Tivoli software

IBM

Certification Study Guide Series: IBM Tivoli Application Dependency Discovery Manager V7.1

- Helps you achieve TADDM V7.1 certification
- Explains the certification path and prerequisites
- Introduces sample test questions

Vasfi Gucer
Christian Fernando Hevia
David Stephenson
Ghufran Shah
Linda E. Miller-Plumley
Petra Unglaub
Seda Isi

ibm.com/redbooks

Redbooks

International Technical Support Organization

Certification Study Guide Series: IBM Tivoli Application Dependency Discovery Manager V7.1

August 2009

SG24-7764-00

Note: Before using this information and the product it supports, read the information in “Notices” on page ix.

First Edition (August 2009)

This edition applies to IBM Tivoli Application Dependency Discovery Manager V7.1.

Contents

Notices ix
Trademarks x

Preface xi
The team who wrote this book xii
Become a published author xiv
Comments welcome xv

Chapter 1. Certification overview 1
1.1 IBM Professional Certification Program 2
1.1.1 Benefits of certification 3
1.2 Tivoli Software Professional Certification 4
1.3 Test 000-011: IBM Tivoli Application Dependency and Discovery Manager V7.1 Implementation 7
1.3.1 Job role description and target audience: 7
1.3.2 About the test 7
1.3.3 Receive your 15% discount when taking the test 8
1.4 Recommended resources for study 8
1.4.1 Courses 8
1.4.2 Publications 8

Chapter 2. Planning 11
2.1 Your environment 12
2.1.1 Hardware prerequisites 12
2.1.2 Operating system prerequisites 13
2.1.3 Supported database versions 22
2.2 Requirements analysis 22
2.2.1 Documenting your existing server environment 22
2.2.2 Use of lsof utility 22
2.2.3 Documenting your existing network environment 23
2.2.4 Documenting your existing application environment 25
2.3 Planning the TADDM topology 25
2.3.1 Federating with eCMDB 25
2.3.2 Maximum number of configuration items 27
2.4 Directory Services integration 27
2.4.1 User registry 27
2.4.2 LDAP configuration 28
2.5 Discovery Library Adapters 28
2.6 Sensors and discovery 29

2.6.1 Discovery components 29
2.6.2 Level 1 discovery 31
2.6.3 Level 2 discovery 32
2.6.4 Level 3 discovery 34
2.6.5 Sensor flow 34
2.7 Controlling discovery 39
2.8 Common Data Model 41

Chapter 3. Installation 47
3.1 Installation overview 48
3.2 Prerequisite tasks 49
3.2.1 Using a local DB2 database 49
3.2.2 Using a remote DB2 database 50
3.3 Simple installation 50
3.4 Advanced installation 55
3.5 Silent installation 62
3.6 Anchor and gateway installation 64
3.6.1 Anchor considerations 65
3.6.2 Gateway considerations 66

Chapter 4. Configuration 67
4.1 Performance tuning for databases 68
4.1.1 RUNSTATS command 68
4.1.2 Query optimizer 69
4.2 Discovery scopes 69
4.2.1 Adding a scope set 70
4.2.2 Deleting a scope set 70
4.2.3 Adding a scope 70
4.2.4 Editing a scope 71
4.2.5 Deleting a scope 71
4.2.6 Loading a scope set from a file 71
4.3 Access list 73
4.3.1 Adding an access list entry 74
4.3.2 Editing an access list entry 75
4.3.3 Deleting an access list entry 75
4.3.4 Changing the order of the access list entries 75
4.4 Discovery profiles 76
4.4.1 Creating a discovery profile 76
4.4.2 Editing a discovery profile 77
4.4.3 Deleting a discovery profile 77
4.5 Anchors and gateways 77
4.5.1 Adding an anchor or gateway 79
4.5.2 Editing an anchor or gateway 80

4.5.3 Deleting an anchor or gateway 80
4.5.4 Setting an anchor port 80
4.6 Custom server templates 81
4.6.1 Adding a custom server 81
4.6.2 Editing a custom server template 85
4.6.3 Copying a custom server template 85
4.6.4 Deleting a custom server template 85
4.6.5 Changing the order of the custom server templates 86
4.7 Application templates 86
4.7.1 Adding an application template 86
4.7.2 Editing an application template 88
4.7.3 Deleting an application template 88
4.8 Application descriptors 88
4.8.1 Base application descriptor 89
4.8.2 Component application descriptor 91
4.8.3 Application descriptor locations 93
4.9 Security 96
4.9.1 File authentication 96
4.9.2 Configuring for LDAP 99
4.9.3 Configuring for WebSphere federated repositories 100
4.9.4 Enterprise Domain Server 103

Chapter 5. Discovery 105
5.1 Running discoveries 106
5.1.1 Defining a scope 106
5.1.2 Configuring the access list 106
5.1.3 Adding credentials 107
5.1.4 Discovery schedules 107
5.1.5 Running a basic discovery 110
5.1.6 Running a discovery from the command line with api.sh 111
5.1.7 Viewing the discovery history 111
5.2 Creating and managing custom servers 113
5.2.1 Identifying unknown server patterns 113
5.2.2 Adding custom servers 114
5.2.3 Editing a custom server 118
5.2.4 Copying a custom server 118
5.2.5 Deleting a custom server 118
5.2.6 Repositioning custom server entries 119
5.3 Using discovery profiles 120
5.3.1 Creating discovery profiles 121
5.4 Access collections 123
5.4.1 Creating an access collection 124
5.4.2 Editing an access collection 125

5.4.3 Deleting an access collection 125
5.5 Discovering business applications and business services 126
5.6 The TADDM Enterprise Domain Manager 127
5.6.1 TADDM Enterprise Domain Manager overview 127
5.6.2 Exploring the TADDM Enterprise Domain Manager Console 128

Chapter 6. Problem Determination 135
6.1 Log files 136
6.1.1 SplitSensor logging 138
6.1.2 Dynamic logging 139
6.1.3 Extra debugging 139
6.1.4 Collecting data for IBM 139
6.2 Memory issues 140
6.3 Name resolution issues 141
6.3.1 Full Qualified Domain Names (FQDN) 141
6.4 Access and discovery issues 141
6.4.1 Testing the connection 142
6.4.2 WebSphere discovery 142
6.4.3 StackScan 143
6.4.4 Database connectivity 144
6.4.5 Mapping 144
6.4.6 Scopes 145
6.4.7 Database configurations 145
6.4.8 Expired password 145
6.4.9 rmi.clientproxy.server.hostname parameter 146
6.4.10 sslpassphrase 146
6.5 Relationships 147

Chapter 7. Administration 149
7.1 Manually starting and stopping the TADDM server 150
7.1.1 Starting the TADDM server 150
7.1.2 Restarting the TADDM server 151
7.1.3 Stopping the TADDM server 151
7.1.4 Testing the TADDM server status 152
7.2 Updating the database passwords 153
7.3 Discovery schedules 154
7.3.1 Adding a discovery schedule 154
7.3.2 Viewing discovery schedule details 156
7.3.3 Deleting a discovery schedule 156
7.4 Synchronization schedules 157
7.4.1 Adding a synchronization schedule 157
7.4.2 Viewing synchronization details 159
7.4.3 Deleting a synchronization schedule 159

7.5 Versions 159
7.5.1 Adding a version 160
7.5.2 Viewing a version 161
7.5.3 Deleting a version 161
7.6 Manual component creation 161
7.6.1 Adding a component 161
7.6.2 Editing a component 163
7.6.3 Deleting a component 163
7.7 Manual dependency creation 163
7.7.1 Adding a dependency 164
7.7.2 Viewing dependency details 166
7.7.3 Deleting a dependency 166
7.8 Business applications and business services 166
7.8.1 Adding a business application or a business service 167
7.8.2 Viewing business application or business service details 169
7.8.3 Viewing business application or business service topology 169
7.8.4 Editing a business application or a business service 169
7.8.5 Deleting a business application or a business service 170
7.9 Roles and permissions 170
7.9.1 Adding a role 171
7.9.2 Deleting a role 172
7.10 Application programming interface 173
7.10.1 Find command 174
7.10.2 Discover command 176
7.10.3 Topology command 178
7.10.4 Changes command 179
7.10.5 Version command 180
7.10.6 Delete command 182
7.10.7 Import command 183
7.10.8 Export command 184
7.10.9 Naming command 185

Appendix A. Sample certification test questions 187
Questions 188
Answers 193

Related publications 195
IBM Redbooks 195
Online resources 195
How to get Redbooks 196
Help from IBM 196

Index 197

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include 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.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

Preface

This IBM® Redbooks® publication is a study guide for IBM Tivoli® Application Dependency Discovery Manager (TADDM) V7.1 and is aimed at individuals who want to get an IBM Professional Certification for this product.

The IBM Tivoli Application Dependency Discovery Manager V7.1 Professional Certification, offered through the Professional Certification Program from IBM, is designed to validate the skills required of technical professionals who work in the implementation of the IBM Tivoli Application Dependency Discovery Manager V7.1 product.

This book provides a combination of theory and practical experience needed for a general understanding of the subject matter. It also provides sample questions that will help in the evaluation of personal progress and provide familiarity with the types of questions that you will encounter in the exam.

This publication does not replace practical experience, nor is it designed to be a stand-alone guide for any subject. Instead, it is an effective tool that, when combined with educational activities and experience, can be an extremely useful preparation guide for the exam.

For your convenience, we structure the chapters based on the sections of Test 000-011: IBM Tivoli Application Dependency and Discovery Manager V7.1 Implementation, such as Planning, Installation, and so on, so studying each chapter will help you prepare for one section of the exam.

The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Raleigh Center.

Vasfi Gucer is a Project Leader at the International Technical Support Organization, Austin Center. He has been with the ITSO since January 1999. He has more than 12 years of experience in the areas of systems management, networking hardware, and software on mainframe and distributed platforms. He has worked on various Tivoli client projects as a Systems Architect in the U.S. He writes extensively and teaches IBM classes worldwide on Tivoli software. Vasfi is also an IBM Certified Senior IT Specialist, PMP and ITIL® Expert.

Christian Fernando Hevia is an IT Specialist within the IBM Global Technology Services group in IBM Argentina. He has specialized in Tivoli products since 2004 and he is currently working with TADDM for Strategic Outsourcing customers in Argentina and Latin America. His areas of expertise include Tivoli Access Manager, Tivoli Configuration Manager, Tivoli Remote Control and Tivoli Directory Server.

David Stephenson is a native of Sydney, Australia, and for the last ten years has worked for IBM Global Technology Services in distributed Systems Management roles. David specializes in Event Management and his depth of recent experiences includes diverse roles encompassing test management, database administration and design, network management, software and license management, financial modelling, and leverage of IBM Tivoli Monitoring Version 6. David has co-authored many IBM Redbooks publications about Systems Management and holds a Masters of Commerce degree with an Advance Specialization in Information Systems and Management from the University of New South Wales.

David's recent IBM Redbooks are *IT Asset Management Processes using Tivoli Asset Management for IT*, SG24-7601 and *Deployment Guide Series: Tivoli IT Asset Management Portfolio*, SG24-7602.

Ghufran Shah is a IBM Certified Advanced Deployment Professional in Enterprise, Provisioning, and Business Application Management Solutions. He has 15 years of experience in Systems Development and Enterprise Systems Management and holds a degree in Computer Science. His areas of expertise include Tivoli Systems Management Architecture, Implementation, and Tivoli Training, together with Business Process Improvement. He has written extensively about event management, monitoring, and business systems management integration and has taught IBM Tivoli courses worldwide. He is currently at TeamSwift Solutions, a trusted advisor for IT Service Management Solutions with a focus on automation, service provisioning, monitoring, and virtualization.

Linda E. Miller-Plumley has been with Tivoli for 11 years, during which she was a Team Lead for the Latin American/U.S. Framework team, and worked on the operation team functioning as a Systems Data Analyst both for a period of 18 months. Since 1998, she has supported Framework and was promoted to Staff Technical Support Engineer in 2000. She has worked with TADDM/CCMDB since 2006, and was the backup to the CCMDB technical lead as a Staff Technical Support Engineer. She also participated in creating portions of the overview and test questions for the TADDM Certification Test. In addition to support, she has taught Framework, network, and operating system tuning worldwide to support and external customers.

Petra Unglaub is a Level 2 Software Engineer in Austin, Texas. She has 10 years of experience in the Tivoli Support field. She holds a degree from Hardin-Simmons University and the University of Bayreuth, Germany. Her areas of expertise include Level 2 defect support for IBM Tivoli Framework and IBM Tivoli Configuration Managery.

Seda Isi is an IT Specialist at the Integrated Technology Services, IBM Turkey. She has been with IBM for two years, working as a member of the client delivery team for service management implementation.

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

Tamikia Barrow
International Technical Support Organization, Raleigh Center

Diane Sherman
International Technical Support Organization, Austin Center

Sara C. Brumfield, Kristin Wall Gibson, Bart Jacob, Emma Jacobs, Mike Mallo, Janet Taylor
IBM USA

Mariella Angelini
IBM Italy

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

Certification overview

This chapter provides an overview of the skill required to become an IBM Certified Advanced Technical Expert. We designed the following sections to provide a comprehensive review of specific topics that provide essential information for obtaining the certification:

- "IBM Professional Certification Program" on page 2
- "Tivoli Software Professional Certification" on page 4
- "Test 000-011: IBM Tivoli Application Dependency and Discovery Manager V7.1 Implementation" on page 7
- "Recommended resources for study" on page 8

1.1 IBM Professional Certification Program

Having the right skills for the job is critical in the growing global marketplace. IBM Professional Certification, designed to validate skill and proficiency in the latest IBM solution and product technology, can help provide that competitive edge. The IBM Professional Certification Program Web site is available at:

http://www.ibm.com/certify/index.shtml

The IBM Professional Certification Program offers a business solution for skilled technical professionals seeking to demonstrate their expertise to the world.

In addition to demonstrating your skill and proficiency in the latest IBM technology and solutions, professional certification can help you excel at your job by giving you and your employer the confidence that your skills have been tested. You can deliver higher levels of service and technical expertise than non-certified employees and move on a faster career track. Professional certification puts your career in your control.

The certification requirements are difficult but not overwhelming. Certification is a rigorous process that differentiates you from everyone else.

The mission of the IBM Professional Certification Program is to:

- Provide a reliable, valid, and fair method of assessing skills and knowledge.
- Provide IBM with a method of building and validating the skills of individuals and organizations.
- Develop a loyal community of highly skilled certified professionals who recommend, sell, service, support, and use IBM products and solutions.

