc\services on NT) using the port numbers reserved for the instance in the services file on the server, and the name chosen for the node.

```
db2tiv          5500/tcp
db2tivi         5501/tcp
```

4. Add the host name of the database server and its IP address to the /etc/hosts file (c:\winnt\system32\drivers\etc\hosts on NT)

```
9.135.3.76      bombay
```

5. Catalog the database on the new node.

```
db2 catalog database sample as rimsmpl at node db2tiv
```

In our setup, we cataloged a database named sample on the machine with host name bombay using port db2tiv (5500) and gave the database an alias of rimsmpl. From now on, the database is referred to as rimsmpl on the client machine. Defining an alias with the as alias clause is optional. If the as rimsmpl clause is left out, a default alias is defined with the same name as the database. In other words, we would refer to the database, in our setup, as sample.

6. Test the connection between the DB2 client and DB2 server.

```
db2 connect to rimsmpl user db2admin using password
```

If the connection is made successfully, the configuration of both the client and server has been made correctly.

2.6 Creating a RIM Object to Access a DB2 Database

In this section, we describe how to create a RIM object on the DB2 client, which must be a managed node. We followed these steps to define a RIM object for the DB2 sample database:

1. Log on to the TMR server with the RIM administrator's user ID and password.
2. Issue the `wcrtrim` command which creates a RIM object.

2.6.1 DB2 RIM Object

Table 4 on page 59 shows the DB2-specific values that go into the RIM object.

Table 4. DB2 RIM Attributes

Parameter	Value	Description
RDBMS Vendor	DB2	Database vendor name - must be DB2 or db2
RIM Host	host_name	The managed node where the DB2 client software is installed.
Database ID	database (instance)	The name of the DB2 database instance. Specifies the database name (or alias) as defined on the DB2 client. If the database server and the RIM host is on the same machine, the parameter refers to the database name (or alias) as defined on the DB2 server.
Database Home	rdbms_home	The directory where the DB2 client software is installed. Normally this directory is c:\sqllib on NT or /usr/lpp/db2_02_01/ on AIX. See 2.6.5, "Determining the Database Home Directory" on page 63 for more information. Whenever you specify a NT directory as a parameter value for a RIM object, use the '/' to separate the parent and subdirectories. For example, if the RDBMS home is the c:\sqllib directory, you should use the string c:/sqllib for the RDBMS directory.
Server ID	TCPIP	For DB2, you must specify the string tcp/ip.
RDBMS User Name	user	The user ID of the instance owner. For DB2, the user must be the same as the instance name. On AIX, this will always be the case, but, on NT, you must insure that a user with the same name as the database instance is created.
RDBMS User Password		When the RIM object is created, you are prompted for the RDBMS User password. See description for RDBMS User Name above. If you change the RDBMS User password for the database product, you must change the password stored in the RIM object with the <code>wsetrimpw</code> command.

Parameter	Value	Description
Instance Home	instance_home	For DB2 only, this directory is created when the database instance is created. Use the '/' when specifying a NT directory (c:/db2). On AIX, instance home refers to the \$HOME directory of the instance owner. See 2.6.6, "Determining the Instance Home Directory" on page 65 for more information.

For AIX, the Database Home directory is used by the `wcrtrim` command to create a logical link named `libdb2.a` in the `$LIBDIR` directory that points to the `<rdbs_home>/lib/libdb2.a` file where `<rdbs_home>` is the database home directory. The link is created the first time a DB2 RIM object is created. The link is not deleted if any or all of the DB2 RIM objects are deleted.

Note

For AIX, if the Instance Home variable is set incorrectly the first time, you must delete the `$LIBDIR/libdb2.a` logical link before attempting to redefine the RIM object. Otherwise the old (and incorrect) link is used.

In our setup, the DB2 database instance named `db2` (on NT) has an instance home directory `c:\db2`. The DB2 database instance named `db2admin` (on AIX) has an instance home directory `/db2`.

2.6.2 Creating RIM Object for NT Host

In our setup, we have a database with alias `sample` on the DB2 for NT server with host name `london`. We have a managed node with the DB2 for NT CAE installed that we are using as a RIM host with host name `sydney`. On the RIM host, we reference the database on the database server with a locally (on the client) defined alias `rimsmpl`. The client software is installed in the `c:\sqllib` (default) directory. On the server, we have an instance defined named `db2` and the instance home directory is `c:\db2`. We created a RIM object for this database with the `wcrtrim` command executed from the TMR server. See Figure 49 on page 61.


```
c:\> wcrtrim -v DB2 -h sydney -d rimsampl -u db2 -H c:/sqllib
-s tcpip -I c:/db2 rimsampl

RDBMS password:***

c:\>
```

Figure 49. Create RIM Object for NT Host

If you do not specify the `-i` argument, you are prompted for a password, which can be no more than eight characters. In our setup, the password was three characters (see the three asterisks after the password prompt).

2.6.3 Creating a RIM Object for AIX Host

In our setup, we have a database with alias `sample` on the DB2 for AIX server with host name `bombay`. We have a managed node with the DB2 for AIX CAE installed that we are using as a RIM host, and its host name is `dublin`. On the RIM host, we reference the database on the database server with a locally defined alias `rimsampl`. The client software is installed in the `/usr/lpp/db2_02_01/` (default) directory. On the server, we have an instance defined named `db2admin` and the instance owners home directory is `/db2`. We created a RIM object for this database with the `wcrtrim` command executed from the TMR server. See Figure 50 on page 61.

```
root@chicago[/#] wcrtrim -v DB2 -h dublin -d rimsamp -u db2admin
-H /usr/lpp/db2_02_01 -s tcpip -I /db2 sample

RDBMS password:*****

root@chicago[/#]
```

Figure 50. Create RIM Object for AIX Host

The `wcrtrim` command can be executed at any managed node in the local TMR. In our setup, we executed `wcrtrim` on the TMR server, `chicago`, but could just as easily have used the managed node, `dublin`. We could not have executed the `wcrtrim` command on the RDBMS server, `bombay`, because RDBMS server is not in the local TMR.

2.6.4 Testing the RIM Object

We used the `wrimtest` command from the TMR server (toronto) and the RIM object sample to connect to a DB2 for NT database (alias `rimssmpl`) on the database server (london) through the RIM host (sydney).

```
D:\Tivoli\bin\w32-ix86\bin>wrimtest -l rimssmpl

Resource Type : RIM
Resource Label : rimssmpl
Host Name      : sydney
User Name      : db2
Vendor         : DB2
Database       : rimssmpl
Database Home  : c:/sqllib
Server ID      : tcpip
Instance Home  : c:/db2
Opening Regular Session...Session Opened
RIM : Enter Option >x
Releasing session

D:\Tivoli\bin\w32-ix86\bin>
```

Figure 51. RIM Test - DB2 for NT

We used the `wrimtest` command from the TMR server (chicago) and the RIM object sample to connect to a DB2 for AIX database (alias `rimssmpl`) on the database server (bombay) through the RIM host (dublin).

```
root@chicago[/#]# wrimtest -l sample

Resource Type : RIM
Resource Label : sample
Host Name      : dublin
User Name      : db2admin
Vendor         : UNKNOWN
Database       : rimssmpl
Database Home  : /usr/lpp/db2_02_01
Server ID      : tcpip
Opening Regular Session ..Opened
RIM : Enter Option > x
Releasing session

root@chicago[/#]
```

Figure 52. RIM Test - DB2 for AIX

To see a list of valid options enter option `'?`'. To exit enter option `'x'`

Note

The DB2 Instance Home is not shown in the `wrimtest` output for DB2 for AIX RIM object.

The RIM objects are now created and we verified that the RIM objects can be used connect to databases.

2.6.5 Determining the Database Home Directory

The Database Home parameter refers to the directory where DB2 is installed on the client.

If the RIM host is on a NT server, the installation directory is `c:\sqllib` by default.

If the RIM host is on a AIX server, the Database Home directory is `<instance_home>/sqllib` by default, where `<instance_home>` is the home directory of the instance owner. In our setup, the `<instance_home>` is `/db2`.

To verify the selection, check the directory for a subdirectories, such as `adsm`, `bin`, `bnd`, `cfg`, `dba`, `function`, `java`, `misc`, `msg`, `samples`.

```
c:\> dir c:\sqllib

Volume in drive C has no label.
Volume Serial Number is 68D9-6FA0

Directory of C:\sqllib

[.]          [..]          [adsm]        [bin]          [bnd]
[book]       [cfg]         [DB2]        db2cli.ini    db2clias.lst
DB2CLP.DAT  DB2IDIR      db2java.htm  db2java.txt   db2sys1t
DB2SYSIM    [dba]        EVALUATE.AGR [function]     [help]
[install]   [java]       [misc]       [msg]         [msodbc16]
readme.txt  [samples]
                27 File(s)
                197,199 bytes
                168,031,744 bytes free
```

Figure 53. Database Home Directory - DB2 for NT Server

Figure 53 on page 63 displays the Database Home on the DB2 for NT database server (london).

```

c:\> dir c:\sqlib

Volume in drive C has no label.
Volume Serial Number is 68D9-6FA0

Directory of c:\sqlib

[.]          [..]          [bin]          [bnd]          [book]
[.cfg]       [DB2]         db2cli.bak    db2cli.ini    db2clias.lst
DB2CLP.DAT  DB2IDIR      db2java.htm  db2java.txt  DB2SYSTEM
[help]       [install]    [java]        [misc]        [msg]
[msodbc16]  readme.txt   [samples]
                23 File(s)      243,201 bytes
                    51,897,344 bytes free

```

Figure 54. Database Home Directory - DB2 for NT Client

Figure 54 on page 64 displays the Database Home on the DB2 for NT client and RIM host (sydney).

```

root@bombay[ / ]# ls /db2/sqlib

.netls      adsm      db2cshrc   doc        map        sqlbdir
.rtelib     bin       db2dump    function   misc       sqlnodir
.rtelib2    bnd       db2profile include     msg        tmp
Readme      cfg       db2system  java       odbclib
adm         conv      dba        lib        samples

```

Figure 55. Database Home Directory - DB2 for AIX Server

Figure 55 on page 64 displays the Database Home on the DB2 for AIX database server (bombay).

```

root@dublin[/#] ls -pla /db2/sqlib

total 112
lrwxrwxrwx 1 root sysadm 24 Apr 02 10:25 .netls -> /usr/lpp/db2_02_01/netls
lrwxrwxrwx 1 root sysadm 26 Apr 02 10:25 .rtelib -> /usr/lpp/db2_02_01/.rtelib
lrwxrwxrwx 1 root sysadm 27 Apr 02 10:25 .rtelib2 -> /usr/lpp/db2_02_01/.rtelib
lrwxrwxrwx 1 root sysadm 25 Apr 02 10:25 Readme -> /usr/lpp/db2_02_01/Readme
drwxr-xr-x 2 db2admin sysadm 512 Apr 02 10:25 adm/
lrwxrwxrwx 1 root sysadm 23 Apr 02 10:25 adsm -> /usr/lpp/db2_02_01/adsm
lrwxrwxrwx 1 root sysadm 22 Apr 02 10:25 bin -> /usr/lpp/db2_02_01/bin
lrwxrwxrwx 1 root sysadm 22 Apr 02 10:25 bnd -> /usr/lpp/db2_02_01/bnd
drwxrwsr-t 2 db2admin sysadm 512 Apr 02 10:25 cfg/
lrwxrwxrwx 1 root sysadm 23 Apr 02 10:25 conv -> /usr/lpp/db2_02_01/conv
-rwxr-xr-x 1 db2admin sysadm 9707 Apr 02 10:25 db2cshrc
drwxrwsrwt 2 db2admin sysadm 512 Apr 02 10:25 db2dump/
-rwxr-xr-x 1 db2admin sysadm 9876 Apr 02 10:25 db2profile
-rw-rw-r-- 1 db2admin sysadm 2048 Apr 02 10:25 db2system
lrwxrwxrwx 1 root sysadm 22 Apr 02 10:25 dba -> /usr/lpp/db2_02_01/dba
lrwxrwxrwx 1 root sysadm 22 Apr 02 10:25 doc -> /usr/lpp/db2_02_01/doc
drwxrwsr-t 3 db2admin sysadm 512 Apr 02 10:25 function/
lrwxrwxrwx 1 root sysadm 26 Apr 02 10:25 include -> /usr/lpp/db2_02_01/include
lrwxrwxrwx 1 root sysadm 23 Apr 02 10:25 java -> /usr/lpp/db2_02_01/java
lrwxrwxrwx 1 root sysadm 22 Apr 02 10:25 lib -> /usr/lpp/db2_02_01/lib
lrwxrwxrwx 1 root sysadm 22 Apr 02 10:25 map -> /usr/lpp/db2_02_01/map
lrwxrwxrwx 1 root sysadm 23 Apr 02 10:25 misc -> /usr/lpp/db2_02_01/misc
lrwxrwxrwx 1 root sysadm 22 Apr 02 10:25 msg -> /usr/lpp/db2_02_01/msg
lrwxrwxrwx 1 root sysadm 26 Apr 02 10:25 odbclib -> /usr/lpp/db2_02_01/odbclib
lrwxrwxrwx 1 root sysadm 26 Apr 02 10:25 samples -> /usr/lpp/db2_02_01/samples
drwxrwsr-x 2 db2admin sysadm 512 Apr 02 10:51 sqldbdir/
drwxrwsr-x 2 db2admin sysadm 512 Apr 02 10:50 sqlnodir/
drwxrwsrwt 2 db2admin sysadm 512 Apr 02 10:25 tmp/

root@dublin[/#]

```

Figure 56. Database Home Directory - DB2 for AIX Client

Figure 56 on page 65 displays the Database Home on the DB2 for AIX client and RIM host (dublin).

2.6.6 Determining the Instance Home Directory

The Instance Home parameter refers to the directory on the DB2 client where the instance is created.

On AIX, Instance Home is the home directory of the instance owner. In our setup, the DB2 User Name (instance owner) is db2admin and the DB2 Instance Home is /db2.

```
root@dublin[/]# su db2admin
root@dublin[/]# lsuser db2admin
db2admin id=7 pgrp=sysadm groups=sysadm,system,staff home=/db2
shell=/usr/bin/ksh roles=
root@dublin[/]#
```

Figure 57. Display Instance Owner User Definition

On NT, Instance Home is the instance directory. The instance directory name should be the same as the instance User Name. In our setup, the instance User Name is db2 and the Instance Home is C:/db2.

Even if this parameter is specified incorrectly for a NT RIM host, the `wrimtest` command will work, since this RIM object parameter is not currently used by NT RIM.

Chapter 3. Configuring Oracle for Use with RIM

This chapter describes the use of Oracle as a RDBMS server for Tivoli applications. The RDBMS server should be installed before your Tivoli application, because the Tivoli application installation will ask you to enter some database parameter variables.

If your environment has a supported database installed already, you should contact your database administrator for the values needed to install your Tivoli application.

We set up two environments, one is AIX and the other is Windows NT. In both these setups we installed the RDBMS server outside of the TMR.

3.1 Oracle Overview

Oracle is the most popular RDMS in the world, and the most widely used database by Tivoli customers. Oracle's primary strengths are its portability (Oracle runs on virtually every operating system) and the robustness of its engine. Oracle is the dominant RDBMS vendor primarily because of its portability and an aggressive sales force. Oracle is attempting to expand their product line into object-oriented and web-related technologies. Oracle is also championing the network computer, and has introduced Oracle 8, an object-oriented RDBMS.

3.1.1 Oracle Versions Supported

The RIM supports all versions of Oracle 7.X. However, you should be aware that the following versions have known problems and their use should be avoided. Customers using these versions should be encouraged to get patches or upgrades as appropriate.

Table 5. Oracle Supported Versions

Oracle Version	Description
Prior to 7.1.16.4	Previous versions had a variety of problems. Customers should upgrade.
7.2.2	Has a huge memory leak - upgrade.
7.3.2.1 & 7.3.2.2	Has installation problems - get patches to upgrade to 7.3.3.
7.3.3	Current version of Oracle (as of August 1997).

The RIM did not support Oracle Version 8 when this redbook was written.

3.1.2 SQL*Plus

The Oracle command line SQL interpreter is called `sqlplus` and can be found in `$ORACLE_HOME/bin`. You must have `ORACLE_HOME`, `ORACLE_SID`, and `TWO_TASK` environmental variables set and `$ORACLE_HOME/bin` must be in your `PATH`. The `sqlplus` command has the following syntax:

```
sqlplus [<user id>/<password>] [@sql_script]
```

where `sql_script` is a text file that contains SQL commands. If you just enter `sqlplus` at the command line, you will be prompted for a user ID and password. When you have successfully logged in, you should see a `SQL>` prompt. If not, you will see an Oracle error code and message and then you will be asked for the user ID and password again. You are prompted for the user ID and password a total of three times before exiting. If you cannot log in, write down the Oracle error code and message to give to Tivoli support.

To exit `sqlplus`, type `quit` at the SQL prompt.

3.1.3 SQL*Net

SQL*Net is the Oracle component that allows an Oracle client to communicate with an Oracle server on another machine. SQL*Net consists of an Oracle-specific protocol that rides on top of a network protocol, such as TCP/IP, a listener process that runs on the database server, and configuration files on the server and client that define what servers are available on the network. SQL*Net takes care of all the internal data and character type representations across a heterogeneous network.

3.2 Installing Oracle 7.3.3 for AIX

In our Oracle for AIX setup, the TMR server is `chicago`, the RIM host is `dublin`, the Oracle RDBMS server is `capetown`. In our setup, we installed the RDBMS server on an AIX machine outside our TMR.

3.2.1 Installing Oracle for AIX Server

This section will take you through the steps of installing and configuring the Oracle 7 for AIX Server.

1. Before you start the installation, make sure that the `(bos.adt.base)` fileset is installed. To find out if the fileset is installed, run the following command.

```
# ls1pp -l bos.adt.base
```



```

# lspp -l bos.adt.base

Fileset                Level   State   Description
-----
Path: /usr/lib/objrepos
  bos.adt.base         4.2.1.0 COMMITTED Base Application Development
                               Toolkit
#

```

Figure 58. Verify bos.adt.base Installed

- Two groups must be created, dba and oper. These GIDs are used by the Oracle database administrator and your Oracle operators. You can use SMIT by typing `smit` or `smitty` and then selecting **Security&Users => Groups => Add a Group**. You specify the group names that need to be added, dba and oper.

```

                                Add a Group
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                [Entry Fields]
* Group NAME                    [dba]
ADMINISTRATIVE group?          false      +
Group ID                       []         #
USER list                      []         +
ADMINISTRATOR list             []         +

F1=Help      F2=Refresh    F3=Cancel    F4=List
F5=Reset     F6=Command    F7=Edit     F8=Image

```

Figure 59. SMIT Add Group for Oracle

- You can create your oracle user by using SMIT. Type `smit` (or `smitty`) on the command line and select **Security & Users => Users => Add a User**.

```

                                Add a User

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                [Entry Fields]
* User NAME                          [oracle]
  User ID                             []
  ADMINISTRATIVE USER?                true          #
  Primary GROUP                       [dba]          +
  Group SET                           [dba,staff]   +
  ADMINISTRATIVE GROUPS               []            +
  ROLES                                []            +
  Another user can SU TO USER?       true          +
  SU GROUPS                           [ALL]         +
  HOME directory                      []
  Initial PROGRAM                     []
  User INFORMATION                    []
  EXPIRATION date (MMDDhhmmyy)       [0]
[MORE...37]

F1=Help          F2=Refresh      F3=Cancel      F4=List
F5=Reset         F6=Command       F7=Edit        F8=Image
F9=Shell         F10=Exit         Enter=Do

```

Figure 60. SMIT Add User for Oracle

You can make this user an Administrative User, so that only root can administer this user account. Make sure the umask value is set to 022. You can leave the rest of the fields as default.

Note

Set a password for the user name (UID) that is used to install Oracle, otherwise you can not log in with this user name.

4. Log in as root and edit the /etc/services file to add the SQL*Net V2 listener. If the listener uses port 1526, add the following line in the /etc/services file.

```
listener      1526/tcp          #oracle
```

5. After the oracle user is created, and while you are logged in as the root user, you can create a file system for Oracle. We created a 350 MB file system. The size of your file system is dependent on what Oracle applications that you install. You can refer back to your application installation manual to see the required size for all the applications. We used SMIT to create the file system.

```

smit
=> System Storage Management
=> File Systems
=> Add/Change/Show/Delete File Systems
=> Journaled File System
=> Add a Journaled File System
=> Add a Standard Journaled File System
=> Select the Volume Group that you want to use.'

```

As shown in Figure 61, you must specify the size and the mount point of your file system.

Add a Standard Journaled File System

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]	
Volume group name	rootvg	
* SIZE of file system (in 512-byte blocks)	[700000]	#
* MOUNT POINT	[/oracle]	
Mount AUTOMATICALLY at system restart?	yes	+
PERMISSIONS	read/write	+
Mount OPTIONS	[]	+
Start Disk Accounting?	no	+
Fragment Size (bytes)	4096	+
Number of bytes per inode	4096	+
Allocation Group Size (MBytes)	8	+

F1=Help	F2=Refresh	F3=Cancel	F4=List
F5=Reset	F6=Command	F7=Edit	F8=Image
F9=Shell	F10=Exit	Enter=Do	

Figure 61. SMIT Create Oracle File System

The rest of the fields can be left as default. These specifications create a file system of 350 MB at the mount point /oracle, and the file system is mounted automatically at system restart.

6. When the file system is created, SMIT doesn't automatically mount the file system. You can manually mount the file system:

```
# mount /oracle
```

7. To see all the file systems that are mounted on your machine, you can use the mount command.

```
# mount
```

```

root@dublin[ / ]# mount
node          mounted          mounted over      vfs      date          options
-----
/dev/hd4      /
/dev/hd2      /usr              jfs          Apr 28 11:16  rw,log=/dev/hd8
/dev/hd9var   /var              jfs          Apr 28 11:16  rw,log=/dev/hd8
/dev/hd3      /tmp              jfs          Apr 28 11:16  rw,log=/dev/hd8
/dev/hd1      /home             jfs          Apr 28 11:17  rw,log=/dev/hd8
/dev/lv00     /usr/vice/cache  jfs          Apr 28 11:17  rw,log=/dev/hd8
/dev/lv01     /var/dce          jfs          Apr 28 11:17  rw,log=/dev/hd8
/dev/lv02     /var/dce/adm/dfs/cache jfs          Apr 28 11:17  rw,log=/dev
/hd8
/dev/lv05     /db2              jfs          Apr 28 11:17  rw,log=/dev/hd8
/dev/lv03     /oracle           jfs          Apr 28 11:17  rw,log=/dev/hd8
DFS          /...              dfs          Apr 28 11:18  rw
AFS          /afs              afs          Apr 28 11:19  rw
/dev/lv04     /Tivoli           jfs          Apr 28 11:41  rw,log=/dev/hd8
root@dublin[ / ]#

```

Figure 62. Display Mounted File Systems

8. After the file system is created and mounted, you should create two subdirectories in the file system, one where the database is installed, and one that is the installation link directory used during the Oracle installation from CD-ROM. We used the following directory structure:

```

/oracle/home.....(where database is installed)
/oracle/link.....(links to CD-ROM media)

```

Do this by typing in the following command:

```
# mkdir /oracle/home /oracle/link
```

9. The ownership of your /oracle, /oracle/home and /oracle/link directories is changed to UID=oracle and GID=dba, because the oracle user is the database administrator and needs ownership over these directories. Use the `chown` command to change ownership.

```
# chown oracle:dba /oracle
# chown oracle:dba /oracle/home
# chown oracle:dba /oracle/link
```

10. After the user oracle and installation file system is created, you can set the environment for the user oracle. We created a shell script called `db_setup.sh`. We made an entry into the `/home/oracle/.profile` file so that the environment will be set every time the oracle user log in.

```
. ./db_setup.sh
```

Be sure to enter the dot and space before the path.

Note

Make sure that this entry is inserted after the line where your \$PATH variables are being set, because your \$PATH variables are being changed in this script.

The db_setup.sh file could look like Figure 63 on page 73.

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

Figure 63. Oracle Profile Script (db_setup.sh)

Variables:

- ORACLE_OWNER defines the owner of the Oracle database.
- ORACLE_HOME defines the directory where the Oracle database will be installed.
- PATH defines a list of directories where the Shell searches for executables
- ORACLE_SID variable is specific to a user and defines the name of the database that belongs to the user.