The IBM Professional Certification Program has developed certification role names to guide you in your professional development. The certification role names include IBM Certified Specialist, IBM Certified Solutions/Systems Expert, and IBM Certified Advanced Technical Expert for technical professionals who sell, service, and support IBM solutions. For technical professionals in application development, the certification roles include IBM Certified Developer Associate and IBM Certified Developer. An IBM Certified Instructor certifies the professional instructor.

The IBM Professional Certification Program provides a structured program leading to an internationally recognized qualification. The program is designed for flexibility by enabling you to select your role, prepare for and take tests at your own pace, and, in some cases, select from a choice of elective tests best suited to your abilities and needs. Some roles also offer a shortcut by offering credit for a certification obtained in other industry certification programs.

You can be a network administrator, systems integrator, network integrator, solution architect, solution developer, value-added reseller, technical coordinator, sales representative, or educational trainer. Regardless of your role, you can start charting your course through the IBM Professional Certification Program today.

1.1.1 Benefits of certification

Certification is a tool to help objectively measure the performance of a professional on a given job at a defined skill level. Therefore, it is beneficial for individuals who want to validate their own skills and performance levels, those of their employees, or both. For optimum benefit, the certification tests must reflect the critical tasks required for a job, the skill levels of each task, and the frequency a task must be performed. IBM prides itself in designing comprehensive, documented processes that ensure that IBM certification tests remain relevant to the work environment of potential certification candidates.

In addition to assessing job skills and performance levels, professional certification can also provide the following benefits:

- For employees:
  - Promotes recognition as an IBM certified professional
  - Creates advantages in interviews
  - Assists in salary increases, corporate advancement, or both
  - Increases self-esteem
  - Provides continuing professional benefits
- For employers:
  - Measures the effectiveness of training
  - Reduces course redundancy and unnecessary expenses
  - Provides objective benchmarks for validating skills
  - Facilitates long-range planning
  - Helps to manage professional development
  - Aids as a hiring tool
  - Contributes to competitive advantage
  - Increases productivity
  - Increases morale and loyalty
- For IBM Business Partners and consultants:
  - Provides independent validation of technical skills
  - Creates competitive advantage and business opportunities
  - Enhances prestige of the team
  - Contributes to meeting IBM requirements for various IBM Business Partner programs

Specific benefits can vary by country (or region) and role. In general, after you become certified, you should receive the following benefits:

- Industry recognition

  Certification can accelerate your career potential by validating your professional competency and increasing your ability to provide solid, capable technical support.

- Program credentials

  As a certified professional, you receive (through e-mail) your certificate of completion and the certification mark associated with your role for use in advertisements and business literature. You can also request a hardcopy certificate, which includes a wallet-size certificate.

  IBM Professional Certification acknowledges the individual as a technical professional. The certification mark is for the exclusive use of the certified individual.

- Ongoing technical vitality

  IBM certified professionals are included in mailings from the IBM Professional Certification Program.

1.2 Tivoli Software Professional Certification

The IBM Tivoli Professional Certification program offers certification testing that sets the standard for qualified product consultants, administrators, architects, and partners.

The program also offers an internationally recognized qualification for technical professionals seeking to apply their expertise in today's complex business environment. The program is designed for those who implement, buy, sell, service, and support IBM Tivoli solutions and want to deliver higher levels of service and technical expertise.

Whether you are a Tivoli client, partner, or technical professional wanting to put your career on the fast track, you can start on the road to becoming a Tivoli Certified Professional today.

Benefits of Tivoli certification

Tivoli certification provides the following benefits:

- For the individual:
  - IBM Certified certificate and use of logos on business cards
  - Recognition of your technical skills by your peers and management
  - Enhanced career opportunities
  - Focus for your professional development
- For the IBM Business Partner:
  - Confidence in the skills of your employees
  - Enhanced partnership benefits from the IBM Business Partner program
  - Ability to bill your employees' services at higher rates
  - Strengthens proposals to customers
  - Deepens technical skills available to prospective customers
- For the customer:
  - Confidence in the services professionals handling your implementation
  - Ease of hiring competent employees to manage your Tivoli environment
  - Enhanced return on investment (ROI) through more thorough integration with Tivoli and third-party products
  - Ease of selecting a Tivoli Business Partner that meets your specific needs

Certification checklist

To pursue certification, follow the steps in the checklist:

1. Select the certification that you want to pursue.
2. Determine which test or tests are required by reading the certification role description.
3. Prepare for the test by using the following resources provided:
   - Test objectives
   - Recommended educational resources
   - Sample assessment test
   - Other reference materials
   - List of opportunities for gaining experience

Note: These resources are available from each certification description page, as well as from the test information page.

4. Register to take a test by contacting one of our worldwide testing vendors:
   - Prometric
   - Pearson Virtual University Enterprises (VUE)

   Note: When providing your name and address to the testing vendor, be sure to specify your name exactly as you want it to appear on your certificate.

5. Take the test. Be sure to keep the Examination Score Report provided upon test completion as your record of taking the test.

   Note: After taking a test, your test results and demographic data (including name, address, e-mail, and phone number) are sent from the testing vendor to IBM for processing (allow two to three days for transmittal and processing). After all the tests required for a certification are passed and received by IBM, your certificate is issued.

6. Repeat steps 3 - 5 until all required tests are successfully completed for the certification role. If you must meet additional requirements (such as an *other vendor* certification or exam), follow the instructions on the certification description page to submit these requirements to IBM.
7. After you complete your certification requirements, you are sent an e-mail asking you to accept the terms of the IBM Certification Agreement before receiving the certificate.
8. Upon acceptance of the terms of the IBM Certification Agreement, an e-mail is sent to you containing the following electronic deliverables:
   - A Certification certificate in PDF format, which can be printed in either color or black and white
   - A set of graphic files of the IBM Professional Certification mark associated with the certification achieved
   - Guidelines for the use of the IBM Professional Certification mark
9. To avoid unnecessary delay in receiving your certificate, ensure that your current e-mail is on file by maintaining an up-to-date profile. If you do not have an e-mail address on file, your certificate is sent through postal mail.

   Certificates are sent by e-mail. However, you may also contact IBM at the following e-mail address to request a paper copy of the certificate, including a laminated wallet-sized card:

   mailto:certify@us.ibm.com

Note: IBM reserves the right to change or delete any portion of the program, including the terms and conditions of the IBM Certification Agreement, at any time without notice. Some certification roles offered through the IBM Professional Certification Program require recertification.

1.3 Test 000-011: IBM Tivoli Application Dependency and Discovery Manager V7.1 Implementation

This section describes the IBM Tivoli Application Dependency and Discovery Manager V7.1 Implementation certification test.

1.3.1 Job role description and target audience:

An IBM Certified Deployment Professional - IBM Tivoli Application Dependency and Discovery Manager V7.1 is a technical professional responsible for planning, installation, configuration, operations, administration, and maintenance of a IBM Tivoli Application Dependency and Discovery Manager V7.1 solution. This individual is expected to perform these tasks with limited assistance from peers, product documentation, and support resources.

To attain the IBM Certified Deployment Professional - IBM Tivoli Application Dependency and Discovery Manager V7.1 certification, candidates must pass one test.

1.3.2 About the test

To be certified you must select Test 000-011: IBM Tivoli Application Dependency and Discovery Manager V7.1 Implementation. Note the following information about the test:

- Approximate number of questions: 50
- Duration in minutes: 60
- Format: Multiple choice
- Required passing score: 64%

For the most updated objectives of the IBM Tivoli Application Dependency and Discovery Manager V7.1 Implementation certification test, refer to:

http://www.ibm.com/certify/tests/obj011.shtml

1.3.3 Receive your 15% discount when taking the test

You can receive a 15% discount on the IBM Certified Deployment Professional - Maximo® Asset Management V7.1 certification exam, if taken at any Prometric testing center. Just remember to use the code 15T011.

1.4 Recommended resources for study

Courses and publications are offered to help you prepare for certification tests.

1.4.1 Courses

Refer to the following link for a list of courses related to IBM Tivoli Application Dependency and Discovery Manager V7.1:

http://www.ibm.com/certify/tests/edu011.shtml

The courses are recommended, but not required, before taking a certification test. If you want to purchase Web-based training courses or are unable to locate a Web-based or classroom course at the time and location you desire, contact one of our delivery management teams at:

- Americas:

  mailto:tivamedu@us.ibm.com

- EMEA:

  mailto:tived@uk.ibm.com

- AP:

  mailto:tivtrainingap@au1.ibm.com

Note: Course offerings are continuously being added and updated. If you do not see the courses listed in your location, contact one of the previously listed delivery management teams.

1.4.2 Publications

Before taking Test 000-011: IBM Tivoli Application Dependency and Discovery Manager V7.1 Implementation certification test, we recommend that you review the following product documentation and IBM Redbooks publications.

For online publications of IBM Tivoli Application Dependency and Discovery Manager V7.1, go to:

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/index.jsp?topic=/com.ibm.taddm.doc_7.1/cmdb_welcome.html

IBM Redbooks publications

Refer to the following IBM Redbooks publications as a study resource:

- *Deployment Guide Series: IBM Tivoli Application Dependency Discovery Manager V7.1*, SG24-7616

  This book focuses on deployment and configuration of an IBM Tivoli Application Dependency and Discovery Manager V7.1 environment.

- *IBM Tivoli Application Dependency Discovery Manager Capabilities and Best Practices*, SG24-7519

  This book provides insight into the Tivoli Application Dependency and Discovery Manager (TADDM) capabilities and architecture. It covers procedures for installing and configuring TADDM, tips and techniques for populating the TADDM database and customizing its use, performance considerations, and information about how TADDM integrates with operational management programs.

2

Planning

Proper planning before implementation of IBM Tivoli Application Dependency Discovery Manager (TADDM) in your environment is a very important step. Time spent here will pay off during the implementation and the operation phase. Important considerations for planning of your TADDM implementation are discussed in this chapter.

This chapter contains the following topics:

- "Your environment" on page 12
- "Requirements analysis" on page 22
- "Planning the TADDM topology" on page 25
- "Directory Services integration" on page 27
- "Discovery Library Adapters" on page 28
- "Sensors and discovery" on page 29P
- "Controlling discovery" on page 39
- "Common Data Model" on page 41

2.1 Your environment

Understanding your environment and careful planning are key success factors in a TADDM deployment. You will have to understand the attributes of your environment, its operating systems and application versions, as well as the requirements for the TADDM application. A successful implementation considers the customer's hardware and operating system preferences. This section outlines and explains these requirements and combinations.

2.1.1 Hardware prerequisites

The following list provides the processor, memory, and disk space requirements for a TADDM Server. The requirements are the same whether the machine is an enterprise TADDM Server or a domain TADDM Server.

Each TADDM Server requires a machine with:

- 100 GB of available disk space
- 2 - 4 processors with a minimum process speed of 2 GHz
- 4 - 8 GB of memory
- 4 - 8 GB of swap space on the disk used by the operating system

You must install the database on another machine. Refer to 2.1.3, "Supported database versions" on page 22. For medium to large environments, use more memory.

Note: Server hardware requirements are directly related to the number of configuration items (CIs) that the domain will be supporting.

Configuration items and terminology for sizing

The general rule for TADDM sizing is based on the number of configuration items CIs per host. Understand important terminology when sizing.

Important terminology for sizing your TADDM environment

You should become very familiar with the following Important terms when you size your TADDM environment include:

- Server Equivalent (SE)

  A representative unit of Information Technology (IT) infrastructure is defined as a computer system (with standard configurations; operating system, network interfaces, storage interfaces) installed with server software, such as a database (such as DB2® or Oracle®), a Web server (such as Apache or iPlanet) or an application server (such as WebSphere® or WebLogic). An SE

also accounts for network, storage and other subsystems that provide services to the optimal functioning of the server. Each SE consists of a number of configuration items (CIs).

- Configuration Item (CI)

  As defined by the Information Technology Infrastructure Library (ITIL), a CI is any component that is under the control of configuration management and therefore subject to formal change control. Each CI in the configuration management database (CMDB) has a persistent object and change history associated with it. Examples of a CI are a computer system, an operating system, L2 interface, and database buffer pool size. A Server Equivalent consists of approximately 200 CIs.

Configuration items

Although no common industry agreement exists on the number of CIs per host, a good rule for CIs for each host is about 100 for each processor.

The greater the processing capacity of a computer system, the more likely it is that more business application components, or CIs, are hosted by that computer system, which translates into the following numbers:

- One processor hosts 100 CIs
- Two processors host 200 CIs
- Four processors host 400 CIs

The general rule for the number of CIs within a single TADDM domain is 2 000 000, which allows timely processing of around:

- 40,000 hosts if you have 50 CIs per host
- 20,000 hosts If you have 100 CIs per host
- 10,000 hosts If you have 200 CIs per host

The TADDM Server itself does not scale linearly beyond four processors.

2.1.2 Operating system prerequisites

Table 2-1 summarizes the platforms that TADDM Version 7.1 supports. For each platform, install the latest available service packs.

Table 2-1 Supported operating systems for TADDM V7.1 components

Operating system and supported release	Support details
AIX® 5.2 (release previous to current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server Support is provided for the hardware and the operating system running in 64-bit mode; however, the TADDM Server and the anchor run in a 32-bit Java™ Virtual Machine (JVM™).
AIX 5.3 (current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server Support is provided for the hardware and the operating system running in 64-bit mode; however, the TADDM Server and the anchor run in a 32-bit Java Virtual Machine.
Red Hat® Enterprise Linux® 4.0 x86 (release previous to current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server
Red Hat Enterprise Linux 4.0 x86_64 (release previous to current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server Support is provided for the hardware and the operating system running in 64-bit mode; however, the TADDM Server and the anchor run in a 32-bit Java Virtual Machine.

Operating system and supported release	Support details
Red Hat Enterprise Linux 4.0 for System z® (release previous to current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server Support is provided for the hardware and the operating system running in 64-bit mode only. Both the TADDM Server and the anchor run in a 64-bit Java Virtual Machine on this operating system. Red Hat Update 3 is also required.
Red Hat Enterprise Linux 5.0 x86 (current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server The installation program does not start in GUI mode on Red Hat Enterprise Linux 5.0 systems unless you install the following library file: libXp.so.6. The Red Hat Package Manager (RPM) package libXp-1.0.0-8.i386.rpm must be installed. This package can be found on disk two of the Red Hat Enterprise Linux 5.0 distribution media in the Server directory.
Red Hat Enterprise Linux 5.0 x86_64 (current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server Support is provided for the hardware and the operating system running in 64-bit mode; however, the TADDM Server and the anchor run in a 32-bit Java Virtual Machine. The installation program does not start in GUI mode on Red Hat Enterprise Linux 5.0 systems unless you install the following library file: libXp.so.6. The RPM package libXp-1.0.0-8.i386.rpm must be installed. This package can be found on disk two of the Red Hat Enterprise Linux 5.0 distribution media in the Server directory.