11. You can add and mount the Oracle Installation CD-ROM media as a CD-ROM file system using SMIT.

smit

```
=> System Storage Management
=> File Systems
=> Add/Change/Show/Delete File Systems
=> CDROM File Systems
=> Add a CDROM File System
```

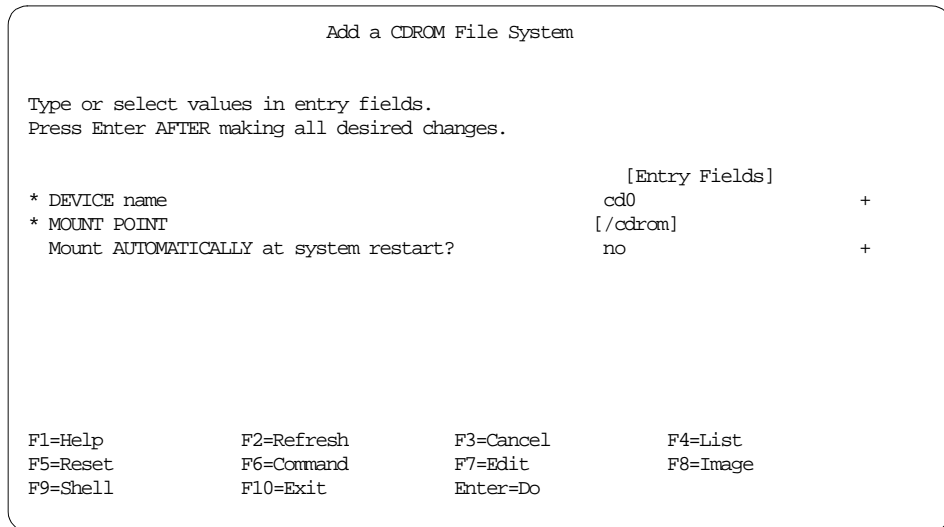


Figure 64. SMIT Add CDROM File System

- In Figure 64 on page 74, we created a CD-ROM file system, that will be mounted at a mount point /cdrom. You can leave the rest of the options as default.

12. SMIT does not automatically mount the CDROM file system when it is created. You can mount it manually by typing:

```
# mount /cdrom
```

NOTE

When mounting and unmounting file systems, you can use the mount point of the file system as the name.

13. As the oracle user, you must now run the /cdrom/orainst/start.sh script. This script defines and sets up your link directory.

```
# su - oracle
$ cd /cdrom/orainst
$ ./start.sh
```

Be sure to enter the dot, but no space, before the path.

When the process starts, you are prompted for the Oracle_link directory, which is /oracle/link in our setup. This process will take about 10 minutes.

14. After the link script completes, log in as root and run a script in your linked directory called rootpre.sh and then exit back to oracle user.

```
$ su - root
enter root's password
# cd /oracle/link/orainst
# ./rootpre.sh
# exit
```

Be sure to enter the dot, but no space, before the path /rootpre.sh.

After you run the rootpre.sh script, you should reboot the machine.

15. As the oracle user, change directory to the orainst directory in the link directory, and run the Oracle Installer (an installation script called orainst). For the orainst script to run, the TERM variable must be set to a specific value. We had our TERM variable set to vt100 and Oracle Installer worked fine with this value. You can also use the Common Desk Environment (CDE). From the CDE, we started an xterm that allowed us to use the Oracle Graphical Installer process.

```
$ xterm &
$ cd /oracle/link/orainst
$ ./orainst /m
```

Be sure to enter the dot, but no space, before the path.

The first window is a prompt that asks you what type of installation you would use.

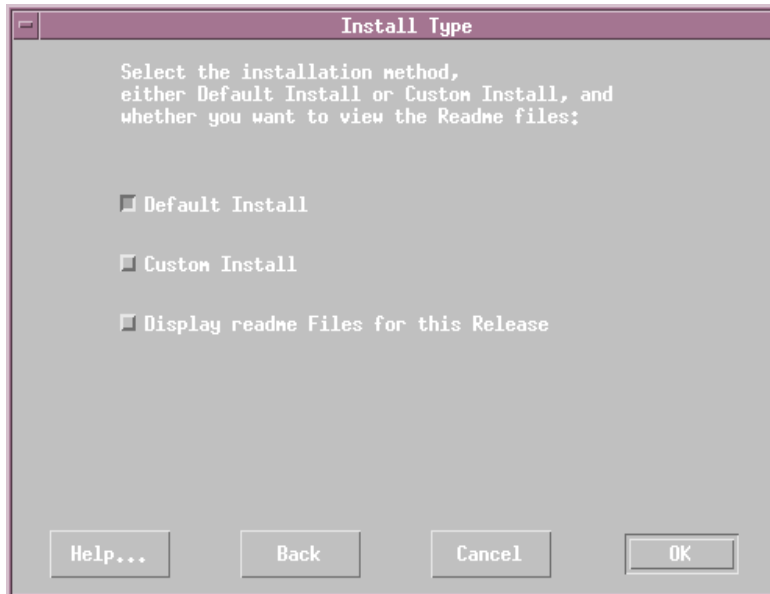


Figure 65. Oracle Installation Menu

16. From this window, we selected:

- => Default Install
- => Install, Upgrade, or Deinstall software.
- => Install New Product, Create DB Object

At this point, see Figure 66 on page 77.

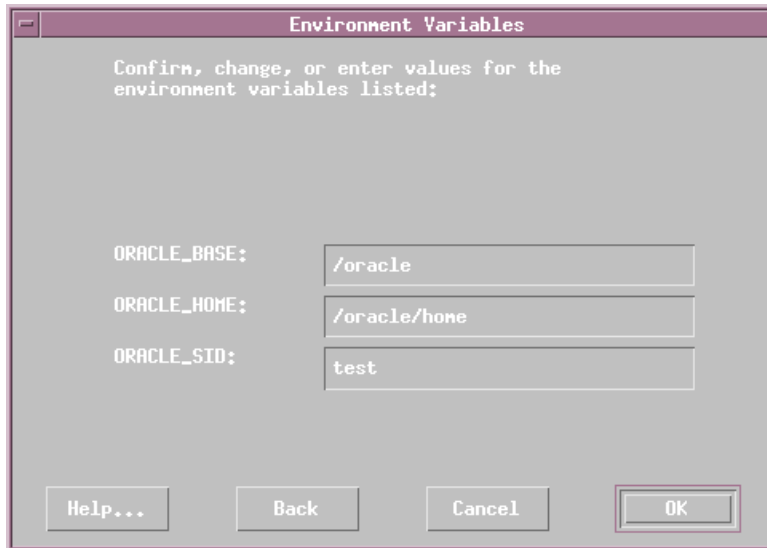


Figure 66. Oracle Environment Variables - Create DB Object

17. You must set the Oracle installation environmental variables for your environment:

ORACLE_BASE is a directory that must be higher in the directory hierarchy than your ORACLE_HOME directory because some of the database configuration files are stored there.

ORACLE_HOME is a directory where your oracle database software is stored.

ORACLE_SID is the database name that is created during installation.

You can see the values we selected in Figure 66 on page 77.

18. On the next screen you are prompted whether or not you ran the rootpre.sh script. If you have, then select yes.

19. The next screen prompts you whether or not you want to relink your Oracle executables, Oracle7 Server will automatically be linked. Linking should be done if you are going to:

- Install an Oracle protocol adapter
- Link Oracle products together
- Change the name of you DBA group from the default (dba)
- Change the name of the OPERATOR group from the default(oper)
- Install the Distributed Database Option

- Install the Spatial Data Option
- Install the Advanced Replication Option
- Install the Parallel Query Option
- Install user exits
- Install patches or bugs fixes

When you relink, the Installer renames and saves some of the old executables. Once the new executables are in place, you can remove the old ones to save disk space. The old executables are in the \$ORACLE_HOME/bin directory with an O appended to the end of their names (for example, exp is renamed exp.O).

We selected to relink.

20. Figure 67 on page 78 displays the values that are used for this installation.

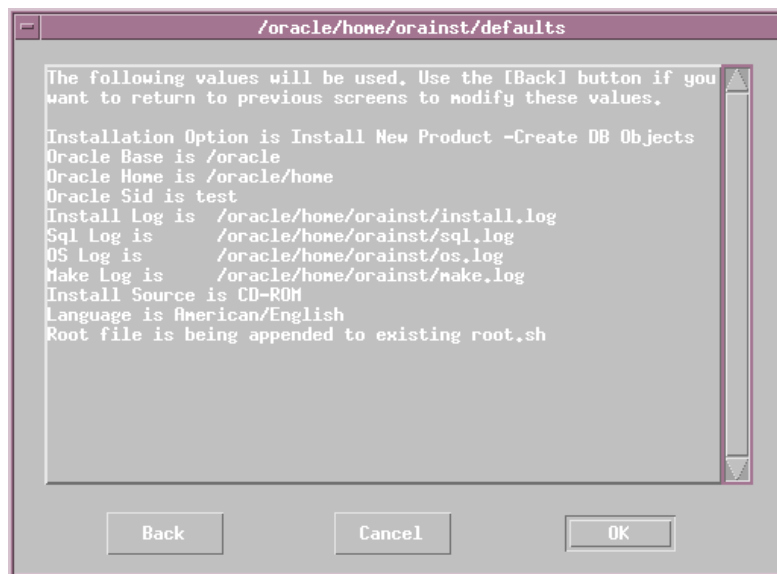


Figure 67. Oracle Installation Defaults

Select OK if you accept these settings.

21. The Software Asset Manager window appears.

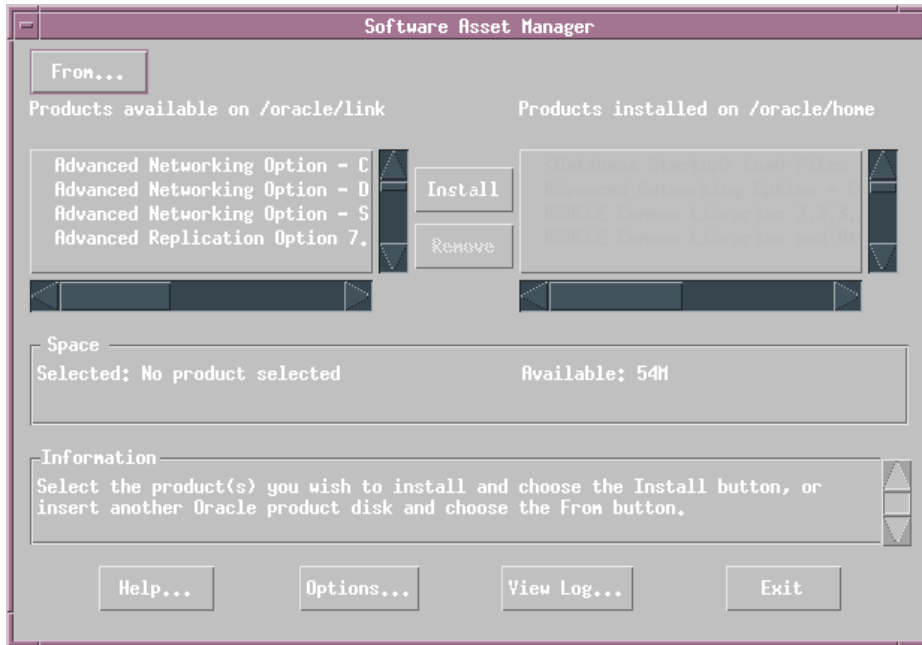


Figure 68. Software Asset Manager - Database Server

The Software Asset Manager window allows you to select the Oracle products that you want to install. The size of the distribution you selected and the space available in the destination directory is displayed.

Select the products you want to install. We selected the following:

- Oracle Server (RDBMS)
- SQL*Net
- SQL*Plus

22. During installation, you are prompted for the group that will act as the dba. We selected dba because dba is the group we specified in the beginning of our installation process. Our oracle user belongs to this group.

23. You are prompted for the storage type that you want to use, file system or raw device. We selected file system.

24. When asked whether or not to distribute over three mount points, we selected no, because one mount point is enough in our setup.

/oracle/home

25. Next you will be prompted to select the character set, US7ASCII is the default. If you want to use a character set other than US7ASCII, you must set the ORA_NLS32 variable. We selected the default.
26. Set the password for the system user on Oracle.
27. Set the password for the sys user.
28. Set internal user passwords for oper and oracle.
29. Set the password for your TNS listener.
30. We selected yes to whether we wanted our SQL*Net listener started automatically.
31. Enter your database domain name. We set our domain name to test_domain.
32. Confirm your installation settings and start the Installation.
33. After the Installation process is complete, you can exit the Oracle installation window.
34. Log in as root and run \$ORACLE_HOME/orainst/root.sh script. This script will set necessary file permissions.

```
# cd /oracle/home/orainst
# ./root.sh
```

Be sure to enter the dot, but no space, before the path.
35. While running the root.sh script the following message is displayed if you run this script from a directory other than the \$ORACLE_HOME directory.

```
ORACLE_HOME is not the same as home for this user continue (Y,N)
```

If you select Yes, the root.sh script continues with the ORACLE_HOME environment variable you specified.
36. Edit the /etc/oratab file so that the database will be started at system restart. This file contains the databases that will be started when the dbstart command are being run.
Figure 69 on page 81 shows the file in our example.

```

# This file is used by ORACLE utilities.  It is created by root.sh
# and updated by the ORACLE7 and SQL*Net install procedures.
#
# A colon, ':', is used as the field terminator.  A new line terminates
# the entry.  Lines beginning with a pound sign, '#', are comments.
#
# Entries are of the form:
#   $ORACLE_SID:$ORACLE_HOME:<N|Y>:
#
# The first and second fields are the system identifier and home
# directory of the database respectively.  The third field indicates
# to the dbstart utility that the database should, "Y", or should not,
# "N", be brought up at system boot time.
#
# Multiple entries with the same $ORACLE_SID are not allowed.
#
#
*/oracle/home:N
test:/oracle/home:Y

```

Figure 69. Oracle Table File

37. To start the database automatically at system start-up, run the following command:

```
# mkitab "oracle:2:wait:/bin/su oracle -c $ORACLE_HOME/bin/dbstart"
```

This AIX command makes an entry in the /etc/inittab file, and the init process starts the test database at system start up.

38. To verify that the installation completed successfully, follow the following steps:

1. Switch to user oracle

```
# su - oracle
```

2. Start the database engine:

```
$/oracle/home/bin/dbstart
```

Figure 70 on page 82 shows the output from the dbstart command.

```

oracle:/home/oracle >>dbstart
Oracle Server Manager Release 2.3.3.0.0 - Production
Copyright (c) Oracle Corporation 1994, 1995. All rights reserved.
Oracle7 Server Release 7.3.3.0.0 - Production Release
PL/SQL Release 2.3.3.0.0 - Production
SVRMGR> Connected to an idle instance.
SVRMGR> ORACLE instance started.
Total System Global Area      5013164 bytes
Fixed Size                    38988 bytes
Variable Size                  4138592 bytes
Database Buffers               819200 bytes
Redo Buffers                   16384 bytes
Database mounted.
Database opened.
SVRMGR>
Server Manager complete.
Database "test" warm started.

```

Figure 70. Oracle Database Start Output

39.To stop the database, you can run the following command:

```
$ /oracle/home/bin/dbshut
```

Figure 71 on page 82 shows the output from the dbshut command.

```

oracle:/home/oracle >>dbshut
Oracle Server Manager Release 2.3.3.0.0 - Production
Copyright (c) Oracle Corporation 1994, 1995. All rights reserved.
Oracle7 Server Release 7.3.3.0.0 - Production Release
PL/SQL Release 2.3.3.0.0 - Production
SVRMGR> Connected.
SVRMGR> Database closed.
Database dismounted.
ORACLE instance shut down.
SVRMGR>
Server Manager complete.
Database "test" shut down.
oracle:/home/oracle >>

```

Figure 71. Oracle Database Shutdown Output

40.To have the database and system shutdown in the same process, you can do the following.

1. Log in as root and create a script that you run instead of shutdown command to shutdown your machine. In our example, we created a script called shut. Figure 72 on page 83 shows the shut script.

```
su oracle -c /oracle/home/bin/dbshut
/etc/shutdown \S*
[Ctrl]-D
```

Figure 72. System Shutdown Script

2. After the shutdown script is created, change the permissions of the file so that shut is executable:

```
# chmod u+x shut
```

3. The following command will stop the database and shutdown your system.

```
# shut -h now
```

3.2.2 Installing Oracle for AIX Client

Below are the steps we used to install the Oracle for AIX client:

1. To install the client, you follow the same steps as installing the Oracle Server, up to the point where you are prompted for your installation options.

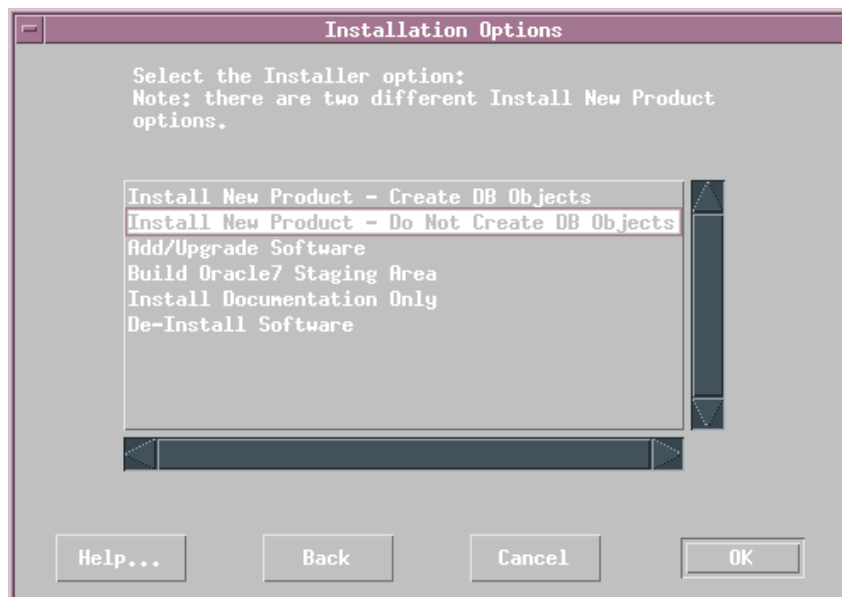


Figure 73. Oracle Client Installation Options

You select Install New Product - Do Not Create DB Object, that is, without creating a database.

2. In the next step, the same as in Installing the Oracle Server, you need to select the Oracle environment variables to use.

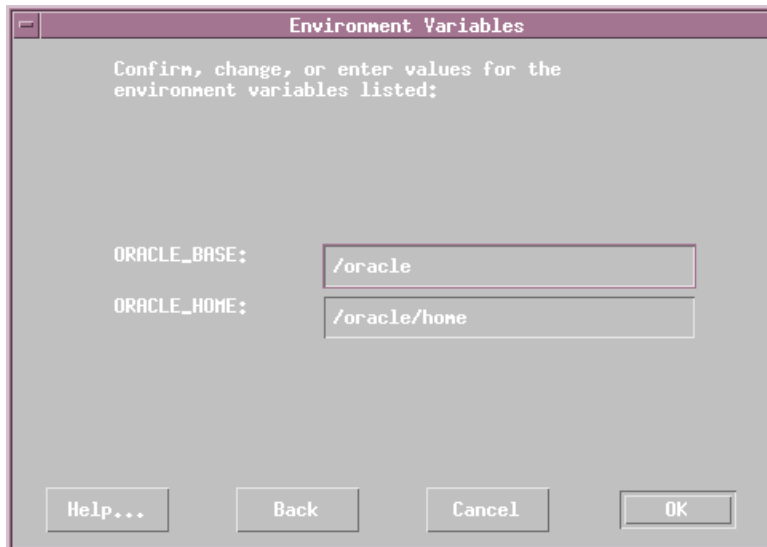


Figure 74. Oracle Environment Variables - Do Not Create DB Object

The Oracle environment variables are:

ORACLE_BASE defines the base of the directory structure for the Oracle Installation.

ORACLE_HOME defines the directory where your Oracle software is installed.

3. The next prompt asks you if you have run the rootpre.sh script before starting the Oracle Installer. Select **yes**, if you have.
4. You are prompted whether or not you would like to re-link the Oracle product executables. We selected yes. See the reasons explained in the steps on Installing the Oracle Server.
5. Next you are given a list of values that will be used for installation. If the values are set correctly, press **OK**.

The Software Asset Manager window appears (see in Figure 75 on page 85) and we selected the following Oracle components:

SQL*Net

SQL*Plus
TCP/IP Protocol Adapter

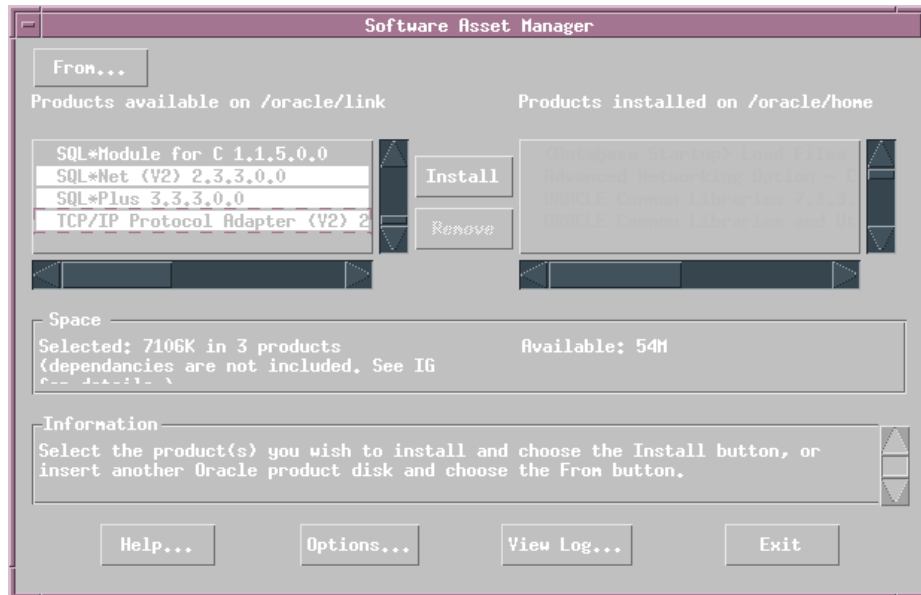


Figure 75. Software Asset Manager - Database Client

6. After installation finishes, exit the Oracle Installer. Change user id to root and run the script in the oracle home directory called db_setup.sh to set up the Oracle environment for root.
7. After the db_setup.sh is run, run a script called root.sh in the \$ORACLE_HOME/orainst directory as the root user to set the necessary file permissions for the Oracle products.

```
# . /home/oracle/db_setup.sh
# cd $ORACLE_HOME/orainst
# . ./root.sh
```

Be sure to enter the dot, space, and dot before the path.

A message is displayed by this script if you run this script from anywhere other than \$ORACLE_HOME directory.

```
ORACLE_HOME is not the same as home for the user continue (Y,N)
```

You can respond with Y (for yes).

3.3 Installing Oracle 7.3.3 for Windows NT

In this section, we install Oracle 7.3.3 for Windows NT, both the server and the client software. We also configure the Oracle client to communicate with the Oracle server. Finally, we create a RIM object that allows a Tivoli application to access the Oracle database via the TIM.

In our Oracle for NT setup, the TMR server is toronto, the RIM host is sydney, and the Oracle RDBMS server is london.

3.3.1 Required Files for RIM from Oracle for NT

The RIM support for Oracle for Windows NT is linked against `orant71.dll`, which is the Dynamic Link Library (DLL) supplied with Oracle 7.1. For RIM support for Oracle for Windows NT to work, this file must be installed and must be in the local path. Unfortunately, the name of this library was changed in Oracle 7.2 and Oracle 7.3, and `orant71.dll` does not get installed with Oracle 7.2 or Oracle 7.3. However, `orant71.dll` is available on the Oracle 7.2 and Oracle 7.3 installation media.

From the Oracle 7.3 media, you can copy the `ORANT71.DLL` from the `NT_X86\7\RSF2` directory to the `$ORACLE_HOME\bin` directory. Two other DLL modules, `CORENT23.DLL` and `MSVCRT10.DLL`, are required for `wrimtest` to work with Oracle 7.3. We copied these DLL modules from the Oracle 7.3.3 media after the Oracle 7.3.3 installation.

3.3.2 Installing Oracle for NT Server

Below are the steps we used to install the Oracle server on Windows NT.

1. Start Windows NT.
2. Log on to the NT Server as a user in the Administrators group.
3. Load the Oracle Server for Windows NT CD-ROM.
4. Verify that you can access the files on the CD-ROM.
5. You can use one of two methods to start the Oracle Installer.
 1. You can use the Windows NT Explorer to go to the drive that you selected to be your CD-ROM drive, and then go into the following directory, `<DRIVE>:\NT_X86\INSTALL`. From this directory, you can run the `ORAINST.EXE` executable.
 2. You can also use a Command Prompt

```
C:\>G:  
G:\>CD \NT_X86\INSTALL
```

G:\NT_X86\INSTALL\>ORAINST.EXE

Either of these methods bring up the Oracle Installer.

6. Select the Language to use during installation. The NLS_LANG parameter is set during installation, but can be changed after installation.
7. Enter your Company name.
8. Specify your Oracle home directory. The default directory is C:\ORANT. Oracle recommends that you use the default settings.

Note

During installation, Oracle Installer will detect previous versions of Oracle installed onto your server. Oracle will install new software in addition to previous Oracle software.

9. Figure 76 on page 88 displays the Oracle Software Asset Manager window.

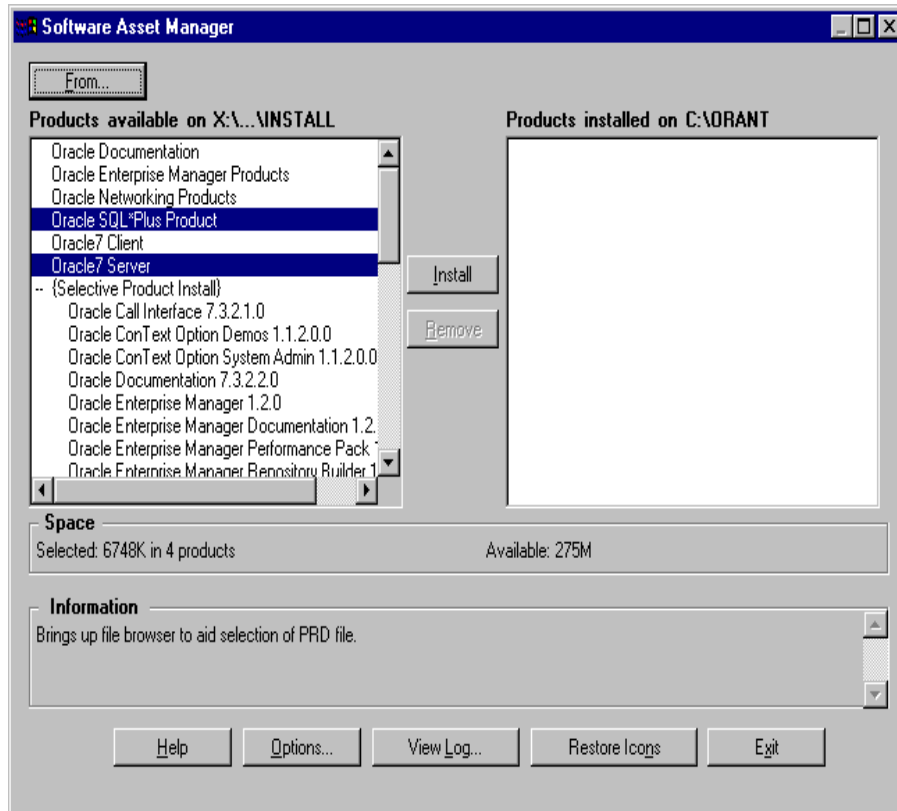


Figure 76. Oracle Asset Manager Window for Windows NT

10. Selected Oracle7 component(s) to install. The Oracle7 Server is a predefined collection of Oracle applications, that the server machine needs to function.

When you select **Oracle7 Server**, you get a list of Oracle server components to install:

- Oracle7 Server
- Oracle7 Server Options
- Oracle Webserver
- Oracle7 Utilities
- Oracle Enterprise Manager Web Administration Tools
- Oracle7 Performance Utility
- Oracle Call Interface

- Oracle Enterprise Manager
- Oracle Intelligent Agent
- SQL*Plus
- SQL*Net Server
- SQL*Net Protocol Adapter
- Oracle Installer

After you select Oracle 7 Server installation, the Oracle Installer displays a list of all the components that make up your Oracle Server selection. From this list, you can deselect the Oracle component(s) that you will not need. We deselected the following components because we do not need them in our setup.

- Oracle Webserver 1.0.2.00
- Oracle Enterprise Manager Web Admin Tools 7.3.2.2.0.

11. Next you are prompted for the server option that you are licensed to use. There are two options available.

1. Oracle 7 Spatial Data Option Product Modules

Spatial Data Option is an integrated set of functions that allow spacial data to be stored accessed and analyzed quickly and efficiently.

2. Oracle ConText Option

ConText Option enables text queries to be performed through SQL and PL/SQL from most Oracle interfaces.

We didn't select either of these options since we don't need them on our setup.

12. During installation, Oracle Installer automatically creates an Oracle starter database. You are prompted for the size of your starter database. We selected 20 MB. The database instance that is created during installation is ORCL.

Note

The starter database from Oracle7 Server for Windows NT contains:

- Four tablespaces
- Four database files
- Two log files
- A preloaded data dictionary
- Usernames

The data dictionary (SYSTEM tablespace) contains tables and views commonly used by many Oracle tools and sample programs.

13. Next you are prompted for the Starter database password. This password is for a user SYS, which has DBA roles for the database.
14. At the next prompt, you select the character set you want to use. We selected the default.
15. Enter the Protocol Adapter you are going to use for SQL*Net. We selected TCP/IP.
16. You are prompted whether or not you want to set up your host as Software Manager host. We selected no.
17. Next you are prompted for the directory name where the Oracle software is stored. The default is C:\ORAWIN. We accepted this value.

Note

If an error appears because the Oracle Installer cannot copy \NT_X86\INSTALL\OIUS.RES file to \ORANT\DBS\OIUS.RES, you can continue because this file (the Oracle Installer user interface) is in use and can be copied manually at the end of the installation.

18. During the configuration of the Enterprise Manager, the Oracle Installer cannot find the TOPOLOGY.ORA file. This file is used to discover and map network services. The Oracle Installer tries to create this file by using the TNSNAMES.ORA file. In our setup, since we were not connecting the Oracle server to any other Oracle servers, it was not necessary to create the TOPOLOGY.ORA file.
19. You can boot Windows NT so that the Oracle services are started. These services are:

- OracleServiceORCL
- OracleTNSListener
- OracleStartORCL

You can start these services manually using Windows NT Control Panel Services. If these services are started manually, the OracleServiceORCL service must be started before the OracleTNSListener.

Note

After the installation has finished, remember to copy the OIUS.RES file to your C:\ORANT\DBS\OIUS.RES from your installation media. If you don't do this, you will have problems running some Oracle applications

The installation of the Oracle server is complete.

3.3.3 Installing Oracle for NT Client

Below are the steps to install the Oracle client on Windows NT:

1. Insert CD-ROM.
2. Run ORAINST.EXE from the \NT_X86\INSTALL directory on the CD-ROM.
3. Select language setup.
4. Specify your company name.
5. From the Software Asset Manager of the Oracle Installer, we only selected Oracle 7 Client.
6. Next, the Oracle 7 Client Option dialog box displays the type of users that can be installed. If you choose Database Administrator. This option installs additional software on the client that allows the DBA to do administrative tasks on the client.
7. After the previous step, a dialog box appears with a list of products that are installed for either the Database Administrator or Application User. We selected the Database Administrator (the default).
8. Select the protocol adapter that you are going to use. We used TCP/IP as the SQL*Net adapter.

Note

The same error message appears (as in the Server installation) when the Oracle Installer tries to copy the OIUS.RES file to C:\ORANT\DBS directory. Remember to copy this file after the installation has finished.

9. The Enterprise Manager displays the message that it cannot find the TOPOLOGY.ORA file, and that it will look for the TNSNAMES.ORA file.

We create the TNSNAMES.ORA file after the client installation.

3.4 SQL*Net Overview

SQL*Net is the Oracle component that allows an Oracle client to communicate with an Oracle server on another machine. First, you need to make sure that SQL*Net is installed. The best way to do this is to run the Oracle installation program (oraInst for UNIX) and see if it is listed in the installed window in the installation dialog. If it isn't, install it on both the server and the RIM host. There is a tool called SQL*Net Easy Configuration on Windows NT to create the tnsnames.ora, but it doesn't exist on UNIX. You can use the SQL*Net Easy Configuration tool on Windows NT and then copy the appropriate files to your UNIX systems.

3.4.1 Creating TNSNAMES.ORA File Entry

The tnsname.ora file is the SQL*Net configuration file that tells the remote clients how to find servers on the network. The entries in this file consist of stanza labels (which we will refer to as server names or server IDs) that specify the information necessary to connect to a remote server. The file can contain multiple entries for different servers. The tnsnames.ora file is located in the \$ORACLE_HOME/network/admin directory.

You need to know the following information to set up a correct tnsnames.ora file:

- Host name of the database server
- Port number (/etc/services entry) to use for SQL*Net
- Oracle SID for database

Figure 77 on page 93 shows the format of the tnsnames.ora file. Items in angle brackets are things you need to fill in. This example assumes that you are using TCP/IP.


```

<Oracle service name> =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        [(COMMUNITY = <community name>)]
        (PROTOCOL = TCP)
        (Host = <database server host name>)
        (Port = <port number>)
      )
    )
    (CONNECT_DATA =
      (SID = <Oracle SID>)
      (GLOBAL_NAME = <Oracle service name>)
    )
  )

```

Figure 77. TNSNAMES.ORA File Format

The Oracle service name is arbitrary, but usually takes the format "name".world. What you can do is use either the database server host name or the SID for the name. So, for example, an Oracle database is running on a machine called pepper has the Oracle service name pepper.world.

Note

Oracle for NT does not use the ".world" suffix.

The Oracle service name is the value used in the Oracle environmental variable TWO_TASK and is the value of the Server ID in the RIM object.

The port number is usually 1521 on UNIX systems and 1526 on Windows NT systems. The way to make sure that this port value is the port value being used by the TNS listener is to look for the Port value in the listener.ora file in the \$ORACLE_HOME/network/admin directory on the RDBMS host or execute the `lsnrctl status` command on the RDBMS host.

RIM does not use the `/etc/services` file to select or verify the port usage, but for good UNIX systems administration, update the `/etc/services` file. An entry in `/etc/services` file may prevent another service from attempting to use the port or being defined to use that port.

Oracle recommends that you reserve a port for your SQL*Net listener in the `/etc/services` file of each node on the network that defines the SQL*Net listener port. The port is commonly 1521 for UNIX. The entry should list the listener name and the port number, for example:

```
<listener> 1521/tcp
```

where <listener> is the name of the listener, as defined in the listener.ora file.

You must reserve more than one port to start more than one listener.

You can display the status of the SQL*Net listener using the `lsnrctl status` command.

```
[root@panda]/> lsnrctl status

LSNRCTL for IBM/AIX RISC System/6000: Version 2.3.3.0.0 -
                Production 02-APR-98 10:52:15

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

Connecting to (ADDRESS=(COMMUNITY=tcp.world)(PROTOCOL=TCP)
                (Host=panda)(Port=1526))

STATUS of the LISTENER
-----
Alias                LISTENER
Version              TNSLSNR for IBM/AIX RISC System/6000:
                    Version 2.3.3.0.0 - Production
Start Date           13-MAR-98 11:47:11
Uptime               19 days 23 hr. 5 min. 5 sec
Trace Level          off
Security             OFF
SNMP                 OFF
Listener Parameter File
                    /usr/local/oracle/7.3.3/network/admin/listener.ora
Listener Log File
                    /usr/local/oracle/7.3.3/network/log/listener.log
Services Summary...
   sid1              has 1 service handler(s)
The command completed successfully

[root@panda]/>
```

Figure 78. Oracle SQL*Net Listener Status

In our setup, the database server host name is panda, the Oracle SID is sid1, and the SQL*Net listener port is 1526.

Figure 79 on page 95 is an example of a tnsnames.ora file on the RIM host.

```

spice.world =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (COMMUNITY = tcp.world)
        (PROTOCOL = TCP)
        (Host = panda)
        (Port = 1521)
      )
    )
    (CONNECT_DATA =
      (SID = sid1)
      (GLOBAL_NAME = spice.world)
    )
  )
)

```

Figure 79. TNSNAMES.ORA File - RIM Host

Where the database server host name is panda, the Oracle SID is sid1, and the SQL*Net listener port is 1521.

3.4.2 Oracle SQL*Net Listener

Since the Oracle SQL*Net listener process must be running for SQL*Net to work, you need to insure that the listener is running on the Oracle database server. The Oracle SQL*Net listener must be started prior to starting the Oracle database server. To see if the listener is running on UNIX, execute the following at the UNIX command prompt:

```
ps -ef | grep lsnrctl
```

You should see an entry `$ORACLE_HOME/bin/lsnrctl <some_name> -inherit`. If this process is not running, you can start it by executing the command:

```
lsnrctl start
```

This command starts the listener with the default parameters. These default parameters can be changed by creating an entry in the listener control file, `listener.ora`, and supplying the name of the entry as an argument to the `lsnrctl start` command.

```
lsnrctl start [<listener_name>]
```

Since the Oracle SQL*Net listener must be started prior to starting the Oracle database for SQL*Net to work, if you don't see the `lsnrctl` process running:

1. Shut down the database. As the user `oracle`, execute `dbshut` immediately.

2. Start the Oracle SQL*Net listener. The command to use is `lsnrctl start [<listener_name>]`.
3. Start the Oracle database. The command to use is `dbstart`.

3.4.3 Determining Name of Current Listener

Figure 80 on page 96 shows how you can determine the name of the current SQL*Net listener.

```
[root@panda]/> lsnrctl show current_listener

LSNRCTL for IBM/AIX RISC System/6000: Version 2.3.3.0.0 -
                Production on 14-APR-98 15:21:41

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

Current Listener is LISTENER
[root@panda]/>
```

*Figure 80. Show Current Oracle SQL*Net Listener*

The current SQL*Net listener on the RDBMS server is LISTENER, which is the label on a stanza in the listener.ora file in the \$ORACLE_HOME/network/admin directory.

3.4.4 Creating LISTENER.ORA File Entry

Figure 81 on page 97 shows the format of the listener.ora file in the \$ORACLE_HOME/network/admin directory.

```

LISTENER=
  (ADDRESS_LIST =
    (ADDRESS =
      (COMMUNITY = )
      (PROTOCOL = TCP)
      (HOST = )
      (PORT = )

    )

  )

STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10

SID_LIST_LISTENER=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=)
      (ORACLE_HOME=)
      (GLOBAL_DBNAME=)
    )
  )

TRACE_LEVEL_LISTENER = OFF
USE_CKPFIL_LISTENER = true
USE_PLUG_AND_PLAY_LISTENER = true
PASSWORDS_LISTENER = (9D3ABA0853BBAEC3)

```

Figure 81. LISTENER.ORA File Format

Figure 82 on page 97 is a TCP example of a listener.ora file.

```

LISTENER =
  (ADDRESS_LIST =
    (ADDRESS =
      (COMMUNITY = tcp.world)
      (PROTOCOL = TCP)
      (Host = panda )
      (Port = 1526)
    )
  )
  (SID_LIST_LISTENER =
    (SID_LIST =
      ( SID_DESC =
        (SID_NAME = sid1 )
        (ORACLE_HOME = /usr/local/oracle/7.3.3 )
      )
    )
  )

```

Figure 82. TCP Entry in LISTENER.ORA File

Figure 83 on page 98 is an example of an IPC entry in a listener.ora file.

```

LISTENER=
  (ADDRESS_LIST=
    (ADDRESS=
      (PROTOCOL=IPC)
      (KEY=sid1)
    )
  )

SID_LIST_LISTENER=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=sid1)
      (ORACLE_HOME=/oracle/home)
      (GLOBAL_DBNAME=sid1.sid1_domain)
    )
  )

```

Figure 83. IPC Entry in LISTENER.ORA File

The default listener.ora file created by the Oracle for AIX install process has an IPC entry.

3.4.5 SQL*Net Listener Log File

The SQL*Net listener log file contains entries for all activity through the listener.

```

10-APR-98 16:06:26 * (CONNECT_DATA=(SID=sid1)
(GLOBAL_NAME=tcp.world)(CID=(PROGRAM=)(HOST=panda)(USER=nobody)))
* (ADDRESS=(PROTOCOL=tcp)(HOST=9.3.1.47)(PORT=4529))
* establish * sid1 * 0

10-APR-98 16:06:38 * (CONNECT_DATA=(SID=sid1)
(GLOBAL_NAME=tcp.world)(CID=(PROGRAM=)(HOST=panda)(USER=nobody)))
* (ADDRESS=(PROTOCOL=tcp)(HOST=9.3.1.47)(PORT=4532))
* establish * sid1 * 0

10-APR-98 16:37:25 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=panda)
(USER=root))(COMMAND=status)(ARGUMENTS=64)(SERVICE=LISTENER)(VERSION=36712448))
* status * 0

10-APR-98 16:39:18 * (CONNECT_DATA=(CID=(PROGRAM=)(HOST=panda)
(USER=root))(COMMAND=status)(ARGUMENTS=64)(SERVICE=LISTENER)(VERSION=36712448))
* status * 0

```

Figure 84. SQL*Net Listener Log File

Figure 84 on page 98 shows a part of a SQL*Net Listener log file.

3.5 Configuring Oracle for AIX Connectivity

Since Oracle uses SQL*Net to establish connection between the client and server, we must configure SQL*Net on both the server and the client.

3.5.1 Configuring SQL*Net Server for AIX

The TNS listener is a process that runs on the Oracle server and waits for any incoming SQL*Net requests from Oracle clients.

1. You may need to set up the listener.ora configuration file which is read by the TNS listener process when it is started. This file is automatically created during installation when you select to create a database object. Figure 85 on page 99 is what our entry looked like after we made the changes.

```
LISTENER=
  (ADDRESS_LIST=
    (ADDRESS=
      (PROTOCOL=TCP)
      (KEY=test)
      (HOST=capetown)
      (PORT=1526)
    )
  )
STARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
SID_LIST_LISTENER=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=test)
      (ORACLE_HOME=/oracle/home)
      (GLOBAL_DBNAME=test.test_domain))
    )
TRACE_LEVEL_LISTENER = OFF
USE_CKPFIL_LISTENER = true
```

Figure 85. Oracle for AIX LISTENER.ORA File - Database Server

We changed the PROTOCOL entry from IPC to TCP. Since the PROTOCOL is TCP, we added two entries, HOST and PORT, that were not there.

We set the PORT value to 1526 because this is the default port that is used by Oracle. If you look in the /etc/services file, you can see that this port is reserved for the Oracle SQL*Net Listener process.

To start the TNS Listener on AIX server, you log in as the user oracle and use the `lsnrctl` command:

```
$ lsnrctl start test
```

Figure 86 on page 100 shows the output of this command.

```
Starting /oracle/home/bin/tnslsnr: please wait...

TNSLSNR for IBM/AIX RISC System/6000: Version 2.3.3.0.0 - Production
System parameter file is /oracle/home/network/admin/listener.ora
Log messages written to /oracle/home/network/log/listener.log
Listening on: (ADDRESS=(PROTOCOL=tcp)(DEV=10)(HOST=9.3.1.39)(PORT=1526))
Connecting to (ADDRESS=(PROTOCOL=TCP)(KEY=test)(HOST=capetown)(PORT=1526))

STATUS of the LISTENER
-----
Alias                LISTENER
Version              TNSLSNR for IBM/AIX RISC System/6000: Version 2.3.3.0.
0 - Production
Start Date           16-APR-98 17:05:29
Uptime               0 days 0 hr. 0 min. 2 sec
Trace Level          off
Security             ON
SNMP                 OFF
Listener Parameter File /oracle/home/network/admin/listener.ora
Listener Log File    /oracle/home/network/log/listener.log

Services Summary...
  test                has 1 service handler(s)

The command completed successfully

oracle:/oracle/home/network/admin >>
```

Figure 86. Oracle Start Listener Output

When the listener starts, you see from the output from the TCP/IP port that the listener uses to listen for any incoming SQL*Net requests. Make sure that the same port number to connect to the server is specified in the `/ORACLE_HOME/network/admin/tnsnames.ora` file on the client.

After the Oracle server processes are started and you are sure that your network is set up correctly, you can now log on to the client machine as the user `oracle`, and test the connection by using any of the following commands to see that a connection can be established with the server.

3.5.2 Configuring SQL*Net Client for AIX

SQL*Net uses two configuration files on the client to connect to the server and enable the workstation to access, modify, share and store software on the Oracle server. These two files are `sqlnet.ora` and `tnsnames.ora`. Both these files reside `$ORACLE_HOME/network/admin` directory.

3.5.2.1 Oracle for AIX sqlnet.ora File

The sqlnet.ora file contains tracing diagnostic parameters for the client, and specifies diagnostics and client information about Oracle Names servers.

```
TRACE_LEVEL_CLIENT = OFF
names.default_domain = world
name.default_zone = world
```

Figure 87. Oracle for AIX SQLNET.ORA File - Database Client

The default domain is world.

3.5.2.2 Oracle for AIX tnsnames.ora File

The tnsnames.ora file can be created manually, or it can be copied from any Oracle client on the network. The values must then be changed according to your specifications. We copied these files from our Oracle for NT client, and then changed the values to what we needed.

```
red.world =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (COMMUNITY = tcp.world)
        (PROTOCOL = TCP)
        (Host = capetown)
        (Port = 1526)
      )
    )
    (CONNECT_DATA = (SID = test)
  )
  )
~
```

Figure 88. Oracle for AIX TNSNAMES.ORA - Database Client

SQL*Net uses aliases to connect to an instance on the Oracle server. This file contains an entry for every alias that is used. At this stage in our setup, we have only one alias called red that is pointing to the database instance, test, on the server. The .world extension specifies the domain. The Oracle default domain is world.

COMMUNITY specifies a network community which is a group of nodes using TNS-based software that communicates using the same transport protocol. A common convention is to include the underlying protocol, such as TCP, in the name and domain of this specific service. The default domain used by Oracle is world.

PROTOCOL specifies the transport protocol adapter used for connection to the Oracle server.

HOST specifies the hostname or host IP address of the Oracle server.

PORT is the TCP/IP port, either the number or the name specified in the /etc/services file on the server that the TNS listener process is using to connect to the client. Make sure that the PORT value corresponds to the Oracle server. You will have problems connecting to the Oracle server if these values are different.

3.6 Configuring Oracle for NT Connectivity

Before trying to connect to the Oracle server from the Oracle client, make sure that your network is configured so that the Oracle client can connect to the Oracle server over TCP/IP.

3.6.1 Configuring SQL*Net Server for NT

You can insure that the Oracle services are started on the Oracle server by clicking on the **Windows NT Start** button on the Taskbar and select => **Settings => Control Panel**. From the Control Panel, select **Services**, and scroll down until you see the Oracle services. The status must be Started.

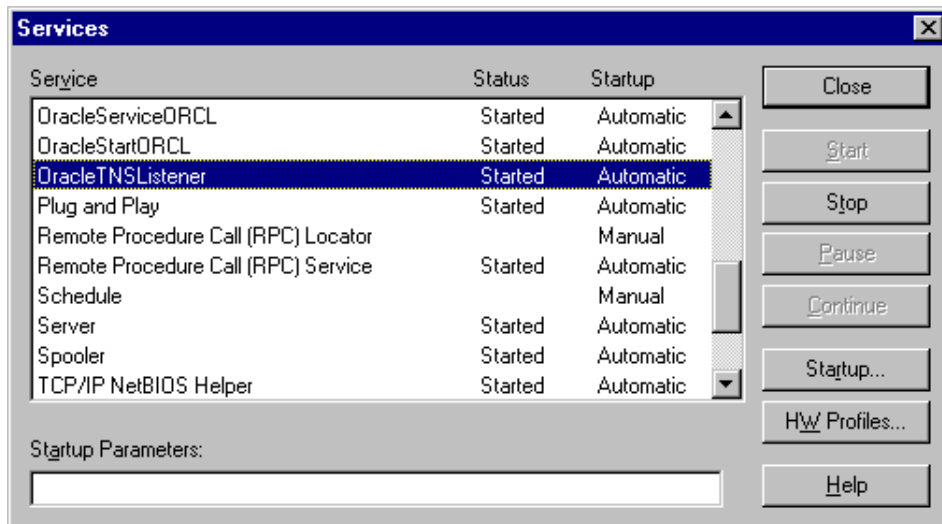


Figure 89. Windows NT Control Panel Services - Oracle

The Windows NT Services that must be started on the Oracle server are OracleService<database_name>, OracleStart<database_name>, and OracleTNSListener. In our setup, <database_name> is ORCL.

3.6.2 Configuring SQL*Net Client for NT

On the client, you need two files in the \ORANT\NETWORK\ADMIN directory, sqlnet.ora and tnsnames.ora. These files are used by SQL*Net on the Oracle client to connect to Oracle server

3.6.2.1 Oracle for NT Client SQLNET.ORA File

The sqlnet.ora file is created during the Oracle for NT client installation.

```
TRACE_LEVEL_CLIENT = OFF

names.directory_path = (TNSNAMES)
names.default_domain = world
name.default_zone = world
```

Figure 90. Oracle for NT SQLNET.ORA File

3.6.2.2 Oracle for NT Client TNSNAMES.ORA File

The tnsnames.ora file is not created by default when you install the Oracle client. You can copy this file from another Oracle client machine that is

connecting to the same Server, or you can create the file using the SQL*Net Easy Configuration utility.

You start the SQL*Net Easy Configuration utility by selecting => **Windows NT Start => Programs => Oracle For Windows NT => SQL Net Easy Configuration.**

This will start up the Oracle Installer process and displays the SQL*Net Easy Configuration window. See Figure 91 on page 104.

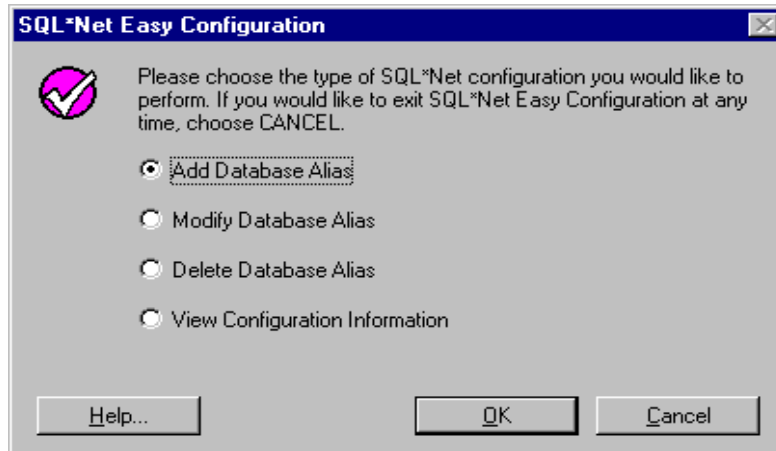


Figure 91. SQL*Net Easy Configuration - Add Database Alias

We selected Add Database Alias.