Operating system and supported release	Support details
Red Hat Enterprise Linux 5.0 for System z (current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server Support is provided for the hardware and the operating system running in 64-bit mode only. Both the TADDM Server and the anchor run in a 64-bit Java Virtual Machine on this operating system. The installation program does not start in GUI mode on Red Hat Enterprise Linux 5.0 systems unless you install the following library file: libXp.so.6. The RPM package libXp-1.0.0-8.i386.rpm must be installed. This package can be found on disk two of the Red Hat Enterprise Linux 5.0 distribution media in the Server directory.
Solaris™ 9 SPARC (release previous to current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server Support is provided for the hardware and the operating system running in 64-bit mode; however, TADDM Server and anchor run in a 32-bit Java Virtual Machine.
Solaris 10 SPARC (current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server Support is provided for the hardware and the operating system running in 64-bit mode; however, the TADDM Server and the anchor run in a 32-bit Java Virtual Machine.

Operating system and supported release	Support details
SUSE® Linux Enterprise Server 9.0 x86 (release previous to current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server At the time of release, IBM Software support is investigating intermittent problems with the JVM and SUSE Linux Enterprise Server 9.0 x86. For production environments, use SUSE 10.
SUSE Linux Enterprise Server 9.0 x86_64 (release previous to current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server Support is provided for the hardware and the operating system running in 64-bit mode; however, the TADDM Server and the anchor run in a 32-bit Java Virtual Machine. At the time of release, IBM Software support is investigating intermittent problems with the JVM and SUSE Linux Enterprise Server 9.0 x86_64. For production environments, use SUSE 10.
SUSE Linux Enterprise Server 9.0 for System z (release previous to current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server Support is provided for the hardware and the operating system running in 64-bit mode; however, the TADDM Server and the anchor run in a 32-bit Java Virtual Machine. At the time of release, IBM Software support is investigating intermittent problems with the JVM and SUSE Linux Enterprise Server 9.0 for System z. For production environments, use SUSE 10. SUSE Patch Level 3 is also required.

Operating system and supported release	Support details
SUSE Linux Enterprise Server 10.0 x86 (current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server SUSE Fix Pack 1 is also required.
SUSE Linux Enterprise Server 10.0 x86_64 (current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server Support is provided for the hardware and the operating system running in 64-bit mode; however, the TADDM Server and the anchor run in a 32-bit Java Virtual Machine. SUSE Fix Pack 1 is also required.
SUSE Linux Enterprise Server 10.0 for System z (current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server Support is provided for the hardware and the operating system running in 64-bit mode only. Both the TADDM Server and the anchor run in a 64-bit Java Virtual Machine on this operating system.
Microsoft® Windows® Server 2003 Datacenter Edition, Enterprise Edition, and Standard Edition (current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server ► Windows gateway Windows Server® 2003 R2 is supported. This environment requires Microsoft Service Pack 2.

Operating system and supported release	Support details
Microsoft Windows Server 2003 Datacenter x64 Edition, Enterprise x64 Edition, and Standard x64 Edition (current platform release)	► Domain Manager ► Product Console ► Anchor ► TADDM Server ► Windows gateway Support is provided for the hardware and the operating system running in 64-bit mode; however, the TADDM Server and the anchor run in a 32-bit Java Virtual Machine. Windows Server 2003 R2 is supported. This environment requires Microsoft Service Pack 2.
Microsoft Windows XP Professional (release previous to current platform release)	► Domain Manager ► Product Console
Windows Server 2003 DataCenter (current platform release)	► Domain Manager ► Product Console
Windows Server 2003 Standard Edition (current platform release)	► Domain Manager ► Product Console

Anchor server OS and hardware

To discover components, each TADDM Server must be able to communicate with other computer hosts and network devices. In cases when a firewall prevents direct access from the discovery server to certain hosts or devices, you can specify a computer system that does have access to the hosts or devices to be an anchor host. An *anchor host* acts as a proxy to assist in the discovery process.

You do not have to configure anchor hosts during the TADDM Server installation process, but you do have to include anchor hosts in your installation plan and verify the system requirements for candidate machines.

The TADDM Server, by default, runs a local anchor process. After the TADDM Server installation, you can use the TADDM Product Console to configure which hosts will serve as additional anchor hosts on your network.

Note the following anchor requirements and considerations:

- You can use any operating system on which the TADDM Server can be installed as an anchor.
- Use Secure Shell (SSH) software to communicate to the central TADDM Server. You have to install Bitvise Version 4.06a or later or Cygwin SSH if you plan to use a Windows anchor server.

Supported anchor operating systems include:

- Red Hat Enterprise Linux 5.0 x86_64
- Red Hat Enterprise Linux 5.0 for System z
- Solaris 9 SPARC
- Solaris 10 SPARC
- SUSE Linux Enterprise Server 9.0 x86
- SUSE Linux Enterprise Server 9.0 x86_64
- SUSE Linux Enterprise Server 9.0 for System z
- SUSE Linux Enterprise Server 10.0 x86
- SUSE Linux Enterprise Server 10.0 x86_64
- SUSE Linux Enterprise Server 10.0 for System z
- Microsoft Windows Server 2003 Datacenter Edition, Enterprise Edition, and Standard Edition (R2, Service Pack 2)
- Microsoft Windows Server 2003 Datacenter x64 Edition, Enterprise x64 Edition, and Standard x64 Edition (R2, Service Pack 2)

Network connectivity must exist to the remote servers within the discovery scope in the firewall zone.

Browser support

Table 2-2 lists browsers that are supported for the Product Console and the Domain Manager.

Table 2-2 Product Console and Domain Manager supported browsers

Operating system	Supported browser
Windows Operating Systems	Microsoft Internet Explorer® v.7 and Mozilla Firefox 2.0
Linux, Solaris, AIX, and Linux on System z operating systems	Mozilla Firefox 2.0

Windows gateway and operating system

A Windows gateway provides protocol translation services for the TADDM application for communicating to Windows devices.

This component must run on a Windows operating system. Supported versions of Windows for the gateway component are:

- Microsoft Windows Server 2003 Datacenter Edition, Enterprise Edition, Standard Edition
- Microsoft Windows Server 2003 Datacenter x64 Edition, Enterprise x64 Edition, Standard x64 Edition

Microsoft Windows Server 2003 R2 is supported as an operating system. Microsoft Service Pack 2 is a requirement for both these versions.

The gateway must also have IIS Manager installed to handle network COM/DCOM operations, and BitVise or Cygwin SSH

Network Mapper

The Open Source Nmap (Network Mapper) application may be used to remotely identify the operating system type. Nmap must be installed on the TADDM server and any anchor server.

Supported platforms include:

- Windows
- Linux
- Solaris

When choosing an operating system for hosting Nmap, ensure that you consider the anchor server supported platforms. For example, Nmap runs on Windows 2000, however, only Windows 2003 R2 is a supported version for an anchor server.

Unzip verification for AIX

On supported AIX operating systems, you must have an *unzip utility* available in the `/usr/bin` or `/usr/local/bin` directory for both the AIX TADDM Server and any AIX anchor. If you did not install the unzip utility with the supported AIX operating system, you must put an unzip utility into one of those directories before beginning the installation.

2.1.3 Supported database versions

The following database versions are supported by the TADDM application:

- IBM DB2 Version 9.1 and Fix Pack 2 for AIX (64-bit), Solaris SPARC (64-bit), Linux on System z (64-bit), Linux x86 (32-bit and 64-bit), and Windows (32-bit and 64-bit) operating systems
- IBM DB2 Version 8.2 and Fix Pack 10 for AIX (32-bit and 64-bit), Solaris SPARC (32-bit and 64-bit), Linux on System z (32-bit and 64-bit), Linux x86 (32-bit and 64-bit), and Windows (32-bit and 64-bit) operating systems
- Oracle 9i and 10g

2.2 Requirements analysis

Requirements analysis is critical for a successful TADDM deployment. You have to document the existing environment to make architectural decisions for the TADDM application

2.2.1 Documenting your existing server environment

Documenting your server environment is critical to planning your TADDM installation. An interview with the server administrators is a good practice and can assist you with:

- Verifying the output of your level1 discovery
- Evaluating the requirement for Windows gateways
- Collecting operating system level credentials

2.2.2 Use of lsof utility

Your discussions with the system administrators should include determining whether the **`lsof`** utility is installed on UNIX®-like systems.

To provide complete information about dependencies, TADDM requires the **`lsof`** tool to be installed on all UNIX and Linux host computers with the setuid-root setting so the user running the lsof tool can see processes and ports belonging to other users. On certain platforms, the `setuid` setting is not required. You can download a copy of the lsof tool for your system from independent vendor sources.

Note: For more information about the lsof tool for your AIX 5 platform, go to:

http://www.ibm.com/servers/aix/products/aixos/linux/date.html

Linux distribution CDs, such as Red Hat 4, also typically provide this tool.

2.2.3 Documenting your existing network environment

To document your existing network environment, an interview with your network administrators is a good practice. The objective of the interview should be to:

- Determine and document the types of network equipment.
- Determine and document the number of each type of network equipment.
- Determine and document firewalls.
- Discuss how TADDM works.

Your interview should also cover any credentials required by TADM. These may include Simple Network Management Protocol (SNMP) community strings that are used by TADDM as credentials for network devices.

Firewall considerations

Firewall considerations are a major determining factor in physical placement of TADDM components. Firewalls are used to separate network zones from each other. A zone can contain one or more network-attached devices, such as servers.

Between each zone, only permitted communications are allowed to cross the firewall.

TADDM exploits common, secured communications application protocols, such as SSH, to traverse the firewall. These communications reach a TADDM anchor within the desired zone that in turn has permissions to communicate to the remaining devices in that network zone.

Note: Two basic configuration scenarios for the TADDM application exist. One is to open the firewalls to the traversal of discovery protocols. The second is to limit discovery protocols across the firewall and to permit only secured SSH across the firewall. The second option requires anchor and possibly gateway components.

Decision: open firewalls

If you have no firewalls or a policy of centralizing infrastructure around a data center core, the communications listed in Table 2-3 must be permitted between the TADDM server and the managed device:

Table 2-3 Ping Sensor and Port Sensor Ports

Port name	Port number
CiscoWorks	1741
DNS	53
LDAP	389
SSH	22
WBEM	5988
WMI	135

Decision: anchor placement

If you have a network zone that is separated by a firewall from the other parts of the environment, place an anchor in that zone. The firewall must at least permit SSH between the TADDM domain server and the TADDM anchor.

Decision: gateway placement

If you have Windows servers in the network zone, place a Windows gateway in the zone. This approach permits the use of Windows protocols, such as Server Message Block, for discovery within the zone.

Example of a small firewall environment

As an example, a customer network has a firewalled environment. The TADDM server is located in zone 1. A firewall exists between zone 1 and zone 2. Zone 2 contains 500 UNIX-like systems. Between zones 2 and 3 is a firewall. Zone 3 has 100 Windows 2000 systems and 300 Windows 2003 systems.

We assume that the only protocol allowed across the firewalls is SSH.

Our placement decisions are as follows:

- Small enough server numbers for a single domain (a domain server and its database server)
- Use of the TADDM server itself as the anchor server in zone 1
- Placement of UNIX-like anchor server in zone 2 to manage the UNIX servers
- Placement of a Windows server to act as an anchor and gateway in zone 3 to manage the Windows servers

2.2.4 Documenting your existing application environment

For each business application to be managed with TADDM, determine the various servers, components, and applications that are in the business application.

Discuss how TADDM works, what the credential requirements or components and applications are, and begin the process to manage and obtain credentials.

If system and network diagrams, or application architecture documents and diagrams are available, review the available documents and determine the credentials required for discovery of the application environment. Credentials should be created as soon as possible

Infrastructure components

The infrastructure services components that can be discovered with TADDM are:

- Network File System (NFS)
- Domain Name System (DNS)
- Lightweight Directory Access Protocol (LDAP)

2.3 Planning the TADDM topology

The following points are important when planning the TADDM topology in a customer's environment:

- Determine the best estimate of the number of TADDM V7.1 components.
- Determine which authentication and security protocols are in use.
- Determine whether machines exist behind firewalls to determine the need for anchor servers and Windows gateways.
- Determine the number of assets to be managed.
- Estimate the number of configuration items to be created.
- Estimate the number of computer systems to be discovered.
- Determine the number of domains and enterprise servers that are required for each environment monitored by TADDM V7.1.

2.3.1 Federating with eCMDB

Federating of data is possible by use of multiple domain managers and a single enterprise-level domain manager or eCMDB.

In Figure 2-1, a single Enterprise Configuration Management Database (eCMDB) is used to span various business units. Each business unit is managed as a separate entity by its own domain manager, with the single eCMDB being used to federate the data into a single view across the enterprise.

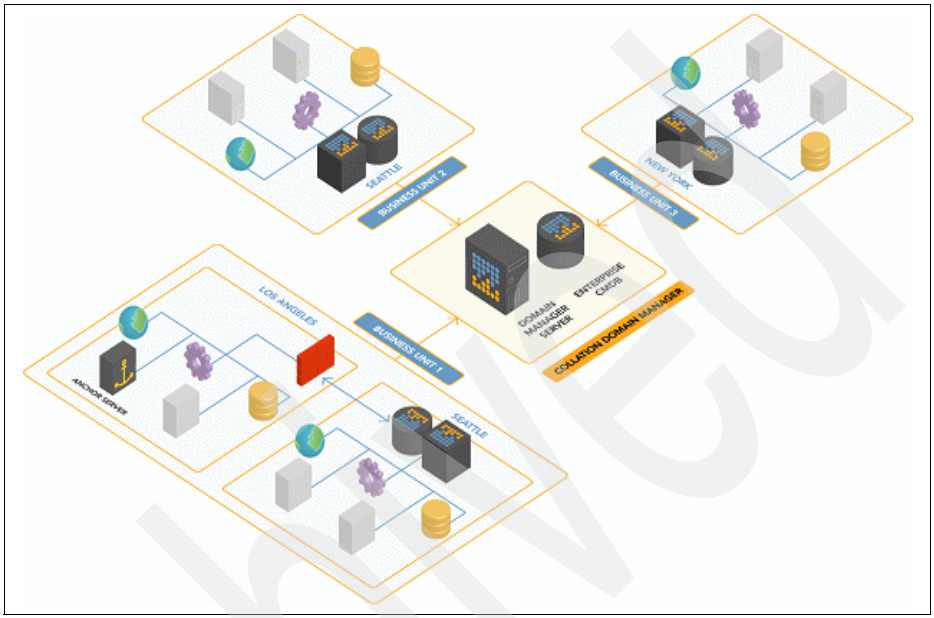

Figure 2-1 eCMDB architecture

The eCMDB may also be deployed where the number of managed servers exceeds the processing windows available to a single domain server, or where network locations indicate large concentrations of servers in multiple data centres.