1. You are prompted for a database alias name. This is the name you will use on the Oracle client to connect to the database on the Oracle server. In our setup, we created an alias for the default database ORCL. We used GREEN as our alias name.
2. Now you are prompted for the protocol to be used. We selected TCP/IP in our setup.
3. Next you are prompted for the host name of the database server and the Oracle database instance name on the Oracle server (see Figure 92 on page 105). In our setup, the host name of the database server is london and the Oracle database instance is ORCL.

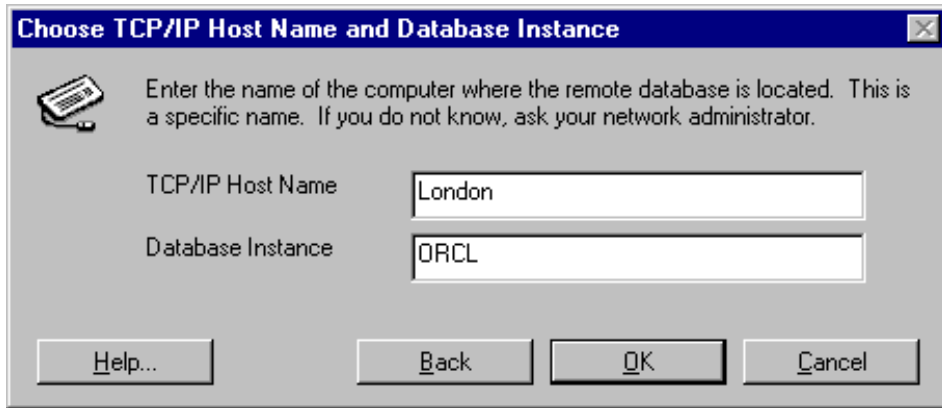


Figure 92. Choose TCP/IP Host Name and Database Instance

4. On the next screen (see Figure 93 on page 105), you are prompted to select yes, if the values are correct.

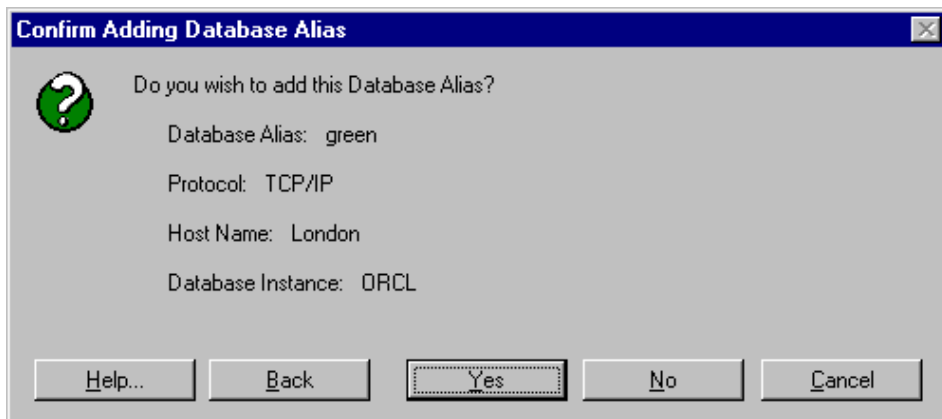


Figure 93. Confirm Adding Database Alias

When you select yes, the TNSNAMES.ORA file is created with a stanza entry (see Figure 94 on page 106) for the alias called GREEN that will point to the database instance ORCL on the Oracle server. In our setup, the database server is london.

```

#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.

green.world =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (COMMUNITY = tcp.world)
        (PROTOCOL = TCP)
        (Host = london)
        (Port = 1521)
      )
      (ADDRESS =
        (COMMUNITY = tcp.world)
        (PROTOCOL = TCP)
        (Host = london)
        (Port = 1526)
      )
    )
  )
  (CONNECT_DATA = (SID = ORCL)
)

```

Figure 94. Oracle for NT TNSNAMES.ORA - Database Client

After the tnsnames.ora and sqlnet.ora files are created, you can test the connection between the server and the client.

3.7 Connecting the Oracle Client and Server for AIX

There are different ways to verify that the client can communicate with the server. But before we try to connect to the server, let us first make sure all the right processes are running. Make sure that the database is started on the server. This procedure is discussed in “Installing Oracle for AIX Server” on page 68.

3.7.1 Using Oracle for AIX TNS Ping Utility

The TNS ping utility is used to determine whether or not you can reach a service on a SQL*Net network. The syntax of command is:

```
tnsping <address> [<count>]
```

where the address is the service name in the tnsnames.ora file and the count determines how many times the program attempts to reach the server. The default count is one.

Figure 95 on page 107 shows the output of the `tnsping` command if the TNS listener is not started on the server.

```
/oracle/home/network/admin >> tnsping red

TNS Ping Utility for IBM/AIX RISC System/6000: Version 2.3.3.0.0 - Production on
16-APR-98 17:41:25

Copyright (c) Oracle Corporation 1995. All rights reserved.

Attempting to contact (ADDRESS=(COMMUNITY=tcp.world)(PROTOCOL=TCP)(Host=capetown
)(Port=1526))

TNS-12541: TNS:no listener

/oracle/home/network/admin >>
```

Figure 95. Oracle TNS Ping Output - TNS Listener Not Started

If the TNS listener was started on the server, the output of the `tnsping` command looks like Figure 96 on page 107.

```
/oracle/home/network/admin >> tnsping red 5

TNS Ping Utility for IBM/AIX RISC System/6000: Version 2.3.3.0.0 - Production on
16-APR-98 17:49:00

Copyright (c) Oracle Corporation 1995. All rights reserved.

Attempting to contact (ADDRESS=(COMMUNITY=tcp.world)(PROTOCOL=TCP)(Host=capetown
)(Port=1526))

OK (20 msec)
OK (3860 msec)
OK (120 msec)
OK (110 msec)
OK (100 msec)

/oracle/home/network/admin >>
```

Figure 96. Oracle TNS Ping Output - TNS Listener Started

When you connect to the TNS service using `tnsping`, an estimate of the round trip time in milliseconds appears. If the `tnsping` fails, a network error message appears without the overhead of a database connect.

3.7.2 Using Oracle for AIX SQL*Plus Utility

SQL*Plus has a command language interface that allows you to use SQL database commands.

You connect to a database on the server from the client using the following command:

```
$ sqlplus sys/sys@red
```

We are connected as user sys and password sys using the alias connection setup of red. This is what the output will look like.

```
/home/oracle >> sqlplus sys/sys@red

SQL*Plus: Release 3.3.3.0.0 - Production on Fri Apr 17 11:05:19 1998

Copyright (c) Oracle Corporation 1979, 1996. All rights reserved.

Connected to:
Oracle7 Server Release 7.3.3.0.0 - Production Release
PL/SQL Release 2.3.3.0.0 - Production

SQL>
```

Figure 97. Using SQL*Plus on AIX.

From the SQL> prompt, you can execute SQL commands.

3.8 Connecting Oracle Server and Client for NT

There are different methods to test connectivity between the Oracle server and the Oracle client.

3.8.1 Using Oracle for NT TNS Ping Utility

The Oracle `tnsping` command is used from the Oracle client to contact the Oracle server.

From the Oracle client, we used `tnsping` to contact Oracle server ORCL instance five times using the alias name green.


```
C:\>tnsping green 5

TNS Ping Utility for 32-bit Windows:version 2.3.2.1.0 - Production on 21-APR-98
 16:14:45

Copyright , 1996<c> Oracle Corporation 1995. All rights reserved.

Attempting to connect <ADDRESS=<COMMUNITY=tcp.world><PROTOCOL=TCP><HOST=london><
Port=1521>>
OK <560 msec>
OK <50 msec>
OK <70 msec>
OK <60 msec>
OK <50 msec>

C:\>
```

Figure 98. TNS Ping Output - Oracle for NT Client

The `tnsping` command contacts the Oracle TNS listener service on the Oracle server.

3.8.2 Using Oracle for NT SQL*Plus Utility

SQL*Plus is used to connect to the Oracle server and then you can execute SQL commands.

To start SQL*Plus, select => **Windows NT Start => Programs => Oracle for Windows NT => SQL Plus 3.3.**

This will display a log on window. See Figure 99 on page 110.

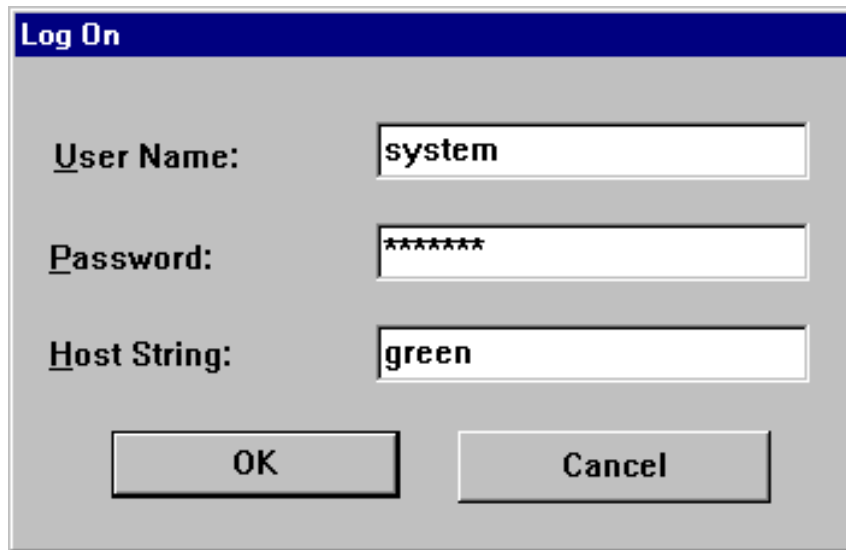


Figure 99. SQL*Plus Log On

You are prompted for information to connect to the Oracle database as setup by the alias.

User Name is the user name used to connect to the database. In our setup, this user name is system.

Password is the password of the user system. The default password is manager.

Host String is the alias name for the Oracle starter database ORCL on the Oracle server created with SQL NET Easy Configuration utility. In our setup, the alias name is green.

Figure 100 on page 111 shows the successful connection from the Oracle client to the Oracle server.

```

SQL*Plus: Release 3.3.2.0.2 - Production on Tue Apr 21 17:28:05

Copyright (c) Oracle Corporation 1979, 1994. All rights reserve
Connected to:

Oracle7 Server Release 7.3.2.2.0 - Production Release

With the distributed, replication and parallel query options
PL/SQL Release 2.3.2.2.0 - Production

SQL>

```

Figure 100. SQL*Plus - Oracle for NT Client

When you have connected to the ORCL instance on the Oracle server, you can execute SQL commands to query and update the database.

3.9 Creating RIM Object to Access Oracle Database

After you set up the Oracle server and client so that the client can connect to the server create a RIM object to allow a Tivoli application to connect to the Oracle server through the Tivoli RIM host.

The command used to create a RIM object is `wcrtrim`. The RIM object is created using a Tivoli administrator that has the appropriate roles. By default, the UNIX root user has the appropriate roles.

3.9.1 Oracle RIM Object

Table 6 on page 111 shows the Oracle-specific values that go into the RIM object.

Table 6. Oracle RIM Attributes

Parameter	Value	Description
RDBMS Vendor	Oracle	Database vendor name - must be Oracle.
RIM Host	RIM Host Name	The machine where the RIM server will run.
Database ID	ORACLE_SID	The name of the Oracle database instance, or Oracle System ID. This value can be found in the <code>tnsnames.ora</code> file in the <code>SID=</code> parameter in the <code>CONNECT_DATA=</code> statement.

Parameter	Value	Description
Database Home	ORACLE_HOME	The directory where Oracle software is installed. The tnsnames.ora file is in this directory plus the path elements network/admin.
Server ID	TWO_TASK	The Oracle listener service name and is the service name label on the stanza that defines the database in the \$ORACLE_HOME/network/admin/tnsnames.ora file on the RIM host.
RDBMS User Name	Oracle User ID	The RDBMS user that owns the application's data space, by default this is "tivoli" for Inventory and "tec" for TEC. For Windows NT, this value must take the form userid@server_id where server_id is the RIM object value for Server ID. See description for Server ID above.
RDBMS User Password		When the RIM object is created, you are prompted for the RDBMS User password. See the description for RDBMS User Name above. If you change the RDBMS User password for the database product, you must change the password stored in the RIM object with the <code>wsetrimpw</code> command.

ORACLE_SID, ORACLE_HOME, and TWO_TASK are operating system environmental variables.

3.9.2 Creating RIM Object for AIX RIM Host

We created a RIM object called `testeraix` where the vendor is Oracle, the `host_name` of the RIM Host is `dublin`, the database instance name is `test`, the user name used to connect to the database is `sys`, the `rdbms_home` directory name where Oracle is installed is `/oracle/home`, the `server_id` alias used to connect to server is `red.world`, and the `rim_name` (RIM object name) is `testeraix`.

```
# wcrtrim -v oracle -h dublin -d test -u sys -H /oracle/home -s red.world testeraix
RDBMS password:
#
```

Figure 101. Create Oracle RIM Object for AIX RIM Host

NOTE

For Oracle for AIX, you must specify the server_id in the format alias_name.domain_name. In our setup, the domain name is world.

If you use the `-i` option, the `wcrtrim` command will not prompt you for the sys users password. The `-I` option is only used with DB2.

After the RIM object is created, we use the `wgetrim` command to display the current RIM object values.

```
# wgetrim testeraix

RIM Host:      dublin
RDBMS User:    sys
RDBMS Vendor:  Oracle
Database ID:   test
Database Home: /oracle/home
Server ID:     red.world
Instance Home:

#
```

Figure 102. Display Oracle RIM Object for AIX RIM Host

The Instance Home should be blank.

3.9.3 Creating RIM Object for NT RIM Host

We created a RIM object called `testernt`, where the vendor is Oracle, the host_name is `sydney`, the database is the database instance name, `ORCL`, the user is the user name that will connect to database, `system@green`, the `rdms_home` is the directory where Oracle is installed, `C:\orant`, and the `server_id` is the alias that is used to connect to the server `green`.

```
C:\> wcrtrim -v oracle -h sydney -d ORCL -u system@green -H c:\orant -s green testernt

RDBMS password:

C:\>
```

Figure 103. Create Oracle RIM Object for NT RIM Host

Note

With Oracle for NT, you must specify the user name in the format `user_name@alias_name` , `system@green` in our setup.

When the `wcrtrim` command is executed without the `-i` option, you are prompted for the password of the database user that will connect to the database. In our setup, the password of user `system` is `manager` (the default).

After you have created your RIM object you can look at the values that you have entered by using the `wgetrim` command.

```
C:\>wgetrim testernt

RIM Host:      sydney
RDBMS User:    system@green
RDBMS Vendor:  Oracle
Database ID:   ORCL
Database Home: c:\orant
Server ID:     green
Instance Home:

C:\>
```

Figure 104. Display RIM Object - NT RIM Host

The Instance Home should be blank.

3.9.4 Testing RIM Object

After you have a connection, you can use `wrimtest` options to interact with the database. Use the `?` option at the prompt to see a list of options that are available for `wrimtest`. After you have created the RIM object you can use the `wrimtest` command to see if RIM can connect to the RDBMS Server.

We used the `wrimtest` command from the TMR server (`chicago`) and the RIM object (`testeraix`) to connect to the Oracle for AIX database (`test`) on the database server (`capetown`) through the RIM host (`dublin`).

```

root@chicago[/#]# wrimtest -l testeraix

Resource Type : RIM
Resource Label : testeraix
Host Name      : dublin
User Name      : sys
Vendor         : Oracle
Database       : test
Database Home  : /oracle/home
Server ID      : red.world
Opening Regular Session ..Opened
RIM : Enter Option > x
Releasing session

root@chicago[/#]#

```

Figure 105. Oracle for AIX RIM Test Connect Output

The database server (capetown) is found using the alias definition (red.world) in tnsnames.ora file on the RIM host (dublin).

We used the `wrimtest` command from the TMR server (toronto) and the RIM object (testernt) to connect to the Oracle for NT database (ORCL) on the database server (london) through the RIM host (sydney).

```

C:\TEMP>wrimtest -l testernt

Resource Type : RIM
Resource Label : testernt
Host Name      : sydney
User Name      : system@green
Vendor         : Oracle
Database       : ORCL
Database Home  : c:\orant
Server ID      : green
Instance Home  :
Opening Regular Session...Session Opened

RIM : Enter Option >

```

Figure 106. Oracle for NT RIM Test Connect Output

The database server (london) is found using the alias definition (green.world) in tnsnames.ora file on the RIM host (sydney).

You can use `?` at the prompt to list all the options available with `wrimtest`.

3.10 Configuring a Connection between NT and AIX

Now that we have Oracle and RIM working between like operating systems, set up cross-systems access. That is, we use NT RIM host to access Oracle for AIX server, and visa versa.

The first step is to set up SQL*Net so that the NT RIM host can connect to the Oracle for AIX server. Next, set up SQL*Net so that the AIX RIM host can connect to the Oracle for NT server.

3.10.1 Configuring SQL*Net for NT to Connect to AIX Server

To set up a SQL*Net connection from a Oracle for NT client to an Oracle for AIX server, create a new alias on the Oracle for NT client pointing to the Oracle for AIX server.

There are two ways to create the alias. You can edit the tnsnames.ora file and then add an entry for the alias that is used to connect to AIX Oracle server, or you can use SQL*Net Easy Configuration utility.

We used SQL*Net Easy Configuration utility.

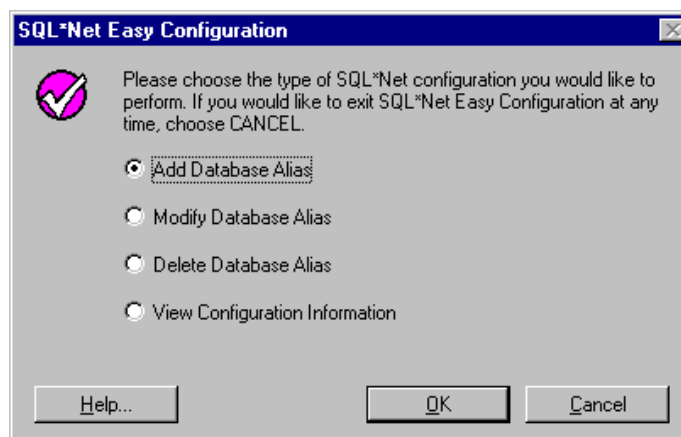


Figure 107. SQL Net Easy Configuration - Add Database Alias

From the SQL*Net Easy Configuration window, you select Add Database Alias.

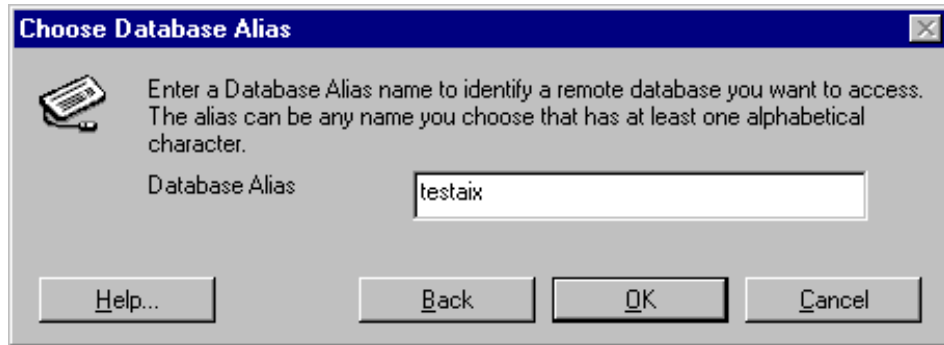


Figure 108. Choose Database Alias for AIX Server on NT Client

Then enter Database Alias name. We selected the alias testaix.

Next, you are prompted for the protocol used for the SQL*Net connection. We used TCP/IP.

Then you are prompted for the TCP/IP Host Name of the Oracle server where the Database Instance resides and the Oracle Database Instance name on the Oracle server.

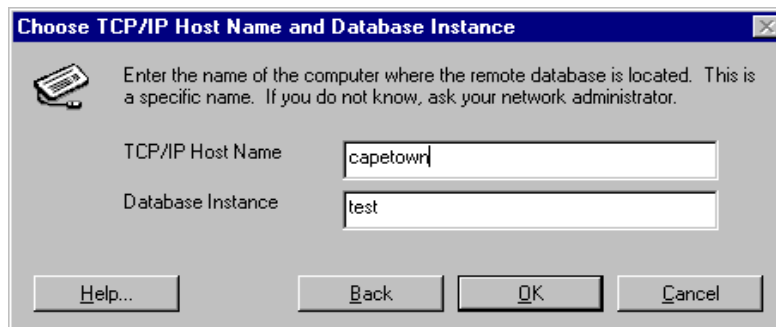


Figure 109. Choose Host Name & Database Instance for AIX Server on NT Client

When you select **OK**, a confirmation window is displayed. See Figure 110 on page 118.

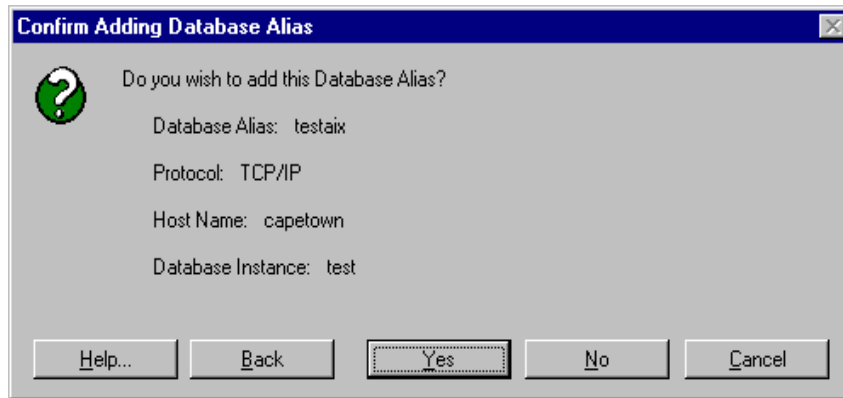


Figure 110. Confirm Adding Database Alias for AIX server on NT Client

If all the settings are correct, select **yes**, to add a stanza entry in the `tnsnames.ora` file for the alias `testaix` pointing to the instance `test` on the Oracle server `capetown`.

After the alias is created, you can test the connection between the Oracle for NT client and the Oracle for AIX server using the `tnsping` command or SQL*Plus, as we have done earlier.

3.10.2 Configuring SQL*Net for AIX to Connect to NT Server

Before creating a RIM object, you should verify that you can have a SQL*Net connection between an Oracle for AIX client and an Oracle for NT server.

On AIX, you must manually add an entry in the `tnsnames.ora` for the alias that is used by SQL*Net to connect to the Oracle server. This stanza for the alias can be inserted anywhere in the file. The alias we added was `nt`. As you can see, you must add the `.world` extension to the alias name to specify the domain.

The `nt.world` entry is similar to the other alias we created called `red`. The only differences are the `HOST = london` (the Oracle for NT server), and `SID = ORCL` (the Oracle database instance name on NT). Figure 111 on page 119 shows an extract from the `tnsnames.ora` file for this alias.

```

nt.world =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (COMMUNITY = tcp.world)
        (PROTOCOL = TCP)
        (Host = london)
        (Port = 1526)
      )
    )
    (CONNECT_DATA = (SID = ORCL)
  )
)

```

Figure 111. TNSNAMES.ORA Entry for NT Server on AIX Clients

The PORT number is used by SQL*Net to connect between the Oracle server and the client. The PORT for the alias must be the same as the PORT that the TNS listener is using on the Oracle for NT server, and the SID must be the database instance name on the Oracle for NT server.

After the alias is created, you can use either the `tnsping` command to see if you can contact the TNS listener or you use SQL*Plus to connect to the database instance.

3.10.3 Connecting Oracle AIX Client and NT Server

We used The Oracle TNS ping utility and the Oracle SQL*Plus utility to verify that the Oracle for AIX client can connect to the Oracle for NT server.

Figure 112 on page 120 shows the `tnsping` output from Oracle for AIX client to Oracle for NT server.

```

# tns ping nt 5

TNS Ping Utility for IBM/AIX RISC System/6000: Version 2.3.3.0.0 - Production on
28-APR-98 17:05:43

Copyright (c) Oracle Corporation 1995. All rights reserved.

Attempting to contact (ADDRESS=(COMMUNITY=tcp.world)(PROTOCOL=TCP)(Host=london)(
Port=1526))
OK (1290 msec)
OK (60 msec)
OK (60 msec)
OK (60 msec)
OK (30 msec)

#

```

Figure 112. TNS Ping From AIX Client To NT Server

Figure 113 on page 120 shows the SQL*Plus connection from the Oracle for AIX client to the Oracle for NT server.

```

# sqlplus system/manager@nt

SQL*Plus: Release 3.3.3.0.0 - Production on Tue Apr 28 17:08:47 1998

Copyright (c) Oracle Corporation 1979, 1996. All rights reserved.
Connected to:

Oracle7 Server Release 7.3.2.2.0 - Production Release
With the distributed, replication and parallel query options
PL/SQL Release 2.3.2.2.0 - Production

SQL>

```

Figure 113. SQL*Plus From AIX Client To NT Server

At the SQL> prompt, you can execute SQL commands.