Example of federated across data centers

As an example, a customer has four data centers with 2000, 2500, 3000, and 3000 servers.

The minimum architectural requirement for the TADDM installation would be a single domain server with an anchor in each data center.

A growth strategy, or organizational boundaries, may also derive an alternate architecture of a domain server in each data center, with a single eCMDB.

2.3.2 Maximum number of configuration items

The rule-of-thumb for the recommended number of configuration items (CIs) within a single TADDM domain is 500,000.

This value equates to the processing of:

- 10,000 hosts if you have 50 CIs per host
- 5,000 hosts if you have 100 CIs per host
- 2,500 hosts if you have 200 CIs per host

Note: For more details about sizing of TADDM, refer to the Deployment Planning Case Study in *IBM Tivoli Application Dependency Discovery Manager Capabilities and Best Practices*, SG24-7519.

2.4 Directory Services integration

TADDM provides support for external user authentication data stores, such as LDAP. The TADDM administrators are always authenticated locally. If LDAP authentication is enabled, TADDM bypasses internal authentication for all users.

TADDM supports both anonymous and password-based authentication with an external LDAP server. To configure the LDAP data store, you have to provide the LDAP server host name, port, base DN and bind DN, and password (required for password-based authentication). TADDM also provides the ability to search for specific naming attributes to match the UID.

Two user registries are supported. One is LDAP (either LDAP V2 or V3) and the other is IBM WebSphere Application Server Federated Repositories.

Note: TADDM V7.1 support has been tested with IBM Tivoli Directory Server Version 6 Release 0.

2.4.1 User registry

Several security-related properties (options) are in the following file:

```
$COLLATION_HOME/etc/collations.properties
```

These options include:

- file-based user registry
- LDAP user registry
- WebSphere federated repositories user registry

2.4.2 LDAP configuration

To configure TADDM to use LDAP for the user registry, you should have the following information:

- LDAP host name
- LDAP port number
- User ID to use for the LDAP server
- Password for the LDAP user id
- LDAP Base Distinguished Name. This is case sensitive.
- Optional use of the LDAP anonymous binding function. In this case, the user name and password fields are not required.

2.5 Discovery Library Adapters

A Discovery Library Adapter (DLA) is a software program that extracts data from a source application, such as IBM Tivoli Monitoring, IBM Tivoli Business Services Manager, IBM Tivoli Composite Application Management (ITCAM), and so on. You must create a DLA store in order to use the bulk load program.

For a Tivoli collection of books that can load the TADDM Database with data from other Tivoli products, go to the IBM Tivoli Open Process Automation Library (OPAL) Web site at:

http://catalog.lotus.com/wps/portal/tccmd

DLAs are specific to a particular product, because each product has a distinct method of accessing the resources from the environment. The configuration and installation of a DLA is different for every application. A typical DLA is installed on a system that has access to the data of a particular application. For example, the DLA for IBM Tivoli Monitoring is installed on a computer that has access to the IBM Tivoli Monitoring enterprise management system database. All DLAs are run using the command-line interface and can be scheduled to run using any type of scheduling program in your environment (for example, cron).

You can create a DLA to extract information from existing products or databases in your environment. For more information about creating a Discovery Library Adapter, refer to the *Tivoli Application Dependency Discovery Manager Discovery Library Adapter Developer's Guide* at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/cmdb_dladevguide.pdf

2.6 Sensors and discovery

Given that TADDM defines different levels of discovery, you should understand the discovery requirements of the environment and then explain to the customers what options are available, what are the capabilities of these different levels of discovery mechanisms, and how customer requirements can be matched by creating different discovery processes. This information is critical for you to understand.

The TADDM agent-free discovery engine manages the overall discovery process. The discovery process collects the data that is required to instantiate the IBM Model (Common Data Model) to represent the specific data center infrastructure.

2.6.1 Discovery components

The TADDM discovery system consists of the components described in this section. Figure 2-2 on page 29 shows the discovery components.

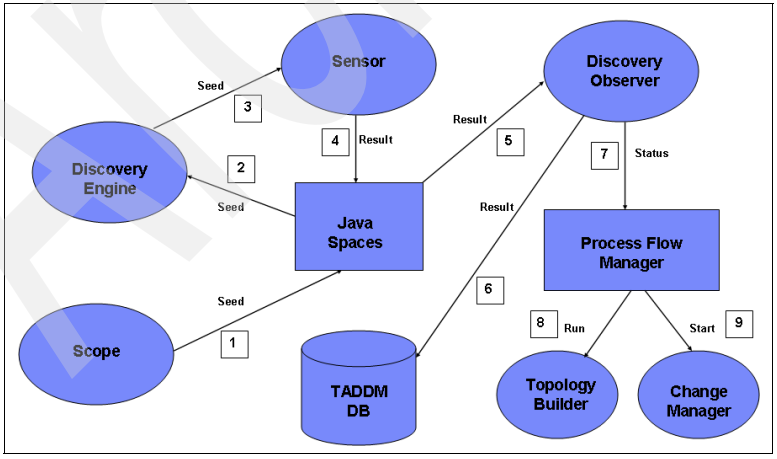

Figure 2-2 Discovery components

Discovery Engine

The Discovery Engine service is responsible for discovering the contents of the data center, which it does by running discovery sensors. A discovery workflow decides which sensor to un against what target.

Each discover sensor has an input object called a seed, then a discover step, and an output called a result.

A discovery is started by providing a set of initial seeds (typically an IP address or a range of IP addresses to discover). This seed provides a starting-point for the Discovery Engine, which triggers the initial sensors. The results from these sensors are then converted to new seeds, which trigger a new set of sensors and so on, until no more result-to-seed conversions can be made.

Sensor

The sensor is the primary agent for discovery in TADDM. The sensor is responsible for discovering deep-dive information about the remote system characteristics.

The data that the sensor discovers is mapped into model objects that then get saved to the TADDM database. For some sensors, discovered results also cause new seeds to be created, spawning new sensors in turn.

Java Space

Java Space is used as a synchronization space for the implementation of the discovery logic. During a discovery run, the Discovery Engine stores the seeds and results objects in the JavaSpace, and includes thread operations, such as:

- Selecting the seeds according to the discovery task that represents the target system.
- Discovering the target system and returning the results
- Creating new discovery tasks from the seeds

Discovery Observer

The Discovery Observer service persists the results that the discover sensors discover by communicating with the Topology Manager. It also determines the status of a discovery by observing the contents of the JavaSpace.

Process Flow Manager

The Process Flow Manager service controls the discover state, and it also takes care of running needed post-discover steps, such as the Topology Builder.

Topology Builder

The Topology Builder builds the relations and dependency between the discovered items.

2.6.2 Level 1 discovery

The Level 1 profile is designed to perform credential-free discoveries. The default purpose is to discover and create Computer Systems inventory from a defined set of host, range, or subnet of IP addresses scope.

The main sensor that runs during a Level 1 discovery is the StackScanSensor, which gathers basic information from the discovered computer system.

When TADDM is used with Nmap, much more detailed data is returned and a higher degree of confidence in operating system identification can be achieved.

For UNIX, the StackScanSensor requires a user ID in the sudoers file, as shown in Example 2-1.

Example 2-1 /etc/sudoers

```
# sudoers file.
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the sudoers man page for the details on how to write a sudoers
file.
#
# Host alias specification
# User alias specification
# Cmnd alias specification
# Defaults specification
# User privilege specification
taddmuser ALL=(ALL) NOPASSWD:ALL
#taddmuser  ALL=(root) NOPASSWD:/usr/sbin/lsof

# Uncomment to allow people in group wheel to run all commands
# %wheel        ALL=(ALL)       ALL
# Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL
# Samples
# %users  ALL=/sbin/mount /cdrom,/sbin/umount /cdrom
# %users  localhost=/sbin/shutdown -h now
```

The example shows two entries for the taddm user. The first entry has the permissions to execute all commands as root:

```
taddmuser ALL=(ALL) NOPASSWD:ALL
```

The second entry has been commented out with a number sign (#) at the start and limits the root permissions to running the **lsof** command:

```
#taddmuser  ALL=(root) NOPASSWD:/usr/sbin/lsof
```

For Windows operating systems, sudo access control is not required.

Note: Nmap must be installed separately on each TADDM and anchor server.

2.6.3 Level 2 discovery

The Level 2 discovery is only used to discover hosts with credentials. The Level 2 profile includes the Level 1 profile with the operating system credentials entered into the access list for all the targeted machines. Additional computer system sensors may be specified. This is still more of an inventory discovery with a minimum of discovered dependencies (TCP-only connections).

Discovering target computer systems

If you want TADDM to discover the target computer systems in your environment, they must be configured with the minimum requirements for discovery. The minimum requirements that apply to the target computer systems that you want TADDM to discover with Level 2 and Level 3 discovery are:

- Secure Shell (SSH)

  You can use either OpenSSH, or the vendor-supplied version of SSH that comes with the operating system.

- LiSt Open Files (lsof)

  To provide complete information about dependencies, install the LiSt Open Files (lsof) program on all target computer systems according to the requirements. Because the lsof program depends on the version of the operating system for which it was compiled, ensure that you install the correct lsof program for your version. For example, if you get the following type of message, the correct lsof program is *not* installed:

```
sushpatel79: $ lsof -nP -i | awk '{print $2, $9, $10}' | sort -k 2 | uniq
-f 1
lsof: WARNING: compiled for AIX version 4.3.2.0; this is 5.1.0.0.
10352 *
24770 * (CLOSED)
12904 * (LISTEN)
```

Discovering Linux

To discover the Linux systems, the following items are required:

- The lsof program must be run without a password challenge
- An account with a non-root privilege may be used, however, some commands that TADDM uses might require privilege escalation.
- User access information such as user name and password or key is necessary.

Discovering Windows

Discovery of Windows systems can be performed by two methods:

- WMI-based using user account access through a Windows gateway.
- SNMP-based using SNMP community string access.

WMI-based discovery

To use WMI-based discovery, perform the following steps

1. Install and configure WinSSHD on the TADDM Windows gateway server.
2. Disable host lock-out feature on WinSSHD.

Note that each Windows target requires a user account in the local admin group of the Windows gateway server with WMI access to all WMI objects on that machine. This account can be a local account or a domain account.

SNMP-based discovery

TADDM supports SNMP-based discovery of Windows systems in addition to WMI-based discovery. To use SNMP to discover Windows targets, enable SNMP, and ensure access to SNMP MIB2 GET Community String with permission for MIB2 system, IP, interfaces, extended interfaces, and host resources. If WMI is not available, SSH-based discovery is possible.

Level 2 discovery with shallow application discovery

Level 2 discovery with shallow application discovery is an enhanced scan that can also capture shallow application objects without using application level credentials. With this variable set to true, a CustomAppServer object representing the application running on the target machine will be received.This level of discovery can also build limited application dependencies as they are discovered.

To enable Level 2 discovery with shallow application discovery, set the following variable to true in the `collation.properties` file:

```
com.collation.internalTemplatesEnabled
```

2.6.4 Level 3 discovery

Level 3 discovery is used for deep discovery. This level also includes all of Level 2. The Level 3 discovery can be used to discover the entire application infrastructure, deployed software components, physical servers, network devices, virtual LAN, and host data used in the runtime environment.

The Level 3 discovery requires credentials for all deep discovery sensors that are delivered with the TADDM product. This level builds dependencies of all applications as they are discovered and have sensors for. All requirements for Level 2 discovery also apply for Level 3.

Discovering target computer systems

Refer to “Discovering target computer systems” on page 34 in the description of 2.6.3, “Level 2 discovery” on page 32.

2.6.5 Sensor flow

Sensors for Level 2 and Level 3 commence with the ping sensor and the IPRange Sensor.

Figure 2-3 on page 35 shows the basic discovery sensor sequence.

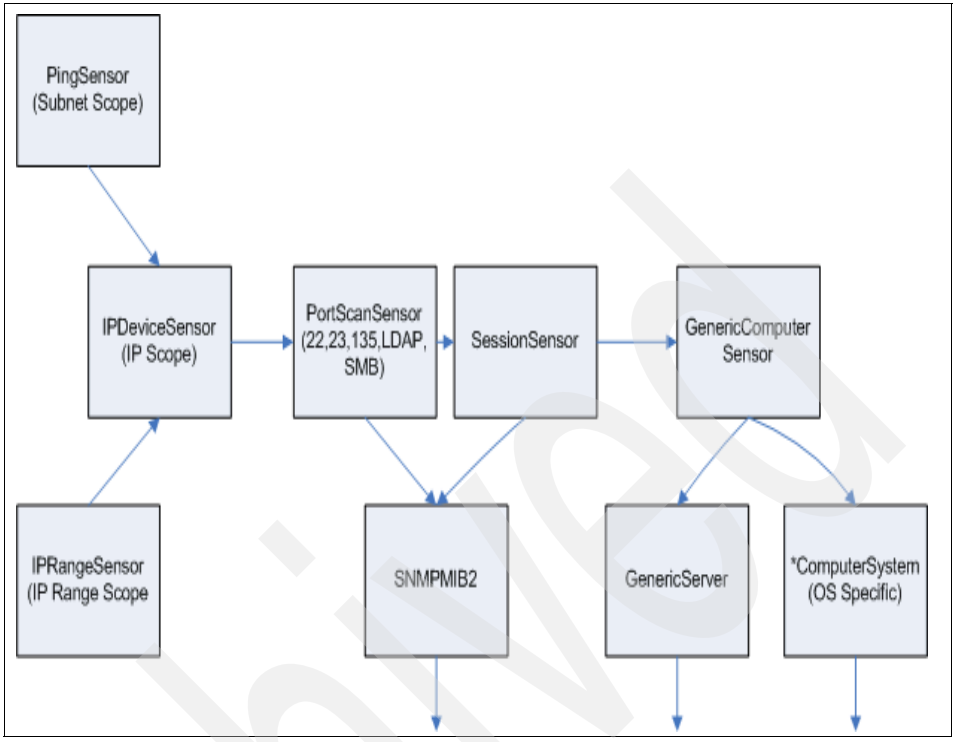

Figure 2-3 Basic discovery sensor sequence

The basic discovery flow actions are:

1. TADDM specifies an initial scope (seed) for a discovery run, either through the GUI or an API call. This initial scope becomes the first seed for the discovery run and is written to the Java Spaces.
2. TADDM identifies the active IP devices in the chosen scope:
   - TADDM attempts a Transmission Control Protocol (TCP) connection on several ports (such as 22 and 135) looking for a response.
   - Any response is enough to notify TADDM that the device exists.
   - An IP device is created, and a PortScan seed is created.
3. TADDM determines if a method of establishing a session to the IP device exists:
   - The PortSensor tries a TCP connection on several ports (including 22 and 135) to try to establish what technology TADDM uses to discover the host.
   - TADDM creates either a SessionSensor seed or an SnmpMib2Sensor seed.

4. The session sensor activities are as follows:
   a. If the SSHPort was open, the session sensor tries to establish a Secure Shell (SSH) connection using credentials from the Access List.
   b. The session sensor tries to use Access List entries of type computer system or Windows computer system, in sequence, from the Access List until an entry works or until the list is exhausted.
   c. If the Windows Management Interface (WMI) port was open, the session sensor establishes an SSH connection with a gateway computer system (provided that a gateway computer system is found for the target).
   d. If the session sensor cannot establish a session, an *SnmpMib2Sensor seed* is created.
   e. If a session is established, a Generic Computer System Sensor seed is created.
5. The Generic Computer System Sensor performs the following tasks:
   – Tries to determine what type of OS is installed, such as AIX, Linux, SunOS™, Hewlett-Packard UNIX (HP-UX), Windows, Tru-64, OpenVMS, and so on.
   – Creates a seed that is appropriate for the OS and discovers other system components, such as sharing, storage, and so on, as shown in Figure 2-4 on page 37.
6. An OS-specific sensor is invoked and uses native commands, such as **`ps`**, **`netstat`**, and **`lsof`**, to find a list of all of the running processes that are listening on a socket.
7. The sensor proceeds to perform its application (ISS, Apache, and so on) discovery, and the discovered results are written back to the Java Space.
8. The Discovery Observer detects that a result was placed in the Java Spaces and extracts the result.

Figure 2-4 illustrates the operating system and application discovery process.

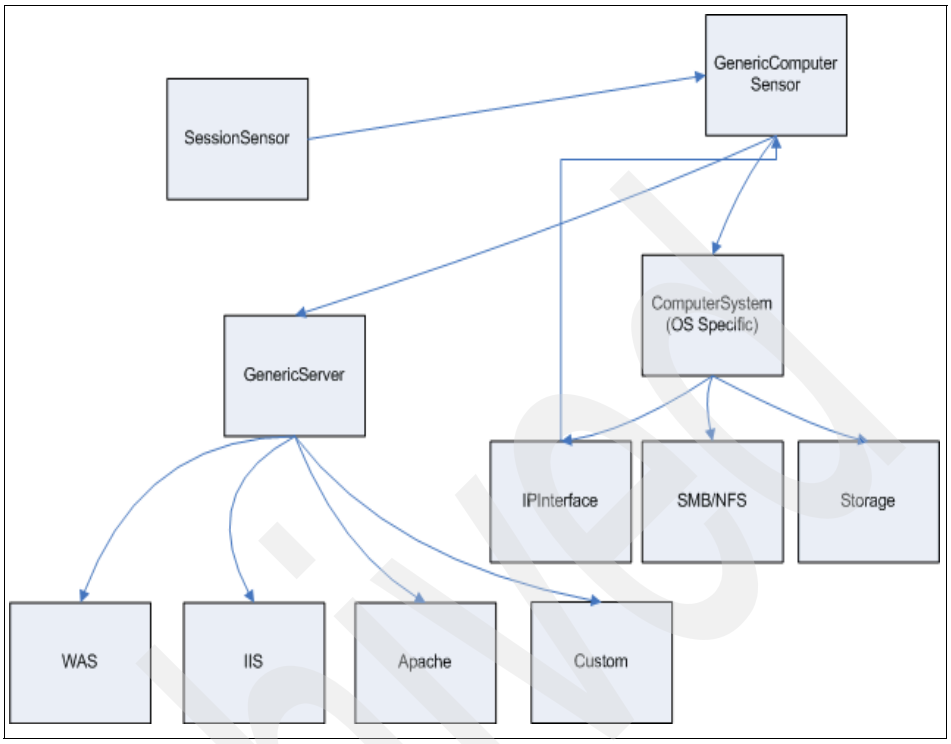

Figure 2-4 Application Discovery

Note: Specific Level 3 sensors are executed for discovering applications such as WebSphere, IIS and Apache. Note that the IPInterface sensor feeds back any newly discovered interfaces to the GenericComputer Sensor for further discovery.

9. The Discovery Observer saves the result to the database. If there is a result converter associated with the sensor, the Discovery Observer passes the result to that result converter.
10. The result converter parses the result and might generate new seeds, which are put back in the Java Spaces.

Discovering WebSphere

The WebSphere sensor now has three discovery levels, which are shallow, medium, and deep. The default level is shallow.

Your discovery scope must include the server hosting WebSphere.

If WebSphere security is disabled, user accounts are not required. If security is enabled, you must have the following items configured:

- WebSphere Administrator user ID and password
- Client-side SSL certificate, including two files, their trust and key stores (including their passphrases)

Certificate setup

When you set up the access list, including the user name, password, and scope limitations, you must click **SSL Settings** to set up the certificates. The certificates have to be on the system running the console, which is not necessarily the server. These certificates can be retrieved from the WebSphere Application Server. If your WebSphere Administrator has not changed the default certificates shipped with WebSphere, you can find the certificates in the following directories:

- For WebSphere Application Server 5.1:

  `<WebSphere_Root_Directory>/etc`

- For WebSphere Application Server 6.x

  `<WebSphere_Root_Directory>/profiles/<profile_name>/etc`

WebSphere ships with two *dummy* certificate files:

- `DummyClientTrustFile.jks`
- `DummyClientKeyFile.jks`

The default passphrase is WebAS.

If your WebSphere Administrator changed the files or passphrase, use the updated files and passphrase when setting up the certificate. If you cannot find the files in the default directory, search for client certificate files with the `.jks` extension.

Discovering MySQL database

MySQL™ database servers can be discovered by extending the attributes collected by a custom server template. The extended attributes must first be defined through the Product Console or the API prior to using custom server templates to set the values of the attributes.

See how to set values of extended attributes during a custom discovery, with a custom server template at:

`http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1/SDKDevGuide/t_cmdbsdk_attributesextendedcustomservext.html`

Discovering OpenVMS

OpenVMS V7.x systems are discovered with the sensor named OpenVmsComputerSystemSensor.

The sensor creates three model objects:

- core.LogicalContent
- sys.openvms.OpenVms
- sys.openvms.OpenVmsUnitaryComputerSystem

Other discovery profiles

Any of the Level 1 - 3 profiles can be cloned and modified for your requirements. New profiles can be created and parameters modified or removed by using the Product Console.

2.7 Controlling discovery

When you create a new profile, you can use an existing profile as a basis. From the Clone existing profile menu, select an existing profile or select **None**.

With the Level 2 profile, you have the option to enable a shallow discovery of applications running on a target system using only the system credentials. To do so, you have to add the following property to the `collation.properties` file:

```
com.collation.internalTemplatesEnabled=true
```

If this property is set to true, you receive a CustomAppServer object, which represents the application running on the target machine. You do not have to provide application credentials to enable this property.

Manually merging discovered configuration items

Manual merging is the process where you decide to combine two or more configuration item (CI) objects displayed in the Product Console into one CI. The fact that the CIs are displayed separately indicates they do not have overlapping naming rules, and as far as TADDM is concerned, the CIs are different. If you are certain that the two CIs represent the same real life CI, you can select the CIs in the Product Console and direct the system to combine them into one CI.

Transient or durable CI

When merging CIs, a single CI is selected from the list of CIs to be merged. This CI is called the durable CI and is retained at the end of the merge operation. The

other CIs are called transient CIs and are deleted at the end of the merge operation

The following rules apply to manually merging CIs:

- When CIs are merged, only the attributes of primitive types (string, integer, and so on) are transferred from the transient CI to the durable CI, and only if the durable CI does not already have a value for that attribute.
- Arrays and objects associated with the transient CI are not transferred.
- When a transient CI is deleted, all of its related CIs are deleted also. For example, if a computer system CI is deleted, then the operating system CI running on the computer system is also deleted, and so are all the software installations on the operating system.
- Merging is not currently supported for business systems or business applications.
- If a CI that is designated as a transient object in a manual or automated merge is later rediscovered or reloaded through the bulk load facility, it updates (combines with) the durable object that it was originally merged with and does not result in a second instance of the CI.

  However, objects contained by the transient object (called child objects) are treated differently. Because only a shallow merge is performed, which combines only the top level objects, the child CIs of the transient might still not be recognized as identical objects. The result is potentially multiple instances of a child CI. If multiple child CI instances do result after a merge, or on subsequent loads of data of merged objects, the extra copies might be deleted.

  For example, if a bulk load operation results in computer systems CS1, CS2 and operating system OS1 being stored with OS1 installed and running on CS1. If CS1 and CS2 are then merged with CS1 as the durable CI, then only CS1 and OS1 will remain but TADDM does recognize that CS2 is the same CI as CS1. If another bulk load operation results in CS2 and OS2 being added with OS2 installed and running on CS2 then the existing CS1 will be updated by the information in CS2 but OS1 and OS2 may remain as separate CIs. In this case, the proper action to take is to delete both operating systems and when the operating system is rediscovered, the next time it is created so that duplicates do not exist in the future.

  In very rare circumstances where the durable and the transient CI share the identical child object (not just having different representations of the same child object), the child object might be deleted as a result of the merge. Rediscovering or reloading the durable object (with its children) restores the child object. If two CIs are mistakenly merged together attempting to rediscover or reload, the transient object results in updating the durable object and does not recreate the original transient CI.

To rectify this situation, the durable CI should be deleted and the durable and transient CIs rediscovered or reloaded. At this point, they again are treated as separate CIs.

2.8 Common Data Model

The Common Data Model (CDM) represents the foundation of TADDM and provides the definition for the data center applications and their supporting infrastructure components, cross-tier relationships, and configuration attributes. Without a reference model, implementation is dependent on expensive, time-consuming, incomplete and error-prone manual modeling. TADDM provides definitions for a wide variety of commonly deployed software applications, hosts, network devices, and network services. The extensible reference model includes an event propagation model that provides the underpinnings to interpret infrastructure component events in the context of the applications that they deliver.

The aim of the CDM is to:

- Facilitate sharing of data among multiple applications.
- Provide a common means of describing objects and relationships in IT environments.
- Provide a common format (IDML) for sharing the data through files.

The CDM is based on the Distributed Management Task Force (DMTF) Common Information Model (CIM) object model. You can find more information at:

http://www.dmtf.org

For platform-specific extensions, such as JSR773, go to:

http://www.jcp.org/en/jsr/detail?id=77

The CDM includes a variety of object types, including:

- Software components (Web, application, and database servers)
- Hosts and operating systems
- Network elements (routers, switches, load balancers, firewalls, and storage)
- Network services, such as Lightweight Directory Access Protocol (LDAP), Network File System (NFS), and Domain Name System (DNS)

The CDM is easily extensible based on client-specific requirements.

The model representation for each component type includes:

- Signature

  The signature uniquely identifies the component type and its dependencies and configuration template.

- Configurations

  Configuration data elements include:

  - The static and the dynamic configurations of the component.
  - The runtime resources that the component uses (for example, the Java Database Connectivity (JDBC™) connection pools or the Java Message Service (JMS) topic queue that an application server uses, the patches deployed on an OS, or the IP routing table of a network element)
  - The deployed application objects (for example, the Enterprise JavaBeans™ (EJBs) and JavaServer™ Pages (JSPs) on an application server that implement the business application and services

- Dependencies

  Dependencies model the runtime relationships among the various components within the data center. TADDM discovers and categorizes several types of cross-tier dependencies, including:

  - Transactional dependencies

    Transactional dependencies are the logical connections (IP-based) between the components of a distributed application. These connections represent the provider-consumer relationships between the components, for example, an application server is the consumer of a service provided by a database server.

  - Containment dependencies

    Containment dependencies are the cross-tier hierarchical relationships (for example, an application server is deployed on a host), and logical grouping relationships (such as a Web, application, and database server that make up a business application).

  - Service dependencies

    Service dependencies are the network services upon which most infrastructure components depend (NFS, DNS, and LDAP services).

Figure 2-5 on page 43 is an example of the Common Data Model specification. This example illustrates the *ComputerSystem sub-model* that defines the key object for Tivoli Management Products. In the Common Data Model, a computer system is a combination of hardware (the machine) and software (the operating system) and is flexible enough to allow the representation of various combinations of hardware and software. In the ComputerSystem sub-model,

both the computer system and the operating system are considered combinations of interesting entities. As a result, both the Common Data Model specification and the ComputerSystem sub-model are derived from the superclass System. There are also linkages from this model to the Process sub-model and the FileSystem sub-model. Operating system processes are considered specific parts of an overall business process. One important concept that is included in this model is the representation of the OperatingSystem sub-model as a place where software can run a hosting environment or an interface. This concept is also applied later to other hosting environments, such as WebSphere.