3.10.4 Creating RIM Object

We create a RIM object for the AIX RIM host that specifies the Oracle for NT server. We used the `wcrtrim` command to create the RIM object called `nt` when the vendor is Oracle, the `host_name` is `dublin`, the database is the database instance name, `ORCL`. The user is the user name that will connect to database, `system`, the `rdbms_home` is the directory where Oracle is installed, `/oracle/home`, and the `server_id` is the alias that is used to connect to the server, `nt.world`.

```
root@dublin[/#]# wcrtrim -v oracle -h dublin -d ORCL -u system -H
/oracle/home -s nt.world nt

RDBMS password:
```

Figure 114. Create Oracle RIM Object for AIX RIM Host

The default password for the Oracle user system is manager.

After we created your RIM object we can look at the values that we entered by using the `wgetrim` command.

```
root@dublin[/#]# wgetrim nt

RIM Host:      dublin
RDBMS User:    system
RDBMS Vendor:  Oracle
Database ID:   ORCL
Database Home: /oracle/home
Server ID:     nt.world
Instance Home:

root@dublin[/#]
```

Figure 115. Display RIM Object for AIX RIM Host

The `Instance Home:` should be blank.

3.10.5 Testing RIM Object

We verify that we can connect to the Oracle for NT server using the RIM object for the AIX client.

```
root@dublin[/#]# wrimtest -l nt

Resource Type : RIM
Resource Label : nt
Host Name      : dublin
User Name      : system
Vendor         : Oracle
Database       : ORCL
Database Home  : /oracle/home
Server ID      : nt.world
Opening Regular Session ..Opened
RIM : Enter Option > x
Releasing session

root@dublin[/#]
```

Figure 116. RIM Test Connect From AIX RIM Host To NT Server

The alias for the Server ID must be exactly the stanza name in the tnsnames.ora file on the AIX RIM host. In our setup, the Server ID alias defined in the tnsnames.ora file is nt.world. See Figure 111 on page 119.

Chapter 4. Troubleshooting Hints and Tips

Sometimes the specifics of what to do when something goes wrong is difficult to describe, but this section describes two Tivoli utilities that are useful when you are resolving problems and some errors that we observed during this project. The `wrimtest` utility is used as a quick and easy way to verify that a connection can be established between the RIM object and the database. The `wrimtrace` utility is used when more detail is required to resolve the problem.

4.1 Using RIM Test Utility to Resolve Problems

A common error is the "Could not connect to RDBMS server to access database" error, which you should resolve before you attempt to use a Tivoli application. The following is a list of errors that can occur while testing a RIM object with the `wrimtest` command and possible solutions.

4.1.1 ORA-12154 TNS:Could Not Resolve Service Name

We found two conditions that will cause this error while attempting to connect an Oracle client to an Oracle server.

1. Since `tnsnames.ora` file is owned by oracle user and dba group, you must add read permission for other users on the `tnsnames.ora` file, or the `wrimtest` command will not work, even if though this command is executed by root.

```
root@dublin[/#] ls -pla $ORACLE_HOME/network/admin/tnsnames.ora
-rw-r----- 1 oracle  dba          620 Apr 27 11:06 /oracle/home/network/admin
/tnsnames.ora
root@dublin[/#]
```

Figure 117. TNSNAMES.ORA Default File Permission Setting

Figure 118 on page 124 shows what the output will look like with no read permissions set for other users.

```

# wrimtest -l testeraix

Resource Type : RIM
Resource Label : testeraix
Host Name      : dublin
User Name      : sys
Vendor         : Oracle
Database       : test
Database Home  : /oracle/home
Server ID      : red.world
Opening Regular Session ..Could not connect to RDBMS server to access database t
est

The RDEMS server call's return code was: 12154

The RDEMS server call's error was: ORA-12154: TNS:could not resolve service name
#

```

Figure 118. ORA-12154 TNS: Could Not Resolve Service Name

You can change the permissions of the tnsnames.ora file with the `chmod` command:

```
# chmod o+r /oracle/home/network/admin/tnsnames.ora
```

2. If you have the Server ID set to the alias only without the domain as an extension, the ORA-12154 error message will appear as above.

```

# wgetrim testeraix

RIM Host:      dublin
RDEMS User:    sys
RDEMS Vendor:  Oracle
Database ID:   test
Database Home: /oracle/home
Server ID:     red
Instance Home:

#

```

Figure 119. Server ID Without Domain

You must use the fully qualified alias name (red.world).

4.1.2 ORA-03121 Error While Trying to Retrieve Text for Error

We found that you must add `@alias` of the Oracle database instance to the User Name, if the Oracle client is on NT. You see the error message in Figure 120 on page 125 when the user name is not set correctly.


```

C:\>wrimtest -l testernt

Resource Type : RIM
Resource Label : testernt
Host Name : sydney
User Name : system
Vendor : Oracle
Database : ORCL
Database Home : c:\orant
Server ID : green
Instance Home :

Opening Regular Session..Could not connect to RDEMS server to access database ORCL
The RDEMS server call's return code was: 3121
The RDEMS server call's error was:Error while trying to retrieve text for error
ORA-03121

C:\>

```

Figure 120. ORA-03121 Error while Trying to Retrieve Text for Error

As you can see, the error message is not very descriptive.

4.1.3 SQL1032N No Start Database Manager Issued

This error can occur if the database instance is not started.

```

C:\>wrimtest -l sample
Resource Type : RIM
Resource Label : sample
Host Name : toronto
User Name : db2
Vendor : DB2
Database : sample
Database Home : d:/sqllib
Server ID : tcpip
Instance Home : d:/sqllib/db2
Opening Regular Session...The RDEMS server call has failed.
The operation was: DB2 Error Code: -1032 SQLState:08001[IBM][CLI Driver] SQL10
32N No start database manager command was issued. SQLSTATE=57019
The RDEMS server call's return code was: -1
C:\>

```

Figure 121. SQL1032N No Start Database Manager Command Was Issued

Check if the instance is running on the database server. Log on to the database server with the instance owner's user ID and issue the `db2start` command in the DB2 command window or the command line on AIX.

4.1.4 SQL1402N Unable to Authenticate User

This error can occur if the DB2 Security Server is not running on NT.

```
C:\>wrintest -l sample
Resource Type : RIM
Resource Label : sample
Host Name : toronto
User Name : db2
Vendor : DB2
Database : sample
Database Home : d:/sqllib
Server ID : tcpip
Instance Home : d:/sqllib/db2
Opening Regular Session...The RDBMS server call has failed.
The operation was: DB2 Error Code: -1402 SQLState:S1000[IBM][CLI Driver] SQL14
02N Unable to authenticate user due to unexpected system error.
The RDBMS server call's return code was: -1
C:\>
```

Figure 122. SQL1402N Unable to Authenticate User

To check if the DB2 Security Server is running, double click on the Services icon in the Control Panel window. Look for the DB Security Server in the list provided. If it is not started, start it by clicking on the Start button in the Services panel.

4.1.5 Authorization Error - Insufficient Authorization

This error can occur if the user ID logged on to the managed node does not have the required Tivoli privileges.

```
C:\>wrintest -l sample
Resource Type : RIM
Resource Label : sample
Host Name : toronto
User Name : db2
Vendor : DB2
Database : sample
Database Home : d:/sqllib
Server ID : tcpip
Instance Home : d:/sqllib/db2
Opening Regular Session...An authorization error of type "insufficient authorization" occurred
Summary of possible error conditions:
"insufficient authorization" means that you have insufficient TME authorization to perform the operation
"permission denied" means that you have insufficient operating system privileges to perform the operations.
"authorization information expired" means that a task's authorization to run has expired
"Kerberos ticket expired" means that your Kerberos ticket expired and you must run kinit to get a new ticket.
"Kerberos unauthorized request" means that Kerberos has rejected the operation.
"Delegation Credential failure" means that invalid security credentials were used in the operation.
C:\>
```

Figure 123. Authorization Error - Insufficient Authorization

Refer to “Adding Appropriate Roles to Tivoli Administrator” on page 11 for instruction on how to correct this problem.

4.2 Using RIM Test Utility to Execute SQL Commands

The RIM test utility can be used from the command line interface to execute various SQL and database commands. In this section, we review the different wrintest options available to a RIM administrator.

4.2.1 List Help - Option ?

Use the ? option to see all the available options of the wrintest command.

```

C:\>wrintest -l sample
Resource Type : RIM
Resource Label : sample
Host Name : toronto
User Name : db2
Vendor : DB2
Database : sample
Database Home : d:/sqllib
Server ID : tcpip
Instance Home : d:/sqllib/db2
Opening Regular Session...Session Opened
RIM : Enter Option >?
Options -
  c : RIM_commit
  d : RIM_delete
  e : RIM_execute_sql
  g : RIM_retrieve_rows
  i : RIM_insert_rows
  r : RIM_rollback
  u : RIM_update_rows
  ? : Help List
  x : Exit
RIM : Enter Option >

```

Figure 124. RIM Test Options

We describe each of these options in the following sections.

4.2.2 RIM Retrieve Rows - Option g

The `g` option is used to generate a select statement that is executed through the RIM interface. When entering the `g` option, the user is prompted to provide a table name. This can be a fully qualified table name, such as `sysibm systables`, or just the name of the table. If just the table name is supplied, the user ID of the instance owner is used as the schema. For example, if the string `EMPLOYEE` is supplied and the instance owner user ID is `db2`, the table `db2.EMPLOYEE` is queried.

4.2.2.1 Enter Simple Select Statement

In the example 1 below, we executed the following SQL statement

```
select DEPTNAME from DEPARTMENT
```

```

RIM : Enter Option >g
Table Name > department
Enter <Field Name> [/n] [/s /l /f /d [<Value>]
Editor? [y/n] [Default n] >
1 > deptname /s
2 >
Where Clause >
Retrieve) Num of Rows [0] >
Row 0
    deptname : (0) SPIFFY COMPUTER SERVICE DIV.
Row 1
    deptname : (0) PLANNING
Row 2
    deptname : (0) INFORMATION CENTER
Row 3
    deptname : (0) DEVELOPMENT CENTER
Row 4
    deptname : (0) MANUFACTURING SYSTEMS
Row 5
    deptname : (0) ADMINISTRATION SYSTEMS
Row 6
    deptname : (0) SUPPORT SERVICES
Row 7
    deptname : (0) OPERATIONS
Row 8
    deptname : (0) SOFTWARE SUPPORT
RIM : Enter Option >

```

Figure 125. Select DEPTNAME from DEPARTMENT

By leaving the `Editor?` prompt blank, the default option `N` was used.

4.2.2.2 Enter Select Statement - Do Not Use Editor

In example 2, we executed the following SQL statement

```

select EMPLOYEE.FIRSTNAME,DEPARTMENT.DEPTNAME from DEPARTMENT,EMPLOYEE where
DEPARTMENT.DEPTNO=EMPLOYEE.WORKDEPT and EMPLOYEE.SALARY > 40000

```

We did not use the editor to enter the SQL statement.

```

RIM : Enter Option >g
Table Name > EMPLOYEE,DEPARTMENT
Enter <Field Name> [/n] [/s /l /f /d [<Value>]
Editor? [y/n] [Default n] >
1 > EMPLOYEE.FIRSTNAME
2 > DEPARTMENT.DEPNAME
3 >
Where Clause > DEPARTMENT.DEPTNO=EMPLOYEE.WORKDEPT AND EMPLOYEE.SALARY > 40000
Retrieve) Num of Rows [0] >
Row 0
      EMPLOYEE.FIRSTNAME : (0) CHRISTINE
      DEPARTMENT.DEPNAME : (0) SPIFFY COMPUTER SERVICE DIV.
Row 1
      EMPLOYEE.FIRSTNAME : (0) VINCENZO
      DEPARTMENT.DEPNAME : (0) SPIFFY COMPUTER SERVICE DIV.
Row 2
      EMPLOYEE.FIRSTNAME : (0) MICHAEL
      DEPARTMENT.DEPNAME : (0) PLANNING
Row 3
      EMPLOYEE.FIRSTNAME : (0) JOHN
      DEPARTMENT.DEPNAME : (0) SUPPORT SERVICES
RIM : Enter Option >

```

Figure 126. Enter Select Statement - Do Not Use Editor

4.2.2.3 Enter Select Statement - Use Editor

In example 3, we illustrate how to use the editor to enter a select statement.

The select statement we executed is the same as the one used in example 2.

```

RIM : Enter Option >g
Table Name > employee,department
Enter <Field Name> [/n] [/s /l /f /d [<Value>]
Editor? [y/n] [Default n] > y
_____vi Editor_____
EMPLOYEE.FIRSTNAME
DEPARTMENT.DEPINAME
~
~
~
~
~
~
~
:wg
_____
Where Clause > DEPARTMENT.DEPTNO=EMPLOYEE.WORKDEPT and EMPLOYEE.SALARY>40000
Retrieve) Num of Rows [0] >
Row 0
      EMPLOYEE.FIRSTNAME : (0) CHRISTINE
      DEPARTMENT.DEPINAME : (0) SPIFFY COMPUTER SERVICE DIV.
Row 1
      EMPLOYEE.FIRSTNAME : (0) VINCENZO
      DEPARTMENT.DEPINAME : (0) SPIFFY COMPUTER SERVICE DIV.
Row 2
      EMPLOYEE.FIRSTNAME : (0) MICHAEL
      DEPARTMENT.DEPINAME : (0) PLANNING
Row 3
      EMPLOYEE.FIRSTNAME : (0) JOHN
      DEPARTMENT.DEPINAME : (0) SUPPORT SERVICES
RIM : Enter Option >

```

Figure 127. Enter Select Statement - Use Editor

Since we executed this example on AIX, the vi editor was used.

4.2.3 RIM Execute SQL Statements - Option e

The *e* option is used to execute any database or SQL statement. This option creates a temporary file and opens it with the vi editor, if UNIX.

Note
 At the time of writing this book, this option was not available yet.

4.2.4 RIM Insert Rows - Option i

The *i* option is used to insert row into an existing table. The wrimtest utility prompts for the following information:

Table Name > The name of the table into which the row or rows are inserted. You may specify a fully qualified name such as db2.employee.

Editor? [y/n] [Default n] > The editor can be used to fill in the column names and the values to insert into the table. To use the editor, type in *y* at the prompt. The default is *n*.

1> Enter the column name followed by the type of data and the value to be used for the column. If no more column values need to be specified, leave the field blank.

Where Clause > Ignore this prompt by leaving it blank.

Valid data types are:

```
/s    string
/l    long
/f    float
/d    date
```

After a row's data is entered, the wrimtest utility prompts for a new table name. If another row needs to be inserted, continue by entering the table name. If no more rows need to be inserted, leave the table name blank to return to the `wrimtest` command prompt.

4.2.4.1 Insert Single Row of Data Into Table - Do Not Use Editor

In example 1, we use the `i` option to insert a single row of data into a table.

```
RIM : Enter Option >i
Table Name > employee
Enter <Field Name> [/n] [/s /l /f /d [<Value>]
Editor? [y/n] [Default n] > n
1 > EMPNO /s 099999
2 > FIRSINME /s JAN
3 > MIDINIT /s K
4 > LASTNAME /s MARAIS
5 > EDLEVEL /l 6
6 >
Where Clause >
Table Name >
RIM : Enter Option >r
RIM : Enter Option >
```

Figure 128. Insert Single Row of Data Into Table - Do Not Use Editor

After inserting the data, we use the `r` option to roll back the `insert` command.

4.2.4.2 Insert Single Row of Data Into Table - Use Editor

In example 2, we insert a single row of data into a table using the `i` option and the editor.

```
RIM : Enter Option >i
Table Name > employee
Enter <Field Name> [/n] [/s /l /f /d [<Value>]
Editor? [y/n] [Default n] > y
_____vi editor_____
EMPNO /s 099999
FIRSTNAME /s JAN
MIDINIT /s K
LASTNAME /s MARAIS
EDLEVEL /l 6
~
~
~
~
~
~
~
~
:wq
_____
Where Clause >
Table Name >
RIM : Enter Option >c
RIM : Enter Option >
```

Figure 129. Insert Single Row of Data Into Table - Use Editor

After inserting the data, we use the `c` option to commit the inserted data.

4.2.5 RIM Delete Rows - Option d

The `d` option is used to delete rows from a table. The `wrimtest` utility prompts for the following information:

- | | |
|----------------|---|
| Table Name > | The name of the table into which the row or rows will be inserted. You may specify a fully qualified name such as <code>db2.employee</code> . |
| Where Clause > | Specify the rows to be deleted. |

In this example, we delete all the rows in the table named `employee` that have column value `MARAIS` for the `LASTNAME` column.

```
RIM : Enter Option >d
Delete) Enter Table Name > employee
Delete) Enter Where Clause > lastname='MARAIS'
RIM : Enter Option >c
RIM : Enter Option >
```

Figure 130. Delete Row

We then commit the changes with the `c` option.

4.2.6 RIM Update Rows - Option `u`

This option can be used to update values of an existing row or rows. When entering the `u` option on the `wrimtest` command line the user are prompted for the same things as with the `g` option.

4.2.6.1 Update Rows - Do Not Use Editor

In example 1, we update the education level value for Jennifer Lutz.

```
RIM : Enter Option >u
Table Name > employee
Enter <Field Name> [/n] [/s /l /f /d [<Value>]
Editor? [y/n] [Default n] >
1 > EDLEVEL /l 7
2 >
Where Clause > firstnme = 'JENNIFER' and lastname ='LUTZ'
RIM : Enter Option >c
RIM : Enter Option >
```

Figure 131. Update Rows - Do Not Use Editor

After the changes are made, we commit the changes by using the `c` option.

4.2.6.2 Update Rows - Use Editor

In example 2, we change the surname for Jennifer Lutz to Smith by using the editor.

```

RIM : Enter Option >u
Table Name > employee
Enter <Field Name> [/n] [/s /l /f /d [<Value>]
Editor? [y/n] [Default n] >y
-----vi Editor-----
LASTNAME /s SMITH
~
~
~
~
~
~
~
~
~
~
~
:wg
-----
Where Clause > firstnme = 'JENNIFER' and lastname ='LUTZ'
RIM : Enter Option >c
RIM : Enter Option >

```

Figure 132. Update Rows - Use Editor

After the changes are made, we commit the changes by using the `c` option.

4.2.7 RIM Commit - Option c

When ever a change of data in the database is made through the insert, update or delete options, and we want the changed data to be permanently written to the database, the `c` option is used. If the `wrimtest` utility is terminated before the `c` option is used to commit all changes, all the changes are rolled back (or discarded).

4.2.8 RIM Rollback - Option r

By using the rollback option, all the changes made after a previous rollback or commit are discarded.

4.3 Using RIM Trace to Resolve Problem

The RIM trace facility provides a background look into the workings of RIM. The trace facility can gather valuable information about what went wrong when an error is reported.

By executing the command `wrimtrace` with no parameters, the syntax is given.

```
C:\>wrimtrace

Usage: wrimtrace rim_object_name [new_trace_level]

C:\>
```

Figure 133. Syntax of wrimtrace Command

The valid trace levels are ERROR, INFORMATION, and TRACE_OFF. These trace levels are not case sensitive.

If you specify a RIM object name and no new trace level, you display the current status of the RIM trace facility.

```
root@dublin[/#]# wrimtrace red

RIM Tracing is turned off

root@dublin[/#]#
```

Figure 134. RIM Trace Status

Next is an example where wrimtrace was used to find the problem.

4.3.1 The RDBMS Server Call Has Failed

The wrimtest utility reported the following error.

```
C:\>wrimtest -l sample
Resource Type : RIM
Resource Label : sample
Host Name : toronto
User Name : db2
Vendor : DB2
Database : boo_boo
Database Home : d:/sqllib
Server ID : tcpip
Instance Home : d:/sqllib/db2
Opening Regular Session...The RDBMS server call has failed.
The operation was:
The RDBMS server call's return code was: -1

C:\>
```

Figure 135. The RDBMS Server Call Has Failed

The error message is not very informative. To find out what went wrong, we decide to run a trace. The first thing we do is find out what the trace level is and then set it to information and error.

```
C:\>wrimtrace sample
RIM Tracing is turned off

C:\>wrimtrace sample "INFORMATION|ERROR"

C:\>wrimtrace sample
RIM Trace Level:  ERROR|INFORMATION

C:\>
```

Figure 136. Change RIM Trace Level

To activate the changes in the trace level for a RIM object, the process associated with it needs to be stopped on the RIM host. To terminate a process on NT:

1. Right-click the mouse on an unoccupied surface of the task bar.
2. Select the **Task Manager** from the displayed list.
3. Click on the **Applications** tab.
4. Select the **RIM_DB2_prog.exe** task.
5. Click on the **End Task** button.

See Part 137, “Windows NT Task Manager” on page 138.

If the RIM host is on UNIX, kill the process RIM_DB2_prog by issuing the `kill` command. Multiple RIM_DB2_prog processes may exist. You can kill all such processes because the process is started again if needed.

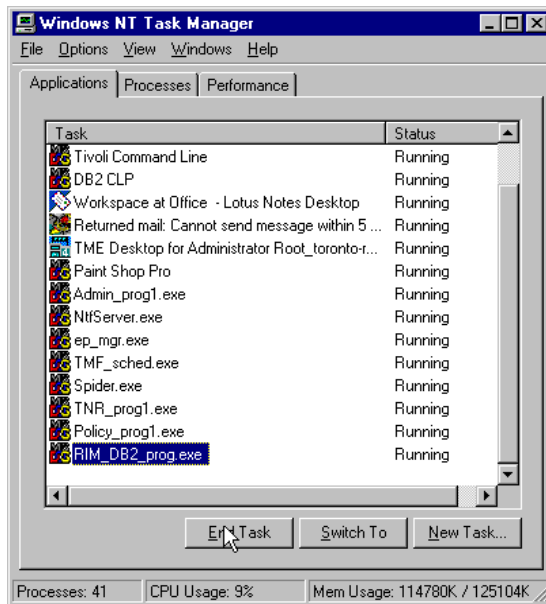


Figure 137. Windows NT Task Manager

Verify that the Tivoli environment is set up correctly on the RIM host. Using the `odadmin environ` command, verify that the RIM trace file is defined. If not, define one. The trace file is written on the RIM host.

```
C:\>odadmin environ get
NLSPATH=D:/Tivoli/msg_cat/%L/%N.cat

C:\>type c:\tmp\tivolienv.txt
NLSPATH=D:/Tivoli/msg_cat/%L/%N.cat
RIM_DB_LOG=C:\tmp\db2_rim_log

C:\>odadmin environ set < c:\tmp\tivolienv.txt

C:\>odadmin environ get
NLSPATH=D:/Tivoli/msg_cat/%L/%N.cat
RIM_DB_LOG=C:\tmp\db2_rim_log

C:\>
```

Figure 138. Define RIM Trace File

The first command displays the current settings for the Tivoli environment on the RIM host.

The second command displays the contents of a text file that contains the RIM trace file definition.

The third command updates the Tivoli environment with the RIM trace file definition.

The fourth command verifies that the Tivoli environment is correct.

Run the `wrimtest` command again.

```
C:\>wrimtest -l sample
Resource Type : RIM
Resource Label : sample
Host Name : toronto
User Name : db2
Vendor : DB2
Database : boo_boo
Database Home : d:/sqllib
Server ID : tcpip
Instance Home : d:/sqllib/db2
Opening Regular Session...The RDBMS server call has failed.
The operation was: DB2 Error Code: -1013 SQLState:08001[IBM][CLI Driver]
SQL1013N The database alias name or database name "BOO_BOO "
could not be found. SQLSTATE=42705
The RDBMS server call's return code was: -1

C:\>
```

Figure 139. SQL1013N Database Alias or Name Could Not Be Found

Now, the SQL1013N error data appears in the trace file. The output of the trace file is as follows:

```
C:\>tail c:\tmp\db2_rim_log

00000277 [Wed Apr 15 19:25:21 1998] Connection ID: 0, Operation: val_connect:SQL
Connect, DB Call: val_connect:SQLConnect
      DB2 Error Code: -1013 SQLState:08001[IBM][CLI Driver] SQL1013N The dat
abase alias name or database name "BOO_BOO " could not be found. SQLSTATE=42705

C:\>
```

Figure 140. RIM Trace File Output

Be sure to turn the RIM trace off after you are finished.

4.4 Oracle Performance Tips

The performance of Tivoli application databases, such as TEC or Inventory, could be greatly improved by using cost-based optimization and running the Oracle `sqlplus ANALYZE` command to gather statistics for the optimizer on a regular basis, for example, after each inventory scan.

4.4.1 Oracle Optimizer

Optimization is the process of choosing the most efficient way to execute a SQL statement. The Oracle optimizer chooses the way that it believes to be the most efficient. The Oracle optimizer uses either a rule-based or a cost-based approach to choose an execution plan for an SQL statement. The rule-based approach chooses execution plans based on heuristically ranked operations. The cost-based approach estimated the cost of each execution plan based on the data distribution and storage characteristics statistics for the tables, clusters, and indexes in the data dictionary. These statistics are generated using the `ANALYZE` command.

Reviewing the statistics generated by the `ANALYZE` command could improve the performance of retrieving hardware and software inventory, running queries, and producing report.

For more information about this topic, see Chapter 13, "The Optimizer", in the *Oracle 7 Server Concepts*, and Chapter 9, Tuning I/O, in the *Oracle 7 Server Tuning Guide*.

4.4.2 Determining Oracle Database Free Space

You can determine how much free space exists in an Oracle database by using `sqlplus` as the system user

```
sqlplus system
```

and executing the following query:

```
select tablespace_name, sum(bytes), sum(blocks) from dba_free_space group  
by tablespace_name
```

Figure 141 on page 141 shows an example of the output.


```
TABLESPACE_NAME          SUM(BYTES) SUM(BLOCKS)
-----
RBS                      23773184   5804
SYSTEM                  27422720   6695
TEC_TEMP                10481664   2559
TEMP                    561152     137
TIVOLI_TEMP_TS          5238784   1279
TOOLS                   24203264   5909
USERS                   1044480    255

7 rows selected.

SQL>
```

Figure 141. Oracle Free Space Query Output

The `sqlplus` command is available for UNIX systems

4.4.3 Sizing Database Objects

Performance could be improved by correctly sizing the objects in the database so that every table and index is contained in a single continuous extent in the database. While this is a precise task that can generally only be accomplished by an experienced database administrator, it can greatly improve the performance of both the retrievals and updates to the database.

An easier way to achieve contiguous storage is, after a full inventory scan of all subscribers, to export all the data using the Oracle `sqlplus exp` command, clean the database by deleting all tables and views, and then importing the database using the Oracle `sqlplus imp` command.

4.4.4 Database Backup

You should ensure that you have good backups and that this procedure is done by an experienced DBA or Systems Administrator who knows exactly what they are doing.

Chapter 5. How to Use the Tivoli Framework Query Facility

The Tivoli Framework query facility GUI tool is used to write and run queries and reduces the need for an end user to understand the syntax of SQL. The *Tivoli Inventory User's Guide* describes how to use this query facility to access information from a specific Tivoli application, Tivoli Inventory. In this section, we create and run queries for non-Tivoli application databases.

5.1 Tivoli Framework Query Facility Overview

The Tivoli Framework query facility enables you to use SQL functions to access information in a RIM repository. A RIM repository is a RDBMS database that a Tivoli application accesses through RIM.

The query facility consists of query libraries and queries. Query libraries reside in policy regions and are created to contain queries. Queries specify which RIM repository to search, which view or table within the repository to query, and what information to retrieve.

Views are created so a group of information can be accessed easily by a query. A view can be described as a custom table, or a way to group information from related tables.

5.2 Creating a Query Administrator

To be able to create a query, a Tivoli administrator needs the required Tivoli authorization. Normally the `query_view`, `query_execute`, `query_edit` would be granted to the person that creates and maintains a particular query.

The query facility can only be used to create a query if the schema or owner of the table or tables referenced in the query is the same as the user ID used by the RIM object to connect and execute SQL statements.

Note

If Oracle is used, you must specify the user ID of the owner of the tables you want to query in the `wcstrim` command.

If DB2 is used, the query facility GUI cannot be used to create a query on a table or tables with schema other than the instance owner's user ID. If you want to create such a query, use the command line to create the query and fully qualify the table.

Since a query can only exist in a query library, before we can create the query, we must create (or select) a query library.

We used the following steps to create a query library:

1. Log on to the Tivoli Desktop.
2. Select a policy region in which to create the query library by double-clicking on a policy region icon.
3. Select **Create => QueryLibrary** from the menu bar in the Policy Region window as shown in Figure 142 on page 144.

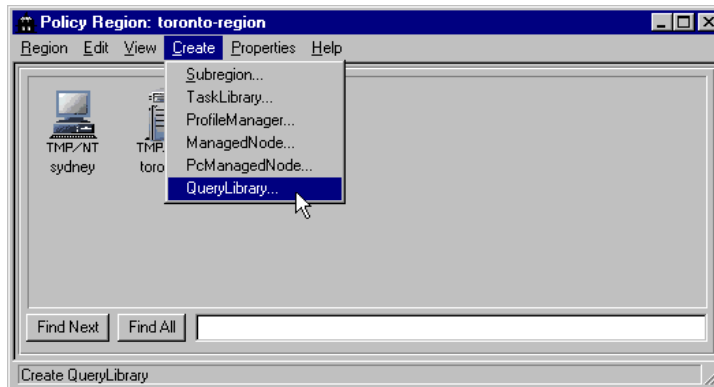


Figure 142. Policy Region: Create Query Library

4. Name the new query library as shown in Figure 143 on page 144. This name is displayed under the icon for the query library.



Figure 143. Create Query Library: Name

5. Click on the **Create & Close** button to create the new query library. Figure 145 on page 145 shows the newly created icon in the Policy Region window.

Now that a query library is created, we create a new query in that query library.

5.3 Creating a Query Using GUI

We created a query named Sample1 in a query library named Samples. The SQL statement we want to execute in the query is shown in Figure 144 on page 145.

```
select DEPARTMENT.DEPTNAME,  
       EMPLOYEE.FIRSTNAME,  
       VARCHAR(EMPLOYEE.HIREDATE),  
       YEAR(CURRENT DATE - EMPLOYEE.HIREDATE)  
from DEPARTMENT, EMPLOYEE  
where DEPARTMENT.DEPTNAME in ('MANUFACTURING SYSTEMS', 'SUPPORT SERVICES')  
and EMPLOYEE.WORKDEPT=DEPARTMENT.DEPTNO  
order by DEPARTMENT.DEPTNAME
```

Figure 144. SQL Statement for Query 1

We used the following steps to create a query:

1. Open a query library in which to create the new query by double clicking on its icon in the Policy Region window as shown in Figure 145 on page 145.

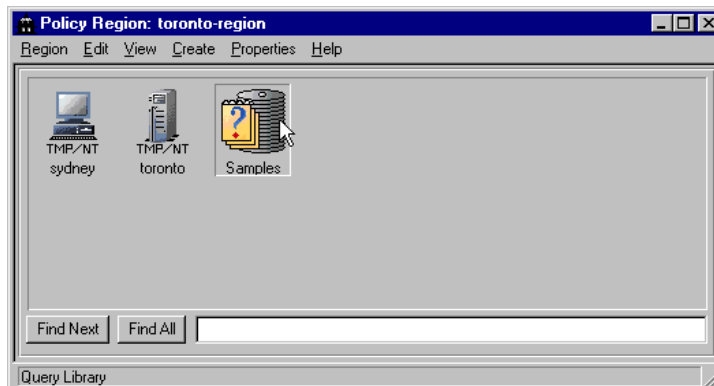


Figure 145. Policy Region: Query Library

2. Select **Create => Query** from the menu bar in the Query Library window as shown in Figure 146 on page 146.

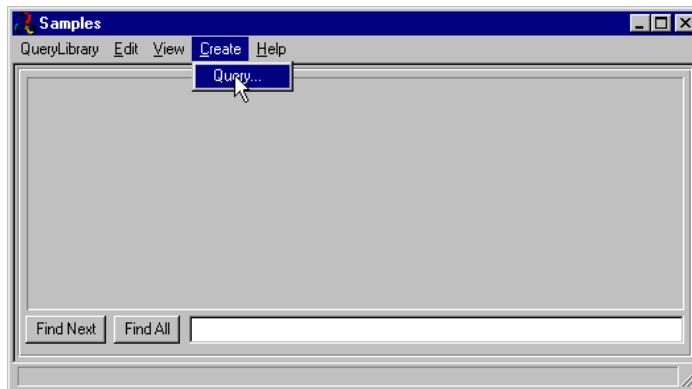


Figure 146. Query Library: Create a Query

3. Fill in all the required information in the Create Query window. See Figure 147 on page 147. We discuss the fields in the order one would normally fill in the fields.

Note

Some of the functions described in this section do not work for a non-Tivoli application created database.

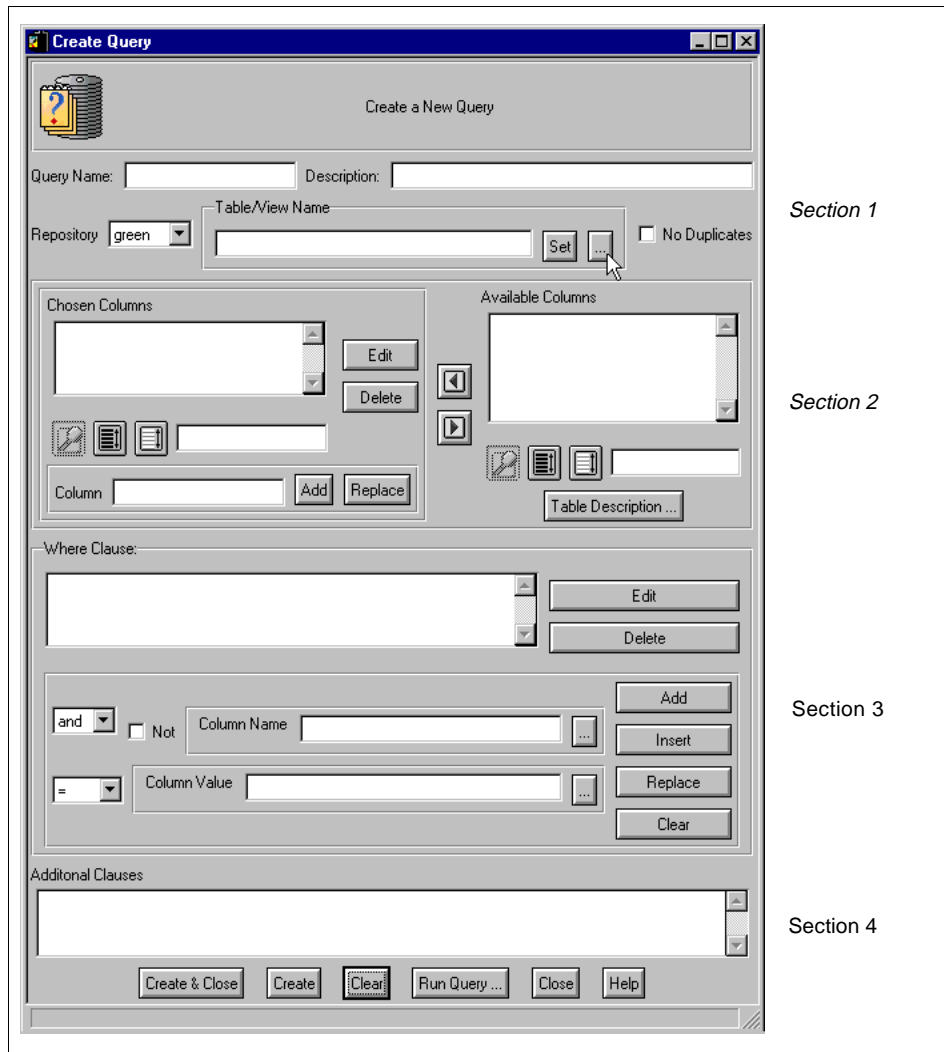


Figure 147. Create Query - All Sections

The Create Query panel is comprised of four sections, some of which are required and others that are optional. We discuss each of the sections separately.

5.3.1 Section 1 - Query Name, Repository, View/Table Name

Section 1 is shown in Figure 148 on page 148.

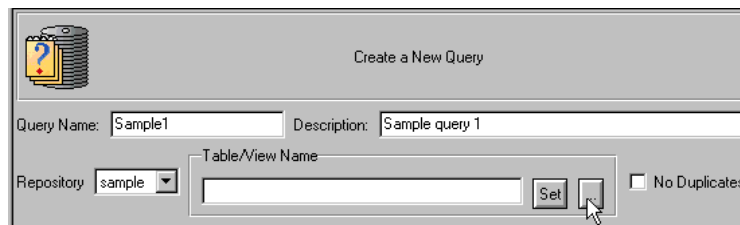


Figure 148. Create Query: Section 1

Query Name is used to refer to this query and as the icon name that is created in the query library window.

Description is an optional field that holds a description of the query.

Repository is the name of the RIM object that is used. The repository can be selected from the drop down list.

Table/View Name is the name of the table or the view that is queried. This field must be filled in. There are two ways to enter the table or view names, manually by typing in the names or by selecting the names from the list. If you want to manually type in multiple table or view names, separate them with a comma. To select multiple tables, press and hold the CTRL key and right-click the mouse on the tables you want to select.

A list of available tables or views is displayed when the '...' button next to the Set button is clicked. See Figure 149 on page 149 for the current list of available tables or views. If an error occurs and the database was created by a non-Tivoli application, refer to "Creating Views for Non-Tivoli Application" on page 156.

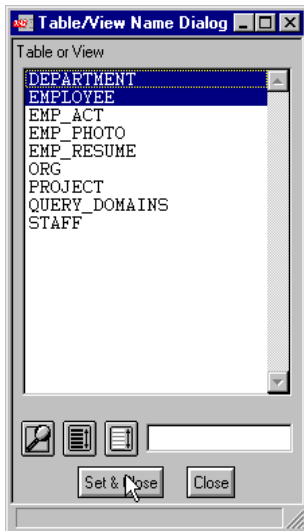


Figure 149. Create Query: Table/View Name Dialog

If the table has a schema or owner other than the user ID used in the `wcrtim` command, you must fully qualify the table name. For example, if a table named `employee` was created with the user ID `johns` and the RIM object uses the user ID `dbadmin` to connect to the database, the table must be qualified as `johns.employee`.

Click on the **Set** button in Section 1 to select the tables and populate the Available Columns in Section 2.

5.3.2 Section 2 - Columns

In Section 2, we specify which columns and or values should be returned.



Figure 150. Create Query: Section 2

Available Columns lists of all the columns of all the tables we specified in the Table/View Name field in Section 1.

Chosen Columns lists of all the columns and/or values we have selected.

Column field is used to edit a selected column from the Chosen Columns list or to add a value or column to the list without selecting it from the Available Columns list.

By clicking on the Table Description button, the Table Description window (see Figure 151 on page 150) is displayed. The Table Description window describes all the available columns in the tables selected.

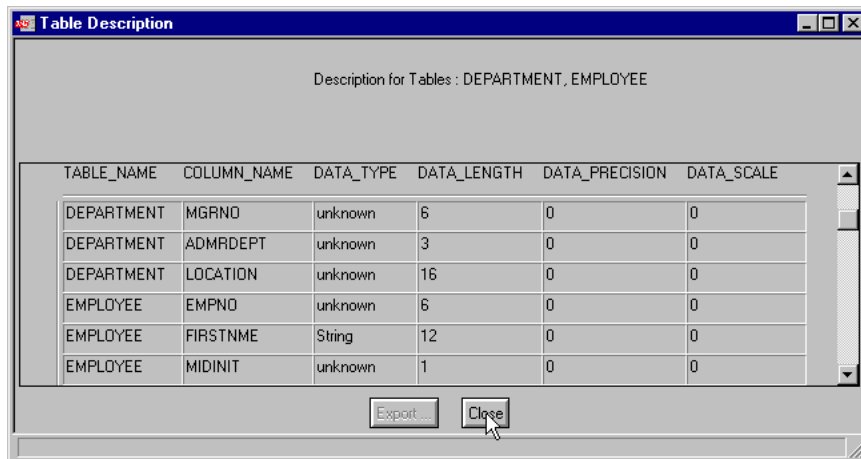


Table Description window showing the description for tables: DEPARTMENT, EMPLOYEE.

TABLE_NAME	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	DATA_PRECISION	DATA_SCALE
DEPARTMENT	MGRNO	unknown	6	0	0
DEPARTMENT	ADMRDEPT	unknown	3	0	0
DEPARTMENT	LOCATION	unknown	16	0	0
EMPLOYEE	EMPNO	unknown	6	0	0
EMPLOYEE	FIRSTNAME	String	12	0	0
EMPLOYEE	MIDINIT	unknown	1	0	0

Buttons: Export..., Close

Figure 151. Create Query: Table Description

To add a SQL function to a column name, do the following:

1. Select the columns that should be returned when the query is run from the Available Columns list.
2. Click on the arrow button pointing left.
3. Highlight the column on which the column function should work.
4. Click on the **Edit** button.
5. Edit the **Column** name, adding the column function.
6. Click on the **Replace** button.

Here is a list of some column functions:

`max(Column_name)`

Returns the highest value in the table for Column_name.

avg(Column_name)	Returns the average value for the Column_name column.
min(Column_name)	Returns the minimum value in the Column_name column.
Column_name/avg(Column_name)	Calculates the Column_name divided by the average value of the Column_name column for each row.
VARCHAR(Column_name)	Changes the column data type to VARCHAR.

Valid values may be:

- CURRENT DATE
- CURRENT TIME
- CURRENT TIMESTAMP

Note

The availability of these column functions and values are dependent on the RDBMS used. Refer to the SQL reference manual supplied with the RDBMS for the valid column functions and values.

5.3.3 Section 3 - Where Clause

Section 3 is used to build the *where clause* of the SQL statement that is used to limit the amount of rows that the RDBMS will return. The where clause is not required. A where clause comprises of one or more predicates. A predicate can be seen as a single condition. Predicates are 'joined' by adding an 'and' or an 'or' between them.

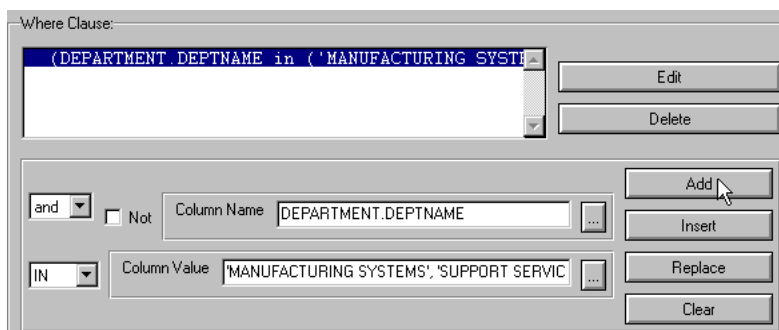


Figure 152. Create Query: Section 3

Section 3 is subdivided into two parts. The top part displays the current or active where clause and the bottom part is used to edit an existing one or build a new predicate in the where clause.

The bottom part consists of the following:

The top left drop down list is used to separate the predicates. Valid values are "and" or "or".

The Not check box is used to negate the compliance to the predicate. For example, if row1 and row5 comply with a predicate and the Not box is then checked, all the rows except row1 and row5 will then comply with the predicate.

The Column Name field is the name of the column that is used in the predicate. Multiple columns can not be filled in. There are two ways to enter the column name. The name can be typed in manually or name can be selected from a list. By clicking on the '...' button next to the Column Name field, a list of all the columns names in all the tables selected in Section 1 is returned. See Figure 153 on page 152. Click on the **Close** button when column is selected. The selected column name is now filled into the Column Name text box.

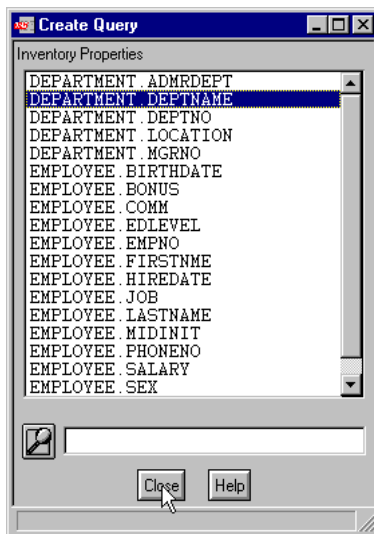


Figure 153. Create Query: Available Columns

The drop-down list in the bottom left corner provides an easy way to select the condition that will be used in the predicate. The LIKE conditional operator is used to compare column values to a list of values separated by commas. The IN condition is normally used to do a wildcard search.

NOTE

The wildcard character in SQL is the "%" character.

Numeric values are always specified without any quotes.

Character string values are always specified within single quotes.

The Column Value field holds the value or values that the column selected in the Column Name field will be compared with. This value can be entered from the keyboard or selected from a list.

By clicking on the '...' button next to the Column Value field, a list will be displayed from which a selection can then be made. See Figure 154 on page 153.

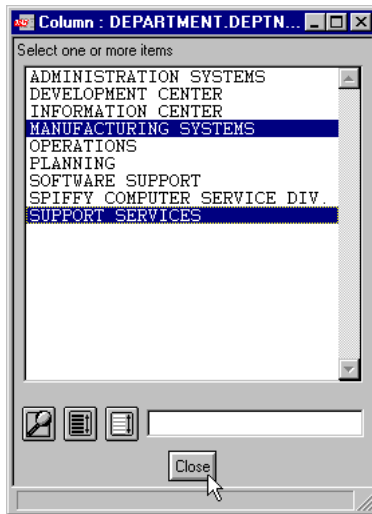


Figure 154. Create Query: Column Values List

To select more than one value, press and hold the **Ctrl** key and click on the values to select. You are only allowed to select multiple values if LIKE or IN is used as a conditional operators.

5.3.4 Section 4 - Additional Clauses

Section 4 is used to add any other conditions that could not be entered through Section 3.

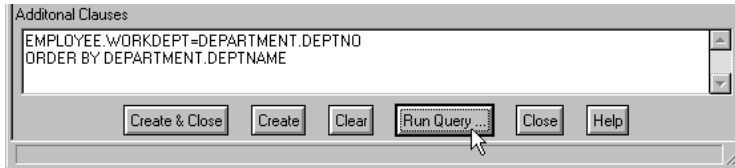


Figure 155. Create Query: Section 4

After you complete the Create Query, click on the **Run Query** button. The result of the query is displayed in a window (see Figure 156 on page 154).

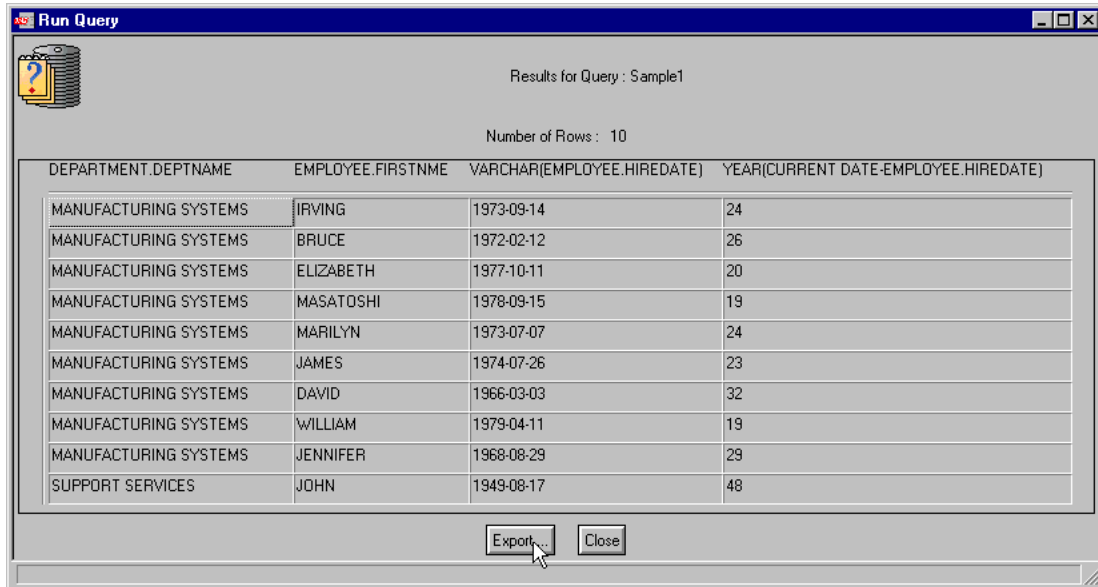


Figure 156. Run Query - Results

When you are satisfied with the query results, click on the **Close** button and then the **Create & Close** button in the Create Query panel as shown in Figure 155 on page 154.

The query is now created and an icon appears in the Query Library window in which it was created.

5.4 Creating a Query Using the Command Line

The `wrcquery` command is used to create a query from the command line. Its syntax is as follows:

Usage:

```
wcrtquery [-d desc] [-r repository] [-v view_name] [-c column]... [-i | -w  
where_clause | -s] query_lib query_name
```

The parameters are:

-d	Specifies the string that is used as the description for the query.
-r	Specifies the name of the RIM object against which the query must run.
-v	Specifies the name of the table(s) and or view(s) that the query is run against. If multiple tables or views are specified, they are separated with a comma and the entire string is within quotes.
-c	Specifies a column name in quotes. Multiple -c parameters are allowed, one per column that is selected. Column functions are also allowed. If more than one table or view is selected from the column names, the column name must be fully qualified. The format is TABLE_NAME.COLUMN_NAME
-i	Specifies that the non-structured where clause is read from standard input.
-w	Specifies a non-structured where clause on the command line. Do not include the keyword <code>where</code> before any of the predicates and put the where clause within quotes.
-s	Specifies that a structured where clause is read from standard input.
query_lib	Specifies the name of the query library in which the query belongs.
query_name	Specifies the name of the query. This name is used below the icon of the query in the query library window.

We want create a query named Sample2 in the query library Sample that already exists. The SQL statement we want to execute is shown in Figure 157 on page 156.

```

select DEPARTMENT.DEPTNAME,
       EMPLOYEE.FIRSTNAME,
       VARCHAR(EMPLOYEE.PHONENO)
from EMPLOYEE,DEPARTMENT
where DEPARTMENT.DEPTNO=EMPLOYEE.WORKDEPT
and EMPLOYEE.SALARY > (select AVG(SALARY) from EMPLOYEE)

```

Figure 157. SQL Statement for Query 2

The `wrtquery` command we used to create Sample2 query is shown in Figure 158 on page 156.

```

C:\> wrtquery -d "Sample query 2" -r sample -v "EMPLOYEE,DEPARTMENT"
-c "DEPARTMENT.DEPTNAME" -c "EMPLOYEE.FIRSTNAME"
-c "VARCHAR(EMPLOYEE.PHONENO)"
-w "DEPARTMENT.DEPTNO=EMPLOYEE.WORKDEPT and EMPLOYEE.SALARY >
(select AVG(SALARY) from EMPLOYEE)" Samples Sample2

C:\>

```

Figure 158. Create Query Command for Query 2

The `wrtquery` command and its parameters are entered on the command line.