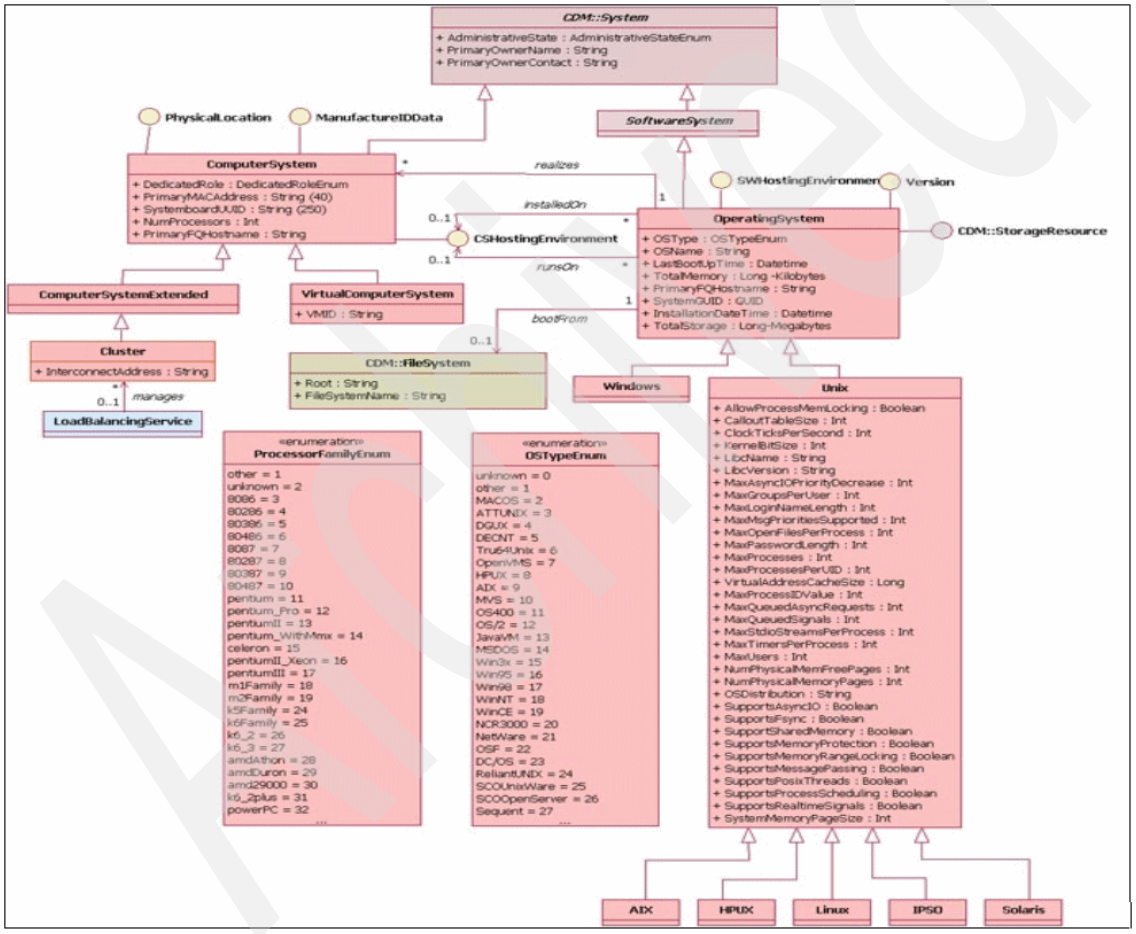

Figure 2-5 TADDM data model

Data model extensibility

The IBM Common Data Model can be easily extended in the field. Through the Java Control Console or the Domain Manager user interface, you can easily add extended attributes to existing object classes. Also, you can either use the GUI or the API to associate values to the extended attributes, and you can view the populated results through the GUI.

Figure 2-6 illustrates a computer system with extended attributes.

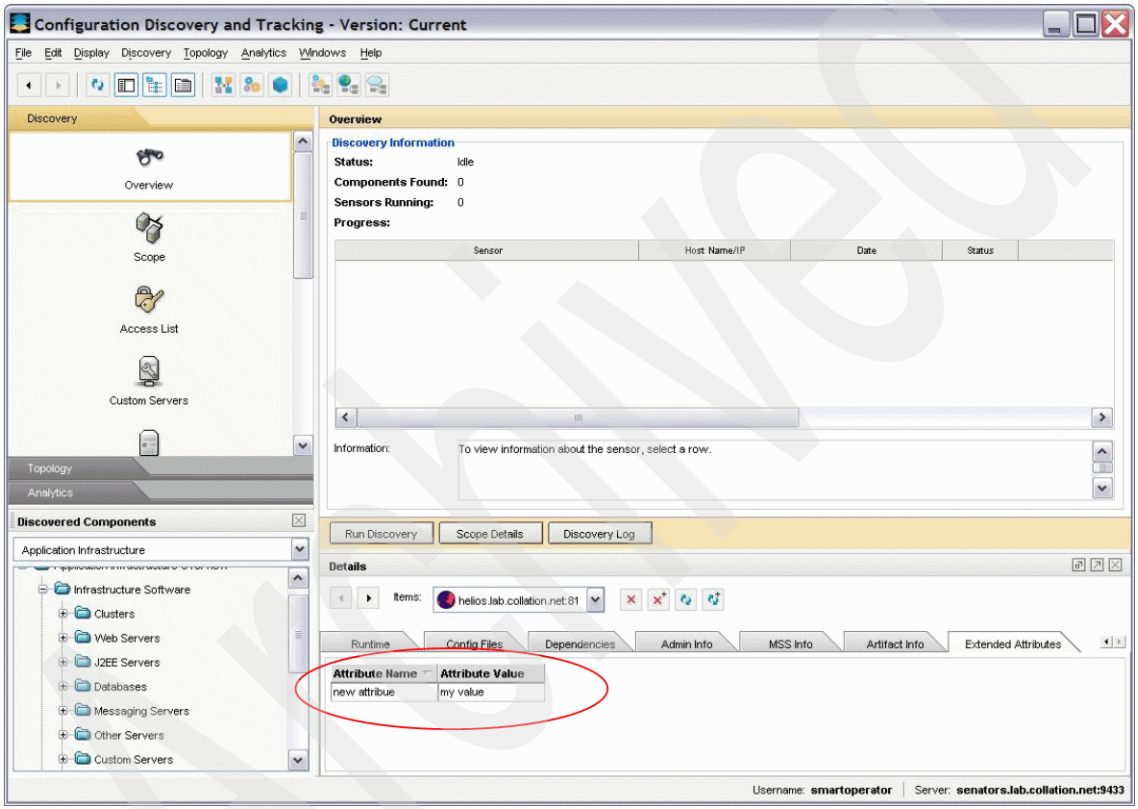

Figure 2-6 Adding extended attributes in TADDM

CDM classes

Of the various objects in the CDM, classes are the only objects in use to represent resource instances. Particular classes are mentioned throughout the TADDM documentation that have particular meaning. These classes are discussed in this section.

ModelObject

This class represents the base or root class in the CDM. All classes derive in some way from ModelObject. The term ModelObject is used in the documentation to represent any defined class in the CDM. The ModelObject and ManagedElement classes are used interchangeably.

ManagedElement

This class is another representation of a base or root class in the CDM, and directly corresponds to the DMTF Common Information Model representation with the same name. The term ManagedElement is also used in the documentation to represent any defined class in the CDM. The ModelObject and ManagedElement classes are used interchangeably.

ManagementSoftwareSystem

Also known as MSS, this class represents the management products that are providing data to TADDM through some mechanism. Each provider of data (including TADDM's sensors) are represented as a resource instance of the type ManagementSoftwareSystem.

3

Installation

This chapter provides installation steps for advanced installation, middleware installation, and considerations for anchor and gateway deployment.

This chapter contains the following topics:

- "Installation overview" on page 48
- "Prerequisite tasks" on page 49
- "Simple installation" on page 50
- "Advanced installation" on page 55
- "Silent installation" on page 62
- "Anchor and gateway installation" on page 64

3.1 Installation overview

IBM Tivoli Application Dependency and Discovery Manager (TADDM) V7.1 offers two types of installation: simple and advanced.

- Simple installation can be with and without the installation of a DB2 database.

  The simple installation process completes the following tasks:

  - If the DB2 database is not installed, gives you the option to install the DB2 database on the local system.
  - Creates the TADDM database on the local DB2 database.
  - If the secondary DB2 system user ID, the archive user ID, does not exist, creates this user ID.
  - Installs the Domain Server.
  - Installs the server by using default values where possible.
  - Uses the default file-based user registry for security.

  Note: When TADDM is installed with the default user registry, the TADDM user credentials are stored in the TADDM installation directory.

- Advanced installation can be with a remote DB2 or Oracle database. The remote DB2 or Oracle database should be installed prior to the installation of the TADDM server.

  Note: Use the advanced installation to customize the server for the production environment.

  The advanced installation process completes the following tasks:

  - Allows you to choose the type of server that you install, either the Enterprise Domain Server or the Domain Server.
  - Allows you to change attributes, for example, port numbers.
  - Allows you to configure TADDM server startup options.
  - Gives you the option to install the WebSphere Federation Server.
  - Allows you to choose the type of user registry to use for security. Options include file-based user registry, LDAP user registry, and WebSphere federated repositories user registry.

3.2 Prerequisite tasks

Before installing the TADDM server, a database server should be configured on a remote system. To prepare the database server on a remote system, you have to install either DB2 or Oracle database. After installing the database, you have to create the primary and secondary user IDs.

3.2.1 Using a local DB2 database

The TADDM server requires a database. This database can be either local or a remote database. Because local databases are not supported in a production environment they should only be used in a test or development environment.

The TADDM server requires two DB2 system user IDs:

- Primary user ID
- Secondary user ID

The *primary user ID* is the DB2 instance owner. This user ID is created during the DB2 installation process.

If the *secondary user ID, the archive user*, does not exist, this user ID is also created during the TADDM installation process. If the secondary user ID already exists, ensure it is in the same DB2 group as the DB2 instance owner.

Being in the same DB2 group as the DB2 instance owner gives the archive user the permission to access the database.

In addition to the user IDs, the TADDM server also requires a database to store discovery data. You can either manually create the TADDM database or let the installation process create the TADDM database.

Note: Installing the TADDM server and the DB2 server on the same system is not feasible for a production environment. For a production environment, install the TADDM server and the DB2 server on separate systems.

3.2.2 Using a remote DB2 database

A remote database is the supported option in a production environment. When you use a remote DB2 database, manual creation of the database is necessary.

Complete the following steps on the system where the DB2 server is installed:

1. Manually create the TADDM database.
2. Create a primary and a secondary DB2 system user ID (archive user).
3. Add the archive user to the DB2 group of the DB2 instance owner. By being in the same DB2 group with the DB2 instance owner, the archive user has permission to access the database.

Notes:

- For the manual creation of the TADDM database, the script `make_db2_db` can be used to create a DB2 database. Furthermore, the primary and the archive user are created with the script. For an Oracle database, the script `make_ora_db` can be used. These scripts can be found on the TADDM server in the `<COLLATION_HOME>/support/bin` folder.
- When using a remote database server in the TADDM architecture, the TADDM server uses JDBC to communicate with the remote database.

No further database configuration task has to be performed before starting the TADDM for the first time.

3.3 Simple installation

This procedure is intended for resources that meet the supported hardware and software criteria. Only DB2 database is supported for simple installation. To use this procedure, you must have one of the following IDs:

- For Windows operating systems, use a Windows logon ID with administrator authority.
- For Linux, Solaris, AIX, and Linux for System z operating systems, use the root user ID.

If you are installing on a dual-stack system that supports both the IPv4 and IPv6 protocols, make sure any numeric IP addresses that you specify during the installation process are IPv4 addresses.

This installation can be a lengthy process, depending on the number and type of components that you are installing. While the installation process is running, you can monitor the progress by viewing changes to the installation log.

If the installation fails, you are directed to a log file that contains information that you can use to troubleshoot the problem.

To complete a simple installation for the TADDM server, perform the following steps:

1. Insert the TADDM Disk 1 installation DVD for your supported operating system into the CD drive.
2. Open a command prompt, navigate to the CD drive, and enter one of the following commands for your operating system:
   - AIX systems: **`setupAix.bin`**
   - Linux systems: **`setupLinux.bin`**
   - Linux on System z systems: **`setupLinux390.bin`**
   - Solaris systems: **`setupSolarisSparc.bin`**
   - Windows systems: **`setupWin32.exe`**

   You can also run the installation process in console mode, using the **`-console`** option.
3. On the Welcome page, click **Next**.
4. On the Software License Agreement page, read the terms of agreement, and if you accept the terms, continue the installation.
5. Verify the location for the server. You may also click **Browse** to navigate to the location. Click **Next**. See Figure 3-1 on page 52.

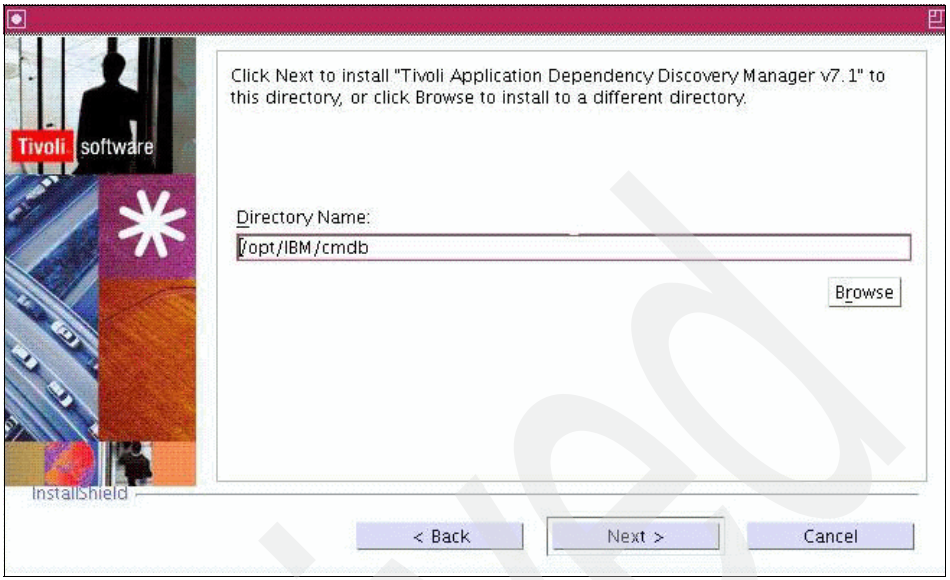

Figure 3-1 Installation location

Important: The two things to remember when you specify a location for the server are:

- Use only ASCII characters in your server installation path.
- Specify an installation path with a name that does not contain spaces.

In addition to causing installation problems, using non-ASCII characters or spaces in the path can also cause problems when starting the TADDM server after a successful installation is complete.

6. Type the user name that you want to use to run the server:
   - Windows systems: The user must belong to the Administrators group. If the user does not exist or is not part of the Administrators group, select the check box to create the user or add the user to the Administrators group.
   - Linux, Solaris, AIX, and Linux on System z systems: The user ID must be non-root.
7. On Windows systems, enter the password that is associated with the user name that you typed in the previous step.
8. Click **Next**. The Installation Type page is displayed.
9. Click **Simple**.