5.5 Creating Views for Non-Tivoli Application

The Tivoli query facility can query data stored in a database created by a non-Tivoli application. In this section, we discuss considerations before accessing such databases.

5.5.1 Creating Catalog Views for Non-Tivoli Application for DB2

If the Tivoli Framework query facility is used to access data stored in a database created by a non-Tivoli application, additional catalog views need to be created. These views are used by the query facility to determine which tables are defined and what columns they are comprised of. The SQL script we used to create these required views for the DB2 sample database is shown in Figure 159 on page 157.


```

drop view QUERY_VIEWS;

drop view USER_TAB_COLUMNS;

drop view QUERY_DOMAINS;

create view QUERY_VIEWS (VIEW_NAME)
  as select varchar(tabname)
     from syscat.tables
     where TABSCHEMA='### Instance owner 's user ID ###'
        and TABNAME not in ('QUERY_VIEWS','USER_TAB_COLUMNS');

create view USER_TAB_COLUMNS
  (TABLE_NAME,COLUMN_NAME,DATA_TYPE,
   DATA_LENGTH,DATA_PRECISION,DATA_SCALE,
   COLUMN_ID)
  as
  select TABNAME,COLNAME,TYPENAME,LENGTH,
         SCALE,SCALE,COLNO
     from SYSCAT.COLUMNS;

create view QUERY_DOMAINS
  (TABLE_NAME,COLUMN_NAME,DOMAIN_VALUE)
  as
  select TABNAME,COLNAME,COLVALUE
     from syscat.coldist;

```

Figure 159. Create Views for DB2 Database

The first three statements delete previously defined QUERY_VIEWS, USER_TAB_COLUMNS and QUERY_DOMAINS views.

The fourth statement creates the QUERY_VIEWS view that is used by the query facility to determine which tables are accessible in the database. The script uses the DB2 catalog view SYSCAT.TABLES as the base to build the new view. In the *where* clause, we limit the tables visible to the end user through the query facility to all the tables created by our instance owner db2, excluding the QUERY_VIEWS, USER_TAB_COLUMNS and QUERY_DOMAINS views.

The fifth statement creates the USER_TAB_COLUMNS view. The query facility uses this view to gather information about columns defined to a table. The script use the DB2 catalog view SYSCAT.COLUMNS as the base to build the new view.

The sixth statement creates the QUERY_DOMAINS view that is used by the query facility to find all the valid data entries in a column. The script uses the DB2 catalog view SYSCAT.COLDIST as the base to build the new view.

This script should be executed under the instance owners user ID.

5.5.2 How to Execute the SQL Script for DB2

On the DB2 database server, connect to the database referenced by the RIM object with the user ID of the instance owner, create a file containing the script in Figure 159 on page 157 and substitute your instance owner's user ID in the line:

```
where TABSCHEMA='### Instance owner 's user ID ###'.
```

Execute the script using the command:

```
db2 -tvf script_file_name | more
```

This command and the SQL script is the same for both DB2 for NT and DB2 for AIX.

5.5.3 RIM Supported DB2 Data Types

Unfortunately not all the data types used by DB2 are currently supported by RIM. The supported data types are:

Table 7. DB2 Data Types Supported by RIM

DB2 data type	RIM data type
TIMESTAMP	DATE_TIME (/d)
VARCHAR	STRING (/s)
INTEGER	LONG (/l)
FLOAT	FLOAT(/f)

This leaves all the other DB2 data types unsupported. However, a query can be structured in such a way that an unsupported data type can be converted to a supported data type before the answer set is passed to RIM. By using this technique, we can "trick" RIM to read an unsupported data type as if the data type was supported. This can be best illustrated by an example.

A table with the name TAB1 is created. It has three columns named COL1, COL2 and COL3. COL1 is type VARCHAR. COL2 is type CHARACTER. COL3 is type DATE. Of the three columns, only COL1 is a supported type and can be used in the Tivoli query facility. Values from COL2 and COL3 can not be returned directly as data types because CHARACTER and DATE are not data types supported by RIM. If we attempt to execute the following SQL statement from within the Tivoli query facility, an error will occur.

```
select COL1,COL2,COL3 from TAB1
```

If, however, we convert the data types of the columns we select to a supported data type, the query will execute successfully. This can be done as follow:

```
select COL1,varchar(COL2),varchar(COL3) from TAB1
```

There are data types that can not be converted to any of the supported data types. These data types, however, are not often used in management applications and are of lesser concern. Typical conversions of unsupported data types are:

CHARACTER => VARCHAR

SMALLINT => INTEGER

DECIMAL => FLOAT

DATE => VARCHAR

5.5.4 Creating Catalog Views for Non-Tivoli Applications for Oracle

If the Tivoli query facility is used to access data stored on the Oracle databases created by non-Tivoli applications, you need to create Oracle views that enable the Tivoli query facility to access the Oracle databases and the defined tables.

You can create these views using SQL*Plus.

The first created view is QUERY_VIEWS, which is used by the query facility to display the Owner and Table name of the tables that do not belong to users SYS, SYSTEM, SYSMAN and DBSNMP.

```
create view QUERY_VIEWS (VIEW_NAMES)
as
    select OWNER||'.'||TABLE_NAME as VIEW_NAMES
    from ALL_TABLES
    where OWNER not in ('SYS','SYSTEM','SYSMAN','DBSNMP');
```

Figure 160. Create QUERY_VIEWS for Oracle

The second created view is USER_TAB_COLUMNS, which is used by the query facility to display information about the columns defined in the tables.

```

create view USER_TAB_COLUMNS
      (TABLE_NAME,COLUMN_NAME,DATA_TYPE,DATA_LENGTH,
       DATA_PRECISION,DATA_SCALE,COLUMN_ID)
  as
      select TABLE_NAME,
             COLUMN_NAME,
             DATA_TYPE,
             DATA_LENGTH,
             DATA_PRECISION,
             DATA_SCALE,
             COLUMN_ID
      from SYS.USER_TAB_COLUMNS;

```

Figure 161. Create USER_TAB_COLUMNS for Oracle

The third created view is QUERY_DOMAINS, which is used by the query facility to find all the valid data entries in a column.

```

create view QUERY_DOMAINS
      (TABLE_NAME,COLUMN_NAME,DOMAIN_VALUE)
  as
      select unique TABLE_NAME,COLUMN_NAME,'Data not available'
      from SYS.USER_TAB_COLUMNS;

```

Figure 162. Create QUERY_DOMAINS for Oracle

Since the Oracle starter database ORCL does not have a view similar to the view SYSCAT.COLDIST available with the sample DB2 database, we populated the QUERY_DOMAINS with the constant `Data not available`. A SQL statement could be written to find all the valid data entries in a column.

Appendix A. Other Databases Supported by RIM

This appendix describes the database products supported by Tivoli RIM.

A.1 Sybase Overview

Sybase is the leading competitor to Oracle in the RDBMS market. The Sybase database is optimized for OLTP-type applications and is generally the fastest RDBMS "out of the box"—that is, without the benefit of tuning. Sybase also tends to be easier to install and administer than Oracle, but offers fewer features and configuration options than does Oracle.

A.1.1 Sybase RIM Object

Table 8 on page 161 shows the Sybase specific values that go into the RIM object.

Table 8. Sybase RIM Attributes

Parameters	Value	Description
RDBMS Vendor	Sybase	RDBMS vendor name - must be Sybase
RIM Host		Managed node where the RIM server will run.
Database ID		Name of the Sybase database that the application will use. For Tivoli Inventory, the default is inventory.
Database Home	SYBASE	Directory where Sybase software is installed and which contains the Sybase interfaces file (UNIX).
Server ID	DSQUERY	Name of the Sybase server. On UNIX, the Sybase server name is defined in the Sybase interfaces file (\$SYSBASE/interfaces).
RDBMS User Name		RDBMS user that owns the application's data space. By default, this is tivoli for Tivoli Inventory and tec for Tivoli Enterprise Console.
RDBMS Password		Password for Sybase user defined as RDBMS User Name (see description above).

SYBASE and DSQUERY are environmental variables.

A.1.2 Sybase Versions Supported

RIM supports all versions of Sybase 10 and 11 on all Tivoli Tier 1 platforms. Tivoli Tier 1 platforms are Sun, Hewlett-Packard, IBM, Microsoft, and Novell. Tivoli recommends that customers running Sybase 10 upgrade to Sybase 11. Currently, Tivoli is testing on Sybase 11.0.3 upgrade.

A.1.3 Using ISQL

The SQL interpreter for Sybase is isql command. You must set the SYBASE and DSQUERY environmental variables to use isql.

```
isql -U <user_name> [-P password]
```

The SYBASE environmental variable is set to the path name of the directory where Sybase is installed. The DSQUERY environmental variable is set to a server name defined in the Sybase interfaces file. See “Sybase Interfaces File (UNIX)” on page 162. You should have the \$SYBASE/bin directory in your \$PATH.

When you successfully log in to isql, you should see the 1> prompt. Unlike Oracle SQL*Plus, there is no command termination character, like ; (semi-colon), that causes the SQL command to be executed. All SQL commands are executed by typing **go** at the isql prompt. You can enter multiple SQL commands at the isql prompt and then execute them as a batch by entering go.

A.1.4 Sybase Interfaces File (UNIX)

The Sybase interfaces file is all that is necessary to allow a client application to talk to a remote server. The Sybase interfaces file is located in the \$SYBASE directory.

You can not edit the Sybase interfaces file on a Solaris system. You must use the sybinit utility.

The Sybase interface files in Figure 163 on page 163 are from the *TME 10 Deployment Cookbook: Inventory and Company* redbook, SG24-2120.

```
## 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 163. Sybase Interfaces File (UNIX)

The system, rs600015, is the Sybase 11 database server on AIX.

Additional information about using Sybase for AIX is in Chapter 3 of the *TME 10 Deployment Cookbook: Inventory and Company* redbook, SG24-2120.

A.1.5 Sybase Interfaces File (Windows NT)

The interfaces file does not exist on Windows NT. Instead, a SQL.INI file is used.

```

;; 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 164. Sybase SQL.INI File (Windows NT)

The system, victor, is the Sybase 11 database server on Windows NT.

Additional information about using Sybase for Windows NT is in Chapter 2 of the *TME 10 Deployment Cookbook: Inventory and Company* redbook, SG24-2120.

A.1.6 Asynchronous I/O Configuration

You must have asynchronous I/O set on AIX machines that are Sybase servers.

Turn off asynch I/O on Windows NT systems due to problems.

A.1.7 Sybase Installation Tips

Sybase installation is discussed in Chapters 2 and 3 of the *TME 10 Deployment Cookbook: Inventory and Company* redbook, SG24-2120.

A.2 Microsoft SQL Server Overview

Microsoft SQL Server is an adaptation of Sybase Version 4.9 to the Windows NT operating system. Thus, Sybase and MS SQL Server are very similar and knowledge of one means that you know most of what you need to know about the other. The primary differences between the two RDBMS's from a RIM user's perspective are:

- There is no equivalent in SQL Server to the Sybase SQL.INI file. SQL Server uses named pipes as its default networking method. Named pipes require no user configuration. TCP/IP can be used instead of named pipes, but involves some configuration work on the part of the user.
- MS SQL Server provides a number of GUI tools not available in Sybase, most notably a GUI-enabled version of isql called isql_w.

A.2.1 MS SQL Server RIM Object

Table 9 shows the MS SQL Server specific values that go into the RIM object.

Table 9. MS SQL Server RIM Attributes

Parameters	Value	Description
RDBMS Vendor	MS_SQL	RDBMS vendor name - must be MS_SQL.
RIM Host		Managed node where the RIM server will run. The RIM host must be a managed node running Windows NT and must have MS SQL Server installed on it, even though the RIM host is not running MS SQL Server.
Database ID		Name of the MS SQL Server database that the application will use, or the alias for that database, if an alias exists..

Parameters	Value	Description
Database Home		Directory where MS SQL Server is installed.
Server ID		Name of the machine running MS SQL Server.
RDBMS User Name		MS SQL Server user that owns the application's data space. By default, this is tivoli for Tivoli Inventory and tec for Tivoli Enterprise Console.
RDBMS Password		Password for MS SQL Server user defined as RDBMS User Name (see description above).

MS_SQL is the designation for Microsoft SQL Server.

A.2.2 MS SQL Server Versions Supported

RIM supports MS SQL Server Version 6.5.

A.2.3 MS SQL Server Installation Tips

MS SQL Server provides a default installation with a set of pre-defined configuration options.

MS SQL Server installation is discussed in Chapter 6 of the *TME 10 Inventory 3.2: New Features and Database Support* redbook, SG24-2135.

Appendix B. Useful DB2 Commands

This chapter gives some useful command line commands to manipulate and check DB2 objects. These commands can be invoked from an AIX terminal after the environment has been setup for the user, or from a DB2 Command Window.

B.1 How to Find Help

Before looking for help, you should be aware of what kind of help it is you are looking for. Is it help on the syntax of a command, or a "How to..." kind of a problem?

B.1.1 Help on Syntax or Commands

The `db2 ?` on NT (or `db2 "?"` on AIX) command lists all the commands that you can use on the command line.

```

D:\SQLLIB\BIN> db2 ?
db2 [option ...] [db2-command | sql-statement |
    [? [phrase | message | sqlstate | class-code]]]
option: -a, -c, -e{c|s}, -finfile, -lhistfile, -o, -p, -rreport, -s, -t, -tdi,
        -v, -w, -zoutfile.
db2-command:
  ACTIVATE DATABASE      EXPORT                PRUNE HISTORY
  ATTACH                 FORCE APPLICATION     QUERY CLIENT
  ATTACH TO              GET AUTHORIZATIONS   QUIESCE TABLESPACES
  BACKUP DATABASE        GET CONNECTION STATE  QUIT
  BIND                   GET INSTANCE          REBIND
  CATALOG APPC NODE      GET MONITOR SWITCHES REGISTER
  CATALOG DATABASE       GET SNAPSHOT          RELEASE
  CATALOG DCS DATABASE   GET/RESET/UPDATE DB CFG REORG TABLE
  CATALOG GLOBAL DATABASE GET/RESET/UPDATE DEM CFG REORGCHK
  CATALOG IPXSPX NODE    HELP                  RESET MONITOR
  CATALOG LOCAL NODE     IMPORT                RESTART DATABASE
  CATALOG NETBIOS NODE   INVOKE                RESTORE DATABASE
  CATALOG TCPIP NODE     LIST APPLICATIONS     ROLLFORWARD DATABASE
  CHANGE DATABASE COMMENT LIST BACKUP/HISTORY   RUNSTATS
  CHANGE SQLISL          LIST COMMAND OPTIONS  SET CLIENT
  CONNECT                LIST DATABASE DIRECTORY SET CONNECTION
  CONNECT RESET          LIST DCS APPLICATIONS SET TABLESPACE CONTAINER
  CONNECT TO             LIST DCS DIRECTORY    TERMINATE
  CREATE DATABASE        LIST INDOUBT TRANSACTIONS UNCATALOG DATABASE
  DB2START/DB2STOP      LIST NODE DIRECTORY   UNCATALOG DCS DATABASE
  DEACTIVATE DATABASE    LIST PACKAGES/TABLES  UNCATALOG NODE
  DEREGISTER            LIST TABLESPACE CONTAINERS UPDATE COMMAND OPTIONS
  DETACH                 LIST TABLESPACES     UPDATE HISTORY
  DISCONNECT            LOAD                   UPDATE MONITOR SWITCHES
  DROP DATABASE          LOAD QUERY
  ECHO                   PREP/PRECOMPILE
Note: Some commands are operating system specific and may not be available.
For further help: ? db2-command - help for specified command
                  ? OPTIONS     - help for all command options
                  ? HELP       - help for reading help screens
                  db2help      - DB2 online manuals.
D:\SQLLIB\bin>

```

Figure 165. DB2 Command Help

The `db2 ? <db2_command>` on NT (or `db2 "? <db2_command>"` on AIX) command will give you a description of the syntax to use when executing the command. If you are unfamiliar with how to read these help screens, try issuing the `db2 ? help` command.

```

D:\SQLLIB\BIN> db2 ? list applications
LIST APPLICATIONS [FOR DATABASE database-alias] [SHOW DETAIL]
NOTE: From the operating system prompt, prefix commands with 'db2'.
      Special characters MAY require an escape sequence (\), for example:
      db2 \? change database
      db2 ? change database xxx comment with \"text\"
D:\SQLLIB\BIN>

```

Figure 166. DB2 List Applications Help Syntax

B.1.2 Help on SQL Statements

The `db2 help` command displays the online DB2 SQL Reference manual.

The `db2 help <sql_command>` command displays the online DB2 SQL Reference Manual and opens it to the section related to the `<sql_command>`.

B.1.3 Help on Error Codes

The `db2 ? <error_code>` displays the meaning of the `<error_code>`. The error code is in the format `SQLxxxxd` where `xxxx` is a four digit error code and `d` a one digit alpha character that can be `n` for a negative return code or `w` for a warning.

```

D:\SQLLIB\BIN>db2 ? sql0911n
SQL0911N The current transaction has been rolled back
         because of a deadlock or timeout. Reason code
         "<reason-code>".
Explanation: The current unit of work was involved in an
unresolved contention for use of an object and had to be
rolled back.
The reason codes are as follows:
2      transaction rolled back due to deadlock.
68     transaction rolled back due to lock timeout.
Note: The changes associated with the unit of work must be
      entered again.
The application is rolled back to the previous COMMIT.
User Response: To help avoid deadlock or lock timeout,
issue frequent COMMIT operations, if possible, for a
long-running application, or for an application likely to
encounter a deadlock.
sqlcode:  -911
sqlstate: 40001
D:\SQLLIB\BIN>

```

Figure 167. DB2 Error Code Help

When an error is return by DB2, make a note of the SQL code, the SQL state, and the reason code, if available. The combination of the three could help you to narrow down the reason for the error occurring.

B.1.4 On-line Manuals.

The DB2 manuals are available online after you install the product.

B.2 Finding the Database Names Known to DB2

Use the following command to find the names of the databases known to DB2:

```
db2 list database directory
```

This command returns a list of databases known to the machine and, in particular, the instance under which the command is executed.

```
D:\SQLLIB\BIN>db2 list database directory

System Database Director

Number of entries in the directory = 1

Database 1 entry:

Database alias           = SAMPLE
Database name           = SAMPLE
Local database directory = D:\DB2
Database release level  = 6.00
Comment                 =
Directory entry type    = Indirect

D:\SQLLIB\BIN>
```

Figure 168. DB2 List Database Directory on DB2 Server

The DB2 instance is SAMPLE.

```
db2admin@dublin[/db2]$ db2 list database directory

System Database Directory

Number of entries in the directory = 1

Database 1 entry:

Database alias           = SAMPLE
Database name           = SAMPLE
Node name                = DB2TIV
Database release level  = 6.00
Comment                  =
Directory entry type    = Remote

db2admin@dublin[/db2]$
```

Figure 169. DB2 List Database Directory on DB2 Client

The Node name DB2TIV is the DB2 service name defined in the `/etc/services` file on the client dublin.

B.3 Finding Other Servers or Instances Known to DB2

Use the following command to find other servers or instances known to DB2:

```
db2 list node directory
```

The output of this command provides information about remote instances or nodes defined to the instance. It gives an idea of what protocol is used and in the case of TCPIP, what the host name is and service port it is using. This is a helpful command when trying to configuring a client to connect to a server.

```

root@dublin[/#]# su - db2admin

db2admin@dublin[/db2]$ db2 list node directory

Node Directory

Number of entries in the directory = 1

Node 1 entry:

Node name           = DB2TIV
Comment             =
Protocol            = TCPIP
Hostname            = bombay
Service name        = db2tivoli

db2admin@dublin[/db2]$

```

Figure 170. DB2 List Node Directory on DB2 Client

The Service name db2tivoli defines the port used to contact the DB2 server and is specified in the /etc/services file.

B.4 Connecting to a DB2 Database

Use the following command to connect to a DB2 database:

```
db2 connect to db_alias user user_id using password
```

In this case, db_alias refers to the alias name in the database directory, user_id to the user ID that should be used to connect to the database and password the password associated with the user ID.

```

D:\SQLLIB\BIN>db2 connect to sample user db2 using db2

Database Connection Information

Database product      = DB2/NT 2.1.2
SQL authorization ID = DB2
Local database alias = SAMPLE

D:\SQLLIB\BIN>

```

Figure 171. DB2 Connect on DB2 Client

The database name is SAMPLE on the DB2 client. The DB2 user name is db2 and the password is db2.

B.5 Listing the Table Names Created in a DB2 Database

Use the following command to list the tables created in a DB2 database:

```
db2 list tables
```

After the DB2 client connects to a database, this command will list all the tables created by the user connected to the database.

```
D:\SQLLIB\BIN> db2 list tables
```

NAME	CREATOR	TYPE	CTIME
DEPARTMENT	DB2	T	1998-04-14-11.01.36.040303
EMP_ACT	DB2	T	1998-04-14-11.01.39.020707
EMP_PHOTO	DB2	T	1998-04-14-11.01.43.030303
EMP_RESUME	DB2	T	1998-04-14-11.01.49.000202
EMPLOYEE	DB2	T	1998-04-14-11.01.37.010404
ORG	DB2	T	1998-04-14-11.01.34.030000
PROJECT	DB2	T	1998-04-14-11.01.42.020202
STAFF	DB2	T	1998-04-14-11.01.34.090101

```
8 record(s) selected.
```

```
D:\SQLLIB\BIN>
```

Figure 172. DB2 List Tables

Use the following command to list all the tables created in a DB2 database:

```
db2 list tables for all
```

This command will list all the table created in the database regardless of who created them.

B.6 Executing SQL Statements

Use the following command to execute SQL statements:

```
db2 "sql_statement"
```

Any statement following the db2 keyword will be executed as a DB2 command. If, however, it is not found to be an DB2 command, an attempt will be made to execute it as a SQL statement. If the SQL statement contains

special characters like %,|,> or < the statement should be encapsulated in double quotes (").

B.7 Listing the Column Names in a DB2 Database Table

The following DB2 command lists the column names in a DB2 database table:

```
db2 select colname,typename,length,nulls from syscat.columns where  
tabname='xxxxxxx'
```

```
D:\SQLLIB\BIN> db2 select colname,typename,length,nulls from syscat.columns where  
tabname='EMPLOYEE'
```

COLNAME	TYPENAME	LENGTH	NULLS
EMPNO	CHARACTER		6 N
FIRSTNAME	VARCHAR		12 N
MIDINIT	CHARACTER		1 N
LASTNAME	VARCHAR		15 N
WORKDEPT	CHARACTER		3 Y
PHONENO	CHARACTER		4 Y
HIREDATE	DATE		4 Y
JOB	CHARACTER		8 Y
EDLEVEL	SMALLINT		2 N
SEX	CHARACTER		1 Y
BIRTHDATE	DATE		4 Y
SALARY	DECIMAL		9 Y
BONUS	DECIMAL		9 Y
COMM	DECIMAL		9 Y

```
14 record(s) selected.  
D:\SQLLIB\BIN>
```

Figure 173. DB2 List Column Names for Database Table

This SQL statement lists all the column names, their types, lengths and whether they allow nulls or not.

B.8 Listing Applications Connected to DB2 Database

Use the following command to list all the applications connected to any of the DB2 databases within an instance:

```
db2 list applications
```

All three connections in Figure 174 on page 175 are made to the sample database with the user ID db2admin.

```
db2admin@bombay[/db2]$ db2 list applications

Auth Id  Application Name      Agent Id
-----  -
Application Id      DB Name

-----  -
-----  -

DB2ADMIN db2bp_41      24784
*LOCAL.db2admin.980417154104  SAMPLE
DB2ADMIN db2bp_41      27296
*TCP/IP.0903014C.980417154520  SAMPLE
DB2ADMIN RIM_DB2_prog  22442
*TCP/IP.0903014C.980417154605  SAMPLE

db2admin@bombay[/db2]$
```

Figure 174. DB2 List Applications

The first connection is a local connection, whereas the other two connections are remote connections using the TCP/IP protocol. The first two connections are connections from the command line of an AIX machine. The last connection is an application that connected to the database. In fact, RIM is connected to the DB2 database SAMPLE through the Tivoli RIM application program RIM_DB2_prog.

B.9 Forcing Applications to Disconnect from a DB2 Database.

Use the following command to force an application or applications to disconnect from a DB2 database and rollback any uncommitted data changes associated with the application(s):

```
db2 force applications [all | agentid xxxxxx ]
```

By using the `all` option, all the applications are forced to disconnect. The agent ID of an application is used to identify the application to disconnect. You can issue the `list applications` command to find a application's agent ID.

B.10 Deleting a DB2 Database

Use the following command to delete and remove a DB2 database from the database directory.

```
db2 drop database database_name
```

This operation is irreversible.

B.11 Listing All DB2 Instances on an Machine

The `db2list` command displays all the DB2 instances defined on the machine where the command is executed

```
D:\SQLLIB\BIN> db2list  
  
DB2  
DB2ADMIN  
  
D:\SQLLIB\BIN>
```

Figure 175. DB2 List Instances

In our setup, we have two DB2 database instances defined DB2 and DB2ADMIN.

B.12 Switching DB2 Instances from the Command Line

Use the following command to switch DB2 instance from the command line:

```
set DB2INSTANCE=instance_name on NT  
  
export DB2INSTANCE=instance_name on AIX
```

The `DB2INSTANCE` variable is used to determine at which instance a command is directed. By changing it, any command issued will be directed to the instance pointed to by the variable.

B.13 Starting and Stopping an DB2 Instance

Use the following command to stop DB2 instance from the command line:

```
db2stop
```

Use the following command to start DB2 instance from the command line:

```
db2start
```

These commands can only be executed by the DB2 System Administrative group. It will execute on the instance referred to by the `DB2INSTANCE` environment variable.

Appendix C. Special Notices

This publication is intended to help technical professionals that must install and configure RDBMS products for use with Tivoli TME 10 applications. The information in this publication is not intended as the specification of any programming interfaces that are provided by TME 10 Framework, TME 10 Inventory, or TME 10 Enterprise Console. See the Tivoli Web page at URL http://corp.tivoli.com/widow/Tivoli_Documents/index.htm 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.

You can reproduce a page in this document as a transparency, if that page has the copyright notice on it. The copyright notice must appear on each page being reproduced.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIE	DB2
DB2 Connect	DB2 Universal Database
DRDA	Distributed Relational Database Architecture
IBM	MVS
OS/2	OS/400
RISC System/6000	

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 D. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

D.1 International Technical Support Organization Publications

For information on ordering these ITSO publications, see “How to Get ITSO Redbooks” on page 183.

- *An Introduction to Tivoli's TME 10*, SG24-4948
- *TME 10 Framework Version 3.2: An Introduction to the Lightweight Client Framework*, SG24-2025
- *TME 10 Deployment Cookbook: Inventory and Company*, SG24-2120
- *TME 10 Inventory 3.2: New Features and Database Support*, SG24-2135
- *TME 10 Cookbook for AIX: Systems Management and Networking Applications*, SG24-4869
- *TME 10 Internals and Problem Determination*, SG24-2034

An Introduction to Tivoli's TME 10 is also published by Prentice Hall (ISBN 0-13-899717-9) and available in most bookstores.

D.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
IBM Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbooks Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PDF)	SBOF-8700	SK2T-8043
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

D.3 Other Publications

These publications which are shipped with the Tivoli TME 10 products are also relevant as further information sources:

- *Tivoli Framework Planning and Installation Guide*
- *Tivoli Framework User's Guide*
- *Tivoli Framework Reference Manual*
- *Tivoli Framework Release Notes*
- *Tivoli Enterprise Console User's Guide*
- *Tivoli Enterprise Console Reference Manual*
- *Tivoli Enterprise Console Release Notes*
- *Tivoli Inventory User's Guide*
- *Tivoli Inventory Release Notes*

The publications are available from the Oracle installation CD-ROM. The publications we used were:

- *Oracle7 Server Getting Started (for Windows NT)*
- *Oracle Networking Products User's Guide (for Windows NT)*
- *SQL*Plus User's Guide (for Windows NT/95)*
- *Oracle Network Products Troubleshooting Guide (Release 2.3)*
- *Oracle Network Manager Administrator's Guide (Release 3.1)*

These publications are available online after DB2 is installed. The publications we used were:

- *DB2 Administration Guide*
- *DB2 Command Reference*
- *DB2 Information and Concepts Guide*
- *DB2 Message Reference*
- *DB2 Problem Determination Guide*
- *DB2 SQL Reference*
- *DB2 Glossary*
- *DB2 Installation and Operation Guide*
- *DB2 Installing and Using AIX Clients*
- *DB2 Planning Guide*

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:

- **Redbooks Web Site on the World Wide Web**

<http://w3.itso.ibm.com/>

- **PUBORDER** – to order hardcopies in the United States

- **Tools Disks**

To get LIST3820s of redbooks, type one of the following commands:

```
TOOLCAT REDPRINT
TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
```

To get BokkManager BOOKs of redbooks, type the following command:

```
TOOLCAT REDBOOKS
```

To get lists of redbooks, type the following command:

```
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
```

To register for information on workshops, residencies, and redbooks, type the following command:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1998
```

- **REDBOOKS Category on INEWS**

- **Online** – send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL

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.html>). 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:

	IBMMAIL	Internet
In United States	usib6fpl at ibmmail	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 Publications Customer Support P.O. Box 29570 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 Allerød Denmark
--	--	--

- **Fax** – send orders to:

United States (toll free)	1-800-445-9269
Canada	1-800-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States) or (+1) 408 256 5422 (Outside USA)** – ask for:

Index # 4421 Abstracts of new redbooks
Index # 4422 IBM redbooks
Index # 4420 Redbooks for last six months

- **On the World Wide Web**

Redbooks Web Site	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibm.com/pbl/pbl

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.html>). 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.

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 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 and Tivoli software, IBM and Tivoli product terminology, as well as selected terms from other sources.

Below are terms used in this document:

administrative authority A level of authority giving a wide ranging set of privileges over a set of objects, such as DBADM, which provides privileges over all objects in a database, or SYSADM, which provides privileges over all objects in a system.

alias An alternate name used to identify a database, that is, database alias.

base table A table created with the CREATE TABLE statement. Such a table has both its description and data physically stored in the database. A base table can be referred to by an alias or a view definition.

base view A view created with the CREATE VIEW statement. Such a view has its description stored in the database. A base view can be referred to by an alias or another view definition.

client Any program (or workstation it is running on) that communicates with and accesses a database server.

client/server An environment where there is a separation of data from applications. The client workstation (where applications are executed) requests data from a database server (where the data resides). Thus, there is a separation of tasks between the client workstations and database servers.

commit The operation that ends a unit of work by releasing locks so that database changes made by that unit of work can be perceived by other processes. This SQL statement allows the data changes to be made permanent.

Data Definition Language (DDL) A subset of the SQL commands used to define, modify, and remove database objects and attributes. For example, CREATE TABLE.

data manipulation language (DML) A series of SQL commands used to manipulate data. For example, SELECT.

data type An attribute for defining data that describes the type of values accepted and imposes certain limits inherent to that type.

database administrator (DBADM) A user with database administrator authority. The database administrator is usually responsible for the database integrity, security, performance and recovery.

database connection A logical connection between an application and an agent process or thread associated with a specific database in a DBMS instance.

database object Anything that can be created or manipulated with SQL. For example, tables, views, indexes, or table spaces.

deadlock A situation where two or more transactions are each waiting for data locked by the other, in such a way that the situation could never be resolved.

Distributed Database Connection Services (DCS) A product that provides the function necessary (DRDA application requester support) for client applications to read and update data stored in DRDA application servers, such as other members of the DB2 family.

Distributed Relational Database Architecture (DRDA) DRDA is an architecture that defines formats and protocols for providing transparent access to remote data. DRDA defines two types

of functions, the application requester function and the application server function.

instance A logical database manager environment similar to an image of the actual database manager environment. You can have several instances of the database manager product on the same workstation.

instance home directory The home directory of the instance owner.

instance name A name associated with the instance. For DB2, the name comes from the userid of the owner of the instance.

Inter-Process Communication (IPC) An operating-system-provided mechanism that allows processes to communicate with each other.

named pipe An IPC mechanism that allows non-related processes to communicate. The access methods are very similar to file I/O.

ODBC driver An ODBC driver is a dynamically-linked library (DLL) that implements ODBC function calls and interacts with a data source.

Open Database Connectivity (ODBC) A Microsoft developed C database API that allows access to database management systems using callable SQL, which does not require the use of a SQL preprocessor. In addition, ODBC provides an architecture which allows users to add modules called database drivers that link the application to their choice of database management systems at run time. This means applications no longer need to be directly linked to the modules of all the database management systems that are supported.

optimizer A component of the SQL compiler that chooses an access plan for an SQL DML statement by modeling the execution cost of many alternative access plans and choosing the one with the minimal estimated cost.

panel A particular arrangement of information grouped together for presentation to the user.

predicate A predicate specifies a condition that is true, false or unknown about a given row, expression or group.

roll back The process of restoring data changed by SQL statements to the state at its last commit point.

schema A logical grouping for database objects. When a database object is created it is assigned to one schema, which is determined by the name of the object. For example, the following creates table X in schema C:

```
CREATE TABLE C.X
```

server On a network, the computer that contains the data or provides the facilities to be accessed by other computers on the network.

socket An endpoint for communication that can be named and addressed in a network. In TCP/IP, a socket address is a unique identifier created by combining the IP address and the port number.

Structured Query Language (SQL) An established set of statements used to manage information stored in a database. By using these statements, users can add, delete, or update information in a table, request information through a query, and display the results in a report.

system administrator (SYSADM) A user with SYSADM authority. SYSADM authority allows a user to perform any operation and gives the user DBADM authority over all databases managed by the instance.

VARCHAR A column data type where the character data varies in the number of characters. No specific number of characters are reserved for each entry.

window An area of the screen with visible boundaries through which a panel or portion of a panel is displayed.

List of Abbreviations

ACC	AutoPack Control Center	DDCS	Distributed Database Connection Services (DB2)
ACF	TEC Adapter Configuration Facility	DES	Data Encryption Standard
ADE	Application Development Environment	DFW	Desktop for Windows
AEF	Application Extension Facility	DHCP	Dynamic Host Configuration Protocol
AMS	Application Management Specification	DII	Dynamic Invocation Interface
ALI	Authentication, Location and Inheritance	DLL	Dynamic Link Library
API	Application Programming Interface	DM	Distribution Manager (or Management)
BAROC	Basic Recorder of Objects in C	DMI	Desktop Management Interface
BDC	Backup Domain Controller (Microsoft)	DMTF	Desktop Management Task Force
BDT	Bulk Data Transfer	DO	Display Object
BO	Behavior Object	DPO	Default Policy Object
BOA	Basic Object Adapter	DRDA	Distributed Relational Database Architecture (DB2)
CAE	Client Application Enabler (DB2)	DSL	Dialog Specification Language
CCMS	Configuration and Change Management System	EIF	Event Integration Facility
CDF	Component Description File	ESM	Enterprise Server Management
CLI	Command Line Interface	FAT	File Allocation Table
CORBA	Common Object Request Broker Architecture	FFST	First Failure Support Technology
DBA	Database Administrator	FSFI	Free Space Fragmentation Index
		GB	gigabytes
		GID	Group Identifier (UNIX)
		GCOS	GE Computer Operating System field

GEM	Global Enterprise Manager	ODBC	Open Database Connectivity (Microsoft, based on X-Open CLI)
IBM	International Business Machines Corporation	OID	Object Identifier
IDL	Interface Definition Language	OMG	Object Management Group
IIS	Microsoft Internet Information Server	ORB	Object Request Broker
IMS	Internet Management Specification	PDF	Portable Document File; Product Description File
IOM	Inter-Object Messaging	PD/PSI	Problem Determination/Problem Source Isolation
IPC	Inter-Process Communication	PDC	Primary Domain Controller (Microsoft)
ITSO	International Technical Support Organization	PO	Prototype Object
JMAPI	Java Management Application Programming Interface	RACF	Resource Access Control Facility
KB	kilobytes	RCS	Revision control System
LCF	Lightweight Client Framework	RDBMS	Relational Database Management System
MB	megabytes	RIM	RDBMS Interface Module
MCSL	Monitoring Capabilities Subscription Language	RPC	Remote Procedure Call
MDist	Multiplexed Distribution	SAM	Security Account Manager
MIF	Management Information File	SDK	Software Developer's Kit (DB2)
MLM	Mid-Level Manager	SID	System Identifier (Oracle)
MSB	Multiple Services Broker	SIS	Software Installation Service
MTS	Multi-thread Server (Oracle)	SMP	Symmetrical Multiprocessor
NFS	Network File System	TACF	Tivoli Access Control Facility
NIS	Network Information Services (formerly "NFS yellow pages")	TEC	Tivoli Enterprise Console
NLS	National Language Support	TEIDL	Tivoli-Extended IDL
NTFS	NT Files System		

<i>TLI</i>	Transport Layer Interface
<i>TLL</i>	Task Library Language
<i>TME</i>	Tivoli Management Environment
<i>TMF</i>	Tivoli Management Framework
<i>TMR</i>	Tivoli Management Region
<i>TNR</i>	Tivoli Name Registry
<i>TNS</i>	Transparent Network Substrate (Oracle)
<i>TNWR</i>	Tivoli NetWare Repeater
<i>TRAA</i>	Tivoli Remote Access Account
<i>TRIP</i>	Acronym for Tivoli Remote Execution Service
<i>UID</i>	User Identifier (UNIX)
<i>UNC</i>	Universal Naming Convention
<i>VPO</i>	Validation Policy Object
<i>XBO</i>	Extended Behavior Object

Index

Symbols

\$BINDIR/TAS/RIM 3
\$HOME directory 60
\$HOME/.profile 45
\$HOME/sqlib 45, 46, 49
\$LIBDIR directory 60
\$ORACLE_HOME 80
\$ORACLE_HOME/bin 68, 78, 81, 86
\$ORACLE_HOME/network/admin 92, 93, 96, 100
\$ORACLE_HOME/orainst 80, 85
\$SYBASE directory 162
/usr/lpp/db2_02_01 59, 61
@alias 124

A

abbreviations 189, 190, 191
acronyms 189, 190, 191
administrator 69
Administrator's Toolkit
 DB2 for NT client 47
 DB2 for NT server 37
Advanced Replication Option 78
alias name 54, 172
 without domain 124
Application User option 91
asynchronous I/O 165
authorization error 126

B

bos.adt.base 68

C

CAE 46
CDE 75
CHARACTER 159
chown 72
client configuration utility 53
column function 150
command termination character 162
COMMUNITY 102
components
 Oracle for AIX 84
 Oracle for NT 88
configure database client

 DB2 for AIX 57
 DB2 for NT 51
configure database server
 DB2 for AIX 56
 DB2 for NT 50
configure SQL*Net client
 AIX 100, 118
 NT 103, 116
configure SQL*Net server
 AIX 99
 NT 102
connect to database server
 DB2 for AIX 46
 DB2 for NT 37
 Oracle for AIX 119
 Oracle for NT 109
ConText Option 89
CORENT23.DLL 86
create catalog views
 DB2 156
 Oracle 159
create database
 DB2 for AIX 44, 46
 DB2 for NT 37
 Oracle for AIX 84
 Oracle for NT 89
create DB2 instance
 AIX 45
create file system 70
create inittab entry 81
create logical link 60
create query
 CLI 154
 GUI 145
create query administrator 143
create query library 144
create RIM object
 DB2 for AIX 61
 DB2 for NT 60
 Oracle 111
 Oracle for AIX 120
 Oracle for NT 113
create user
 DB2 for AIX 31
 DB2 for NT 26
 Oracle for AIX 69
create user group

- DB2 for AIX 29
- DB2 for NT 26
- Oracle for AIX 69
- created CDRROM file system 74

D

- data types 158
- Database Administrator option 91
- Database Alias 117
- database alias name 104
- database and system shutdown script 82
- database backup 141
- database domain name 80
- database home directory 72
- Database Home parameter 63
- database instance 54, 101, 176
 - DB2 for NT 38
 - not started 125
- database instance name 104, 112, 113, 117, 119, 120
 - DB2 for AIX 45
- database instance owner
 - DB2 for AIX 40, 45
- database object size 141
- database server 62
- database user 16
- database user password 61
- DATE 159
- db_setup.sh 72, 73, 85
- DB2
 - administrator user 26
 - Administrator's Toolkit 37, 47
 - Client Application Enabler 35, 42
 - Code Page Conversions 42
 - Command Line Processor 42
 - Communications Support 42
 - configure database client
 - AIX 57
 - NT 51
 - configure database server
 - AIX 56
 - NT 50
 - create sample database
 - AIX 46
 - NT 37
 - database 24
 - Database Director 42
 - database instance 23, 38

- DDCS Multi-User Gateway 34
- define user
 - AIX 31
 - NT 27
- define user group
 - AIX 29
 - NT 28
- disconnect database 38
- Executables 42
- install database client
 - AIX 48
 - NT 46
- install database server
 - AIX 40
 - NT 34
- installation error 34
- licence key 35
- ODBC Driver 37, 55
- Performance Monitor 42
- Product Library - Postscript (documentation) 43
- Product Library (on-line documentation) 43
- sample database (ORCL) 160
- SDK Utilities and Samples 43
- security server 39
- Software Developer's Kit (SDK) 34
- start database instance 38, 40
- stop database instance 38
- supported versions 23
- tables 24
- Utilities and Samples 43
- World Wide Web Connection 34
- db2 60, 66, 128, 157, 158
 - ? 167
 - ? error code 169
 - catalog 58
 - connect 37, 46, 58, 172
 - drop 175
 - execute SQL statements 173
 - force 175
 - help 169
 - list 37, 46
 - list applications 174
 - list database directory 170
 - list node directory 171
 - list tables 173
 - select 174
 - update 50, 56
- db2 directory 32, 60, 61, 63, 65, 66
- DB2 Security Server 126

DB2 Universal Database Version 5 23
db2admin 24, 28, 29, 32, 33, 44, 45, 49, 56, 57,
58, 61, 65
DB2COMM variable 45
db2icrt 45, 49
db2ilist 45
DB2INSTANCE 176
db2list 176
db2profile 45, 49
db2sampl 37, 46
db2start 37, 45, 51, 57, 125, 176
db2stop 38, 51, 57, 176
DBA group 77
dba group 123
dbshut 82, 95
DBSNMP 159
dbstart 81, 96
DECIMAL 159
default alias 54
default database user 16
define roles 11
delete views 157
disconnect database
 DB2 for NT 38
Distributed Database Option 77
DLLs 86
DNS 52
domain 101
DRDA 34
DSQUERY 161

E

EN_US 34
Enterprise Manager 90
environment variables
 DB2 for AIX 45
 Oracle for AIX 72
ERROR 7, 18, 136
execute SQL scripts
 DB2 158
 Oracle 68

F

FLOAT 159
free space 140

G

get method environment 19
group
 sysadm 26, 28, 29, 30

H

home directory 87
HOST entry 99, 102
host name 120
Host String 110
hosts file 52, 54, 58

I

IBM DB2 1, 5
idcall 8
IN condition 152
INFORMATION 7, 18, 136
inittab file 81
install database client
 DB2 for AIX 48
 DB2 for NT 46
 Oracle for AIX 83
 Oracle for NT 91
install database server
 DB2 for AIX 40
 DB2 for NT 34
 Oracle for AIX 68
 Oracle for NT 86
instance home parameter 65, 113
instance name 24, 27
instance owner 24, 26, 27, 49, 55, 158
INTEGER 159
interfaces directory 162
interfaces file 163
IOM channel 4
isql 162
isql_w 165

L

libdb2.a 60
LIKE condition 152
link directory 72, 74
List Tables
 DB2 for NT 37
listener process 68
listener.ora 96, 99
log file 98

lsnrctl
 show current_listener 96
 start 95, 100
 status 94

M

method environment
 get 19
 set 20
Microsoft SQL Server 5
mkitab 81
mount 71
mount CDROM file system 74
mount file system 71
mount point 71
MS SQL Server 1, 165
MS_SQL 21, 166
MSVCRT10.DL 86

N

named pipes 165
NLS_LANG parameter 87

O

object dispatcher 20
odadmin
 environ 19, 138
ODBC Driver 37, 55
odstat 6
OPERATOR group 77
ORA_NLS32 80
ORA-01017 15
ORA-03121 124
ORA-12154 123
Oracle 1, 5, 6, 67
 Call Interface 88
 components 88
 Enterprise Manager 89
 Web Administration Tools 88
 install database client
 AIX 83
 NT 91
 install database server
 AIX 68
 NT 86
 Installer 75, 86, 87, 89, 91, 104
 Intelligent Agent 89

optimizer 140
Server (RDBMS) 79
service name 93
services 90, 102
starter database 89, 160
stop database instance 82
supported versions 67
 Webserver 88
ORACLE_BASE 77, 84
ORACLE_HOME 68, 73, 77, 80, 84, 112
ORACLE_OWNER 73
ORACLE_SID 68, 73, 77, 112
Oracle7
 Client 91
 Performance Utility 88
 Server 88
 Server Options 88
 Utilities 88
Oracle8 67
OracleServiceORCL 91, 103
OracleStartORCL 91, 103
OracleTNSListener 91, 103
orainst 74, 75, 87, 91, 92
orainst directory 74, 75, 80, 85
ORANT71.DLL 86
oratab file 80
ORCL 89, 103, 104, 110, 120, 160

P

Parallel Query Option 78
passwd 33
password 68, 70, 80, 90, 108, 110, 113, 121
PATH 68, 73, 86
PORT 99, 102, 119
port number 50, 51, 54, 57, 58, 70, 93, 100
predicate 151, 152, 155
PROTOCOL 99, 102
protocol adapter 77, 90, 91

Q

query facility 143
query icon 154
QUERY_DOMAINS 157, 160
query_edit 12, 143
query_execute 12, 143
query_view 12, 143
QUERY_VIEWS 157, 159

R

RDBMS home directory 112, 113, 120
RDBMS server call failed 136
repository 148
RIM 6, 8
 object 9
 Object Methods 10
 Recommend Configuration 8
RIM (RDBMS Interface Module) 1
RIM error log 19
RIM host 60, 62
 AIX 61
RIM name 112
RIM object 9, 11, 60
 DB2 58
 MS SQL Server 165
 Oracle 59, 111
 Sybase 161
RIM repository 143
RIM test utility 127
 commit 135
 delete rows 133
 execute SQL statements 131
 insert rows 131
 list help 127
 retrieve rows 128
 rollback 135
 update rows 134
 use editor 130, 133, 134
RIM trace facility 135
RIM_DB_LOG 19
RIM_DB2_prog 137, 175
RIM_update 12
RIM_view 12
root.sh 80, 85
rootpre.sh 75, 77, 84
run query 154

S

security server
 DB2 for NT 39
Server ID 93, 112, 113, 122
server ID 120
service name 54, 106, 171, 172
services file 51, 56, 58, 70, 93, 99, 102, 171
set method environment 19
SID 119
SMALLINT 159

Software Asset Manager 79, 84, 87, 91
Spatial Data Option 78, 89
SQL interpreter 162
SQL*Net 68, 79, 84, 92
 Easy Configuration 92, 104, 116
 listener 80, 93, 95, 96
 listener log file 98
 Protocol Adapter 89
 Server 89
SQL*Plus 68, 79, 85, 89, 107, 109, 120, 159
SQL.INI 165
SQL1013N 139
SQL1032N 125
SQL1402N 126
SQL5043N 46
sqllib directory 59, 60, 63
sqlnet.ora 100, 101, 103
sqlplus 68, 108
 ANALYZE 140
 system 140
start database instance
 DB2 for NT 38, 40
start.sh 74
starter database 110
stop database instance
 DB2 38
 Oracle 82
supported databases
 DB2 23
 MS SQL Server 166
 Oracle 67
 Sybase 162
SVCENAME parameter 50, 56
switch DB2 instance 176
SYBASE 161
Sybase 1, 5, 161
Sybase 10 162
Sybase 11 162
Sybase for AIX 163
Sybase for NT 164
Sybase interfaces file 162
SYS 159
SYSADM authority 24
SYSADM group 24, 28, 29, 30
SYSADM user 24, 34, 50
SYSCAT 25
SYSCAT.COLDIST 157, 160
SYSCAT.COLUMNS 157
SYSCAT.TABLES 157

SYSIBM 25
SYSMAN 159
SYSSTAT 25
SYSTEM 159

T

table name 24, 149
table reference 24
Table/View Name 148
TCP/IP Protocol Adapter 85
TERM 75
test database connection 55
test RIM object
 AIX 62, 114, 121
 NT 62
TMR administrator 11
TNS listener 93
TNS ping utility 106
tnslsnr 100
tnsnames.ora 90, 92, 100, 101, 103, 105, 106, 118
tnsping 106, 108, 119
topology.ora 90
trace file 138
trace level 18, 21, 136
TRACE_OFF 7, 18, 136
TWO_TASK 68, 93, 112

U

umask 70
US7ASCII 80
user
 Administrator 26
 db2 26
 db2admin 28, 32, 58
 oper 80
 oracle 79, 80, 81, 95, 99, 100, 123
 root 45, 56, 58
 SYS 90
 sys 15, 80, 108
 system 80, 121, 140
 tivoli 16
user group 79
user ID 24, 172
user name 110, 112, 113, 120, 124, 128
USER_TAB_COLUMNS 157, 159

V

VARCHAR 158, 159

W

wcrquery 155, 156
wcrtrim 13, 58, 60, 61, 111, 114, 120
wdel 16
wgetrim 14, 113, 114, 121
where clause 151
wlookup 16
wregister 16
wrimtest 8, 17, 62, 114, 115, 127
wrimtrace 7, 18, 135
wsetrim 14
wsetrimpw 15
wtrace 7

X

xterm 75

ITSO Redbook Evaluation

Using Databases with Tivoli Applications and RIM
SG24-5112-00

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.ibm.com>
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?

- Customer** **Business Partner** **Independent Software Vendor** **IBM employee**
 None of the above

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!)