10. To install DB2, click the **Install the DB2 Database** check box. Click **Next**. See Figure 3-2.

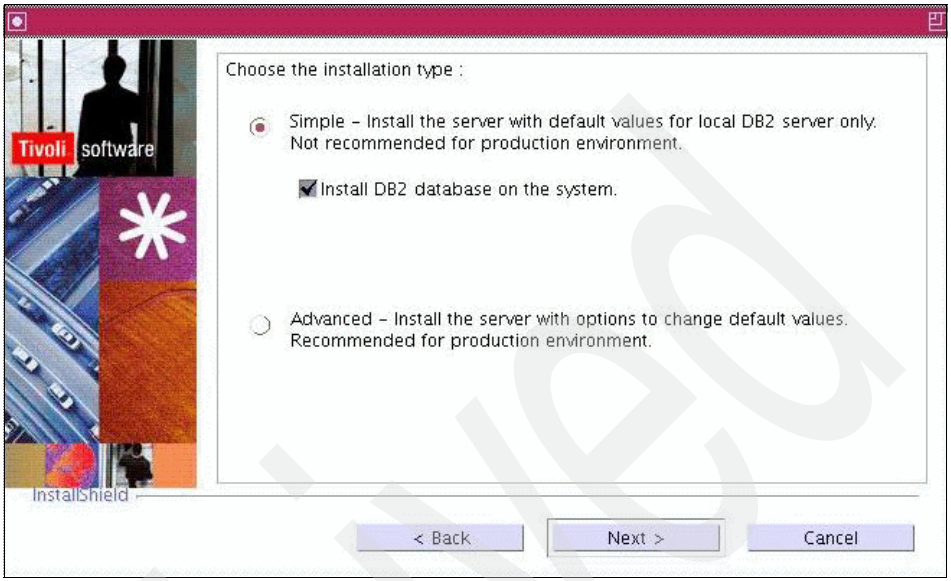

Figure 3-2 Installation type

To complete the installation of the DB2 database, perform the following steps:

a. Verify the name for the database.
b. Verify the user ID for the DB2 database instance.
c. Type the password for the DB2 database instance.
d. Verify the secondary (archive) user ID for the database.
e. Type the password for the archive user ID.

11. If you are not installing DB2, a page with DB2 database configuration parameters for the server is displayed.

To verify the configuration information for the DB2 database, complete the following steps:

a. Verify the host name for the database.
b. Verify the database port.
c. Type the database node name. The database node name is for the DB2 software client.
d. Verify the database name.
e. Verify the user ID for the database instance.

f. Type the password for the database instance.

g. Verify the secondary (archive) database user ID.

h. Type the password for the archive user ID.

12. Click **Next**.

13. Review the summary information and click **Install**.

14. If you are installing DB2, complete one of the following procedures to install DB2 software:

- For Linux, Solaris, AIX, and Linux on System z operating systems:
  i. Change to the root directory.
  ii. Unmount the CD drive.
  iii. Insert the TADDM Disk 2 installation DVD for your supported operating system into the CD drive.
  iv. Click **Next**.
- For Windows operating systems:
  i. Insert the TADDM Disk 2 installation DVD for your supported operating system into the CD drive.
  ii. Click **Next**.

After the DB2 installation is completed successfully, perform one of the following procedures:

- For Linux, Solaris, AIX, and Linux on System z operating systems:
  i. Unmount the CD drive.
  ii. Insert the TADDM Disk 1 installation DVD for your supported operating system into the CD drive.
  iii. Click **Next**.
- For Windows operating systems:
  i. Insert the TADDM Disk 1 installation DVD for your supported operating system into the CD drive.
  ii. Click **Next**.

15. After the installation is completed successfully, a page, indicating a successful installation, is displayed. Click **Finish** to close the installation program. See Figure 3-3 on page 55.

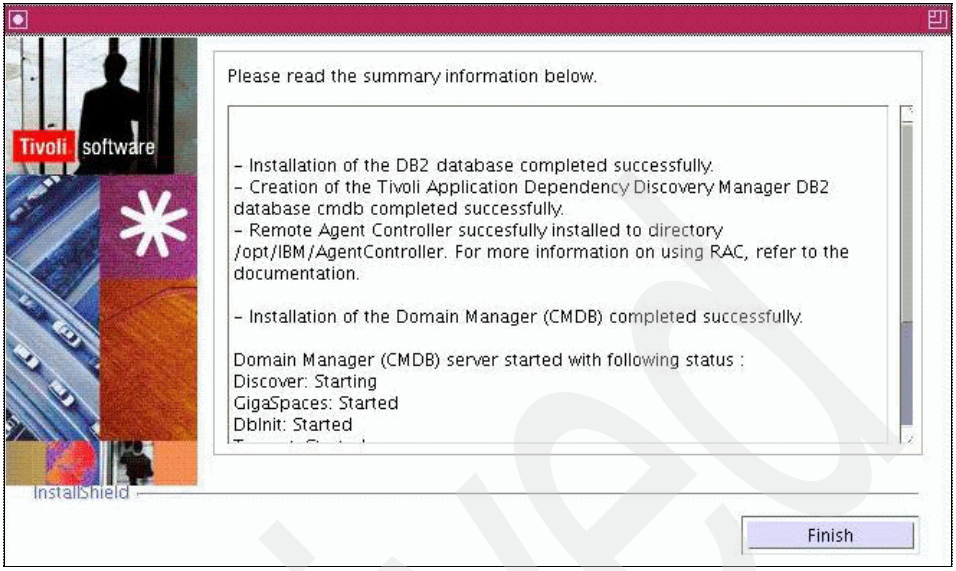

Figure 3-3 Finish page

3.4 Advanced installation

This procedure is intended for resources that meet the supported hardware and software criteria. You can install either the TADDM Domain Server or the TADDM Enterprise Domain Server, and use a remote database. To use this procedure, log in with the user ID that you want to use for installing TADDM. Using a root or administrator ID is not necessary.

Note for Oracle database: To support multilingual data, create the Oracle database with the *AL32UTF8 character set*. Otherwise, data that is in languages other than English might not display correctly. If you get the Oracle message, OALL8 is in inconsistent state, TADDM cannot access some of the national language-specific text in your database. Re-create your database with the correct character set.

If you are installing on a dual-stack system that supports both the IPv4 and IPv6 protocols, make sure any numeric IP addresses that you specify during the installation process are IPv4 addresses.

Note: This procedure assumes a remote DB2 or Oracle server is installed, a database, and the primary and secondary users are created.

For the manual creation of the TADDM database, the script `make_db2_db` can be used to create a DB2 database. Furthermore, the primary and the archive user are created with the script. For an Oracle database, the script `make_ora_db` can be used. These scripts can be found on the TADDM server in the `<COLLATION_HOME>/support/bin` folder.

For more information about these tasks, refer to TADDM information center:

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/index.jsp?topic=/com.ibm.taddm.doc_7.1.2/cmdb_welcome.html

This installation can be a lengthy process, depending on the number and type of components that you are installing. While the installation process is running, you can monitor the progress by viewing changes to the installation log.

If the installation fails, you are directed to a log file that contains information that you can use to troubleshoot the problem.

To complete an advanced installation for the server with a remote database, perform the following steps:

1. Insert the TADDM Disk 1 installation DVD for your supported operating system into the CD drive.
2. Open a command prompt, navigate to the CD drive, and enter one of the following commands for your operating system:
   - AIX systems: **`setupAix.bin`**
   - Linux systems: **`setupLinux.bin`**
   - Linux on System z systems: **`setupLinux390.bin`**
   - Solaris systems: **`setupSolarisSparc.bin`**
   - Windows systems: **`setupWin32.exe`**

   A Welcome page is displayed.
3. Click **Next**.
4. On the Software License Agreement page, read the terms of agreement, and if you accept the terms, continue the installation.
5. Verify the location for the server. You can also click **Browse** to navigate to the location. Click **Next**.

Important: The two things to remember when specifying a location for the server are:

- Use only ASCII characters in your server installation path.
- Specify an installation path with a name that does not contain spaces.

In addition to causing installation problems, using non-ASCII characters or spaces in the path can also cause problems when starting the TADDM Server after a successful installation is complete.

6. Type the user name that you want to use to run the server:
   - Windows systems: The user must belong to the Administrators group. If the user does not exist or is not part of the Administrators group, select the check box to create the user or add the user to the Administrators group.
   - Linux, Solaris, AIX, and Linux on System z systems: The user ID must be non-root.
7. On Windows systems, enter the password associated with the user name typed in the previous step.
8. Click **Next**. The Installation Type page is displayed.
9. Click **Advanced**.
10. Click **Next**. A page requesting the type of server that you want to install is displayed.
11. Use one of the following procedures:
    - To install only one server, select **TADDM server**. Click **Next**. A page requesting information about ports for the server is displayed.
    - To install the Domain Manager to manage your servers, select **Enterprise TADDM server**. Click **Next**. A page requesting information about ports for the server is displayed.
12. To install the server, specify the following information in the fields:
    - Web server port number
    - Web server SSL port number
    - GUI server communication port number
    - GUI server communication SSL port number
    - JNDI port number
    - RMI port number
    - Topology manager communication port number
    - Topology builder port number
    - RMID port number
13. Click **Next**. The ports are validated.

14. For a TADDM Enterprise Domain Server installation, skip this step and go to step 15.

    For a TADDM Domain Server installation, complete this step, as follows:

    a. Specify the following information in the fields:

       - Security manager port number
       - Topology manager port number
       - API server port number
       - Change manager port number
       - Report server port number

    b. Click **Next**. The ports are validated.

15. Click **Next**.

16. Type the name for the RMI server host name. The default for this field is `default`.

17. Select the platform binaries that you want to install. If you configure the binaries later, the binaries are pushed to the remote gateway or anchor. If you are not sure which platform binaries that you want to install, select all.

18. Select the mode for the Discovery Manager Server. If you chose to install the TADDM Enterprise Domain Server, select **Distributed**; otherwise, select **Local**.

19. To start the server when the system starts, select **Start the server at system boot**.

20. To start the server after the installation process is complete, select **Start the server after install**.

21. Click **Next**. A page requesting optional information is displayed. The optional information includes the host name and port number for an IBM Service Management server.

22. This step is optional. If you want to connect the server with an IBM Service Management server, type the host name and password for the IBM Service Management server. Click **Next**.

    You can also enter the IBM Service Management server information after the product installation is complete. The information can be entered in the product user interface.

23. On the database type page, select the database type: **DB2** or **Oracle**.

24. This step is optional.

    If you want to install the WebSphere Federation Server, select **Setup WebSphere Federation Server**, and then, use one of the following procedures.

To use a new WebSphere Federation Server, complete these steps:

a. Click **Next**. A page requesting that you select the type of WebSphere Federation Server setup is displayed.

b. Select **Install a new instance of WebSphere Federation Server**. For a new WebSphere Federation Server, you can install IBM WebSphere Federation Server, version 9.1.

c. Click **Next**. A page requesting information about the WebSphere Federation Server is displayed.

d. Complete the following steps:

   i. Type the database name for the WebSphere Federation Server to use when holding the data source mapping between the TADDM Database and other external databases. The TADDM Server and the WebSphere Federation Server use different databases.

   ii. Type the DB2 instance user ID for WebSphere Federation Server database.

   iii. Type the password for the DB2 instance user ID.

   iv. Retype the password for the DB2 instance user ID.

To use an existing WebSphere Federation Server, complete these steps:

a. Click **Next**. A page requesting that you select the type of WebSphere Federation Server setup is displayed.

b. Select **Use an existing WebSphere Federation Server**.

c. Click **Next**. A page requesting information about the WebSphere Federation Server is displayed.

d. Complete the following steps:

   i. Type the database host name.

   ii. Type the database port number.

   iii. Type the database name for the WebSphere Federation Server to use when holding the data source mapping between the TADDM Database and other external databases. The TADDM Server and the WebSphere Federation Server use different databases.

   iv. Type the DB2 instance user ID for WebSphere Federation Server database.

   v. Type the password for the DB2 instance user ID.

   vi. Retype the password for the DB2 instance user ID.

If you have problems with IBM WebSphere Federation Server, report thes to the IBM Software Support team for IBM WebSphere Federation Server.

25. Click **Next**. A page with database configuration information is displayed.
    - To configure the DB2 database, complete the following steps:
        i. Verify the host name for the database.
        ii. Verify the database port.
        iii. Type the database node name. The database node name is for the DB2 software client.
        iv. Verify the database name.
        v. Verify the user ID for the database instance.
        vi. Type the password for the database instance.
        vii. Verify the additional database user ID.
        viii. Type the password for the additional database user ID.
    - To configure the Oracle database, complete the following steps:
        i. Verify the host name for the database.
        ii. Verify the database port.
        iii. Verify the Oracle SID.
        iv. Verify the Oracle user ID.
        v. Type the password for the Oracle database.
        vi. Verify the additional database user ID.
        vii. Type the password for the additional database user ID.
26. Click **Next**. A page with the user registry options is displayed.
27. Use one of the following procedures:
    - To use a file-based user registry, select the corresponding radio button. No additional configuration information is necessary.
    - To use an LDAP user registry, the LDAP server must be installed and running on the local or remote system. To add additional configuration information to identify the LDAP server, complete the following steps:
    a. Select **LDAP user registry**.
    b. Click **Next**. A page requesting information about the LDAP server is displayed.
    c. To configure the LDAP server for user registry, complete the following steps:
        i. Type the LDAP host name.
        ii. Verify the LDAP port number.

iii. Type the user ID to use for the LDAP server. The authorization policy is case sensitive. Type the letters using the correct case.

iv. Type the password for the LDAP server user ID. The authorization policy is case sensitive. Type the letters using the correct case.

v. Retype the password for the LDAP server user ID. The authorization policy is case sensitive. Type the letters using the correct case.

vi. Type the LDAP Base Distinguished Name. The authorization policy is case sensitive. Type the letters using the correct case.

vii. This step is optional. Select **Use anonymous binding**. If you use the LDAP anonymous binding function, the user ID and password fields are not editable.

d. Click **Next**. A page requesting LDAP configuration parameters is displayed. Default parameters are provided for each field. You can use the default parameters or provide specific values for your environment.

e. To complete the configuration of the LDAP server for user registry, perform the following steps:

i. Verify the LDAP user object class.
ii. Verify the LDAP UID naming attribute.
iii. Verify the LDAP group object class name.
iv. Verify the LDAP group naming attribute.
v. Verify the LDAP group member attribute.

– To use WebSphere federated repositories, the WebSphere Application Server must be installed and running on the local or remote system. To add additional configuration information that is necessary to identify the WebSphere Application Server, complete the following steps:

a. Select **WebSphere Federated Repositories**.

b. Click **Next**. A page requesting information about the WebSphere Application Server is displayed.

c. To configure the WebSphere Application Server for user registry, complete the following steps:

i. Type the WebSphere Application Server host name.

ii. Verify the WebSphere Application Server port number.

iii. Verify the WebSphere Application Server HTTP transport port number.

iv. Type the user ID to use for the WebSphere Application Server. The authorization policy is case sensitive. Type the letters using the correct case.

v. Type the password for the WebSphere Application Server user ID. The authorization policy is case sensitive. Type the letters using the correct case.

vi. Retype the password for the WebSphere Application Server user ID. The authorization policy is case sensitive. Type the letters using the correct case.

28. Click **Next**. A summary page is displayed.

29. Review the summary information and click **Install**. The progress of the installation is not displayed.

30. After the installation is completed successfully, a page, indicating a successful installation, is displayed. Click **Finish** to close the installation program.

3.5 Silent installation

You can use the -**silent** flag for a silent installation.

Silent installation can be a lengthy process, depending on the number and type of components that you are installing. While the installation process is running, you can monitor the progress by viewing changes to the installation log.

To run a silent installation of the server, complete the following steps:

1. Go to the server and use one of the following procedures:
   - For Linux, Solaris, AIX, and Linux on System z operating systems, use either root or non-root user ID to log in.
   - For the Windows operating systems, use a Windows logon ID with Administrator authority.
2. To generate a response file, complete one of the following options:
   - Run the installation wizard with the **record** option, as shown in the following list. As you proceed through the pages of the installation wizard, your answers are captured and the response file is generated. When you complete the installation, the response file (for example, `install.rsp`) is available in the `tmp` directory.
     - For AIX operating systems:

       ```
       setupAix.bin -options-record /tmp/install.rsp
       ```

     - For Linux operating systems:

       ```
       setupLinux.bin -options-record /tmp/install.rsp
       ```

- For Linux on System z operating systems:

  ```
  setupLinux390.bin -options-record /tmp/install.rsp
  ```

- For Solaris operating systems:

  ```
  setupSolarisSparc.bin -options-record /tmp/install.rsp
  ```

- For Windows operating systems:

  ```
  setupWin32.exe -options-record c:\temp\install.rsp
  ```

– If you do not want to generate a response file by recording input values to a response file, you can create a response file template to use for the silent installation. To create a response file, run the following command for your operating system:

  - For AIX operating systems:

    ```
    setupAix.bin -options-template /tmp/install.rsp
    ```

  - For Linux operating systems:

    ```
    setupLinux.bin -options-template /tmp/install.rsp
    ```

  - For Linux on System z operating systems:

    ```
    setupLinux390.bin -options-template /tmp/install.rsp
    ```

  - For Solaris operating systems:

    ```
    setupSolarisSparc.bin -options-template /tmp/install.rsp
    ```

  - For Windows operating systems:

    ```
    setupWin32.exe -options-template c:\temp\install.rsp
    ```

  You must edit the template response file, `install.rsp`, with the appropriate values before you can use it. The template response file includes instructions for each value.

  A sample installation response file is provided:

  - For AIX, Linux, Linux on System z, and Solaris operating systems:

    ```
    support/samples/sample_taddm_install_unix.rsp
    ```

  - For Windows operating systems:

    ```
    support\samples\sample_taddm_install_win.rsp
    ```

  You can use any text editor to edit the response file.

  Important: If you are installing on a dual-stack system that supports both the IPv4 and IPv6 protocols, make sure that any numeric IP addresses that you specify in the response file are IPv4 addresses.

3. Run the silent installation using the response file:
   - For AIX operating systems:

     ```
     setupAix.bin -options /tmp/install.rsp -silent -is:log
     /tmp/install.log
     ```

   - For Linux operating systems:

     ```
     setupLinux.bin -options /tmp/install.rsp -silent -is:log
     /tmp/install.log
     ```

   - For Linux on System z operating systems:

     ```
     setupLinux390.bin -options /tmp/install.rsp -silent -is:log
     /tmp/install.log
     ```

   - For Solaris operating systems:

     ```
     setupSolarisSparc.bin -options /tmp/install.rsp -silent -is:log
     /tmp/install.log
     ```

   - For Windows operating systems:

     ```
     setupWin32.exe -options c:\temp\ install.rsp -silent -is:log
     c:\temp\install.log
     ```

3.6 Anchor and gateway installation

Many companies restrict access to areas of their network by using firewalls. TADDM can make use of a server (also known as a anchor) within the firewall zone to perform discoveries on behalf of the primary TADDM server.

If any Windows servers are within the firewall zone, then a Windows gateway is also required.

You do not have to configure anchors or gateways during the installation process, but you must include anchors and gateways in your installation plan, verifying the system requirements for candidate systems. After the installation of the TADDM server, you can use the Product Console to configure hosts to serve as anchors or gateways on your network.

All required software is automatically deployed to the anchor server at the time of the first discovery.

Anchors work on all operating systems supported for TADDM.

Note: The anchor does not start if the sshd daemons are not configured to allow port forwarding. Anchors rely on local port forwarding to be enabled in the remote anchor server.

3.6.1 Anchor considerations

When you install anchors, consider the following information:

- Sudo access is required on the TADDM server and each TADDM anchor. To verify that sudo access is properly configured, run the following commands:

```
cd $COLLATION_HOME/bin
sh ./stop-local-anchors.sh
```

Note: The proper way to stop anchors is by stopping the TADDM server. When TADDM server stops will also stop anchor servers. If an anchor needs to be stopped without TADDM server stopping, kill the java process running under the UNIX account for the anchor server.

- TADDM server uses SSH (port 22) to communicate with the anchor server. In addition, the TADDM server initiates all communications, therefore only port 22 has to be opened on the firewall for outbound communication.
- For UNIX anchors, you have to log into the designated anchor server with root user and create an anchor service User ID.
- For Windows anchors, you have to log into the designated anchor server with administrator user to create an anchor service User ID.

Notes:

- For UNIX, prior to designating a UNIX anchor server, a service account user ID with root user authority associated to *sys* group must be created.
- For Windows, prior to designating a Windows anchor server, a service account user ID with administrative rights belonging to the administrator's group must be created.

- Optionally, the Network Mapper (Nmap) may be used together with TADDM. Nmap must be installed on the TADDM server and each anchor server. Use the latest version of Nmap.

Note: Nmap a utility for port scanning large networks,

3.6.2 Gateway considerations

When you install gateways, consider the following information:

1. For potential Windows gateway, verify that the WinSSHD or Cygwin is installed and configured correctly, so that the Windows server can receive SSH requests.
2. For target Windows servers, raw socket support must be enabled.
3. Add Windows gateway SSH user and password to the access list on the Product Console.
4. Add Windows target service user ID and password to the access list on the Product Console.
5. Execute a test command on the gateway host using SSH:
   - UNIX: `testssh.py <ip> <command>`
   - Windows: `testssh.bat <ip> <command>`
6. Execute a test command to verify that WMI is installed on the gateway system:
   - UNIX: `wmiexec.jy <ip>`
   - Windows: `wmiexec.bat <ip>`
7. Execute a test command on a target Windows system:
   - UNIX: `wmiexec.jy <ip> <command>`
   - Windows: `wmiexec.bat <ip> <command>`

4

Configuration

In this chapter, we describe the configuration tasks for IBM Tivoli Application Dependency Discovery Manager (TADDM) v7.1. We explain various ways to configure TADDM, and TADDM features and interactions with other systems.

This chapter contains the following topics:

- "Performance tuning for databases" on page 68
- "Discovery scopes" on page 69
- "Access list" on page 73
- "Discovery profiles" on page 76
- "Anchors and gateways" on page 77
- "Custom server templates" on page 81
- "Application templates" on page 86
- "Application descriptors" on page 88
- "Security" on page 96

4.1 Performance tuning for databases

The DB2 and Oracle databases run more efficiently for TADDM when you complete performance tuning tasks.

4.1.1 RUNSTATS command

You can update DB2 statistics using the RUNSTATS command. You should run the command after any task significantly changes database contents, for example, after a discovery or bulkload.

In general, you should run the RUNSTATS command once a week. This task can be completed manually, or by using the DB2 facility to automatically enable the RUNSTATS command.

A sample script that you can use to enable the RUNSTATS command is available. You can use the sample provided to design your own production scripts. By using a production script that you develop, you can automate the process so that the RUNSTATS command runs on all TADDM database tables with one command.

The script is located in the following paths of your operating system:

- For Linux, Solaris, AIX, and Linux on System z operating systems:

  `$COLLATION_HOME/support/bin/runstatus_db2_catalog.sql`

- For Windows operating systems:

  `%COLLATION_HOME%\support\bin\runstatus_db2_catalog.sql`

Example 4-1 shows how you can run the sample script.

Example 4-1 Running the sample script

```
su - <db2 instance owner>
db2 connect to <cmdb>
db2 -stf runstats_db2_catalog.sql
```

You should develop your own production script to run the RUNSTATS command for your environment at an appropriate frequency to ensure good database performance.

4.1.2 Query optimizer

The DB2 query optimizer benefits from having recent statistics for the TADDM tables. For example, the query optimizer can help estimate how much buffer pool is available at run time.

A sample script in the TADDM installation directory is useful for the query optimizer. Using this script, you can output the database commands for the Oracle database and DB2 database to update the statistics on the TADDM tables.

The script is located in the following paths of your operating system:

- For Linux, Solaris, AIX, and Linux on System z operating systems:

  `<TADDM_install_dir>/dist/support/bin/gen_db_stats.jy`

- For Windows operating systems:

  `<TADDM_install_dir>\dist\support\bin\gen_db_stats.jy`

Example 4-2 shows how you can run the sample script.

Example 4-2 Running the sample script

```
cd <TADDM_install_dir>/dist/support/bin
./gen_db_stats.jy ><tmpdir>/TADDM_table_stats.sql
```

After these command are complete, you should copy the file to the database server and run the following commands:

```
DB2: db2 -tvf
Oracle: sqlplus
```

4.2 Discovery scopes

A discovery *scope* identifies the network devices, computer systems and other components in the infrastructure to be discovered by the TADDM server. You can specify scopes by host names, IP addresses, IP ranges, and subnets. *Scope sets* put related scopes together.

4.2.1 Adding a scope set

To add a scope set, complete the following steps from the Product Console:

1. Select **Discovery** → **Scope**. The Scope page opens.
2. Click **Add Set**. The Scope Set Name window opens.
3. Enter a name for the scope set.
4. Click **OK**. A new scope set is listed on the Scope Sets list.

4.2.2 Deleting a scope set

To delete a scope set, complete the following steps from the Product Console:

1. Select **Discovery** → **Scope**. The Scope page opens.
2. Select a scope set.
3. Click **Delete Set**. The Confirm Deletion window opens.
4. Click **Yes**. The Scope Sets list appears updated.

4.2.3 Adding a scope

To add a scope, complete the following steps from the Product Console:

1. Select **Discovery** → **Scope**. The Scope page opens.
2. Select a scope set, and click **Add**. The Add Scope window opens. See Figure 4-1 on page 70.

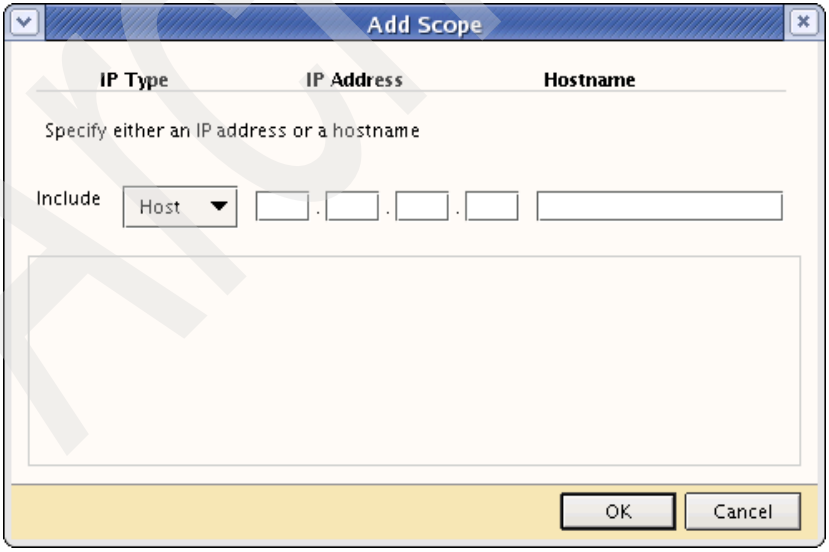

Figure 4-1 Add Scope window

3. Select a type for the scope:
   - *Subnet* type is a class C subnet defined by an IP address and a mask.
   - *Range* type is an IP range defined by a start and an end address.
   - *Host* type is a single host defined by either and IP address or a host name.
4. Enter the information related to the scope. This scope information must be unique within the scope set.
5. To make exclusions from a subnet or range, click **Add Exclusion**.
6. Enter the information related to the exclusion.
7. Click **OK**. A new scope appears on the list.

4.2.4 Editing a scope

To edit a scope, complete the following steps from the Product Console:

1. Select **Discovery** → **Scope**. The Scope page opens.
2. Select a scope set. The Scope list of that scope set opens.
3. Select a scope, and click **Edit**. The Edit Scope window opens.
4. Edit the information related to the scope.
5. Click **OK**. The scope appears updated on the list.

4.2.5 Deleting a scope

To delete a scope, complete the following steps from the Product Console:

1. Click **Discovery** → **Scope**. The Scope page opens.
2. Select a scope set. The Scope list of that scope set opens.
3. Select a scope, and click **Delete**. The Confirm Deletion window opens.
4. Click **Yes.** The Scope list opens updated.

4.2.6 Loading a scope set from a file

Instead of Product Console, you can set up scopes by creating a scope file and loading it into the TADDM server with the `loadscope` command.

The following information is about scope files:

- You can use any number of entries in any combination of the following types in a scope file:
  - *Subnet* (for example, `1.2.3.4/255.255.255.0`)
  - *Range* (for example, `1.2.3.4-5.6.7.8`)
  - *Address* (for example, `1.2.3.4`)
- The format of the entries in a scope file is as follows:

  ```
  scope, [exclude_scope:exclude_scope...],[description]
  ```
- You can use IP addresses only in the scope file. Host names cannot be used.
- Each entry resides on a separate line.
- Address scopes cannot include exclusions.
- You can prefix a comment line with a number sign (#).
- Invalid entries are ignored.

Example 4-3 on page 72 shows a sample scope file.

Example 4-3 Example scope file

```
# This is a comment
10.10.10.10,,
10.10.10.20,,
10.10.10.30,,
10.10.10.0/255.255.255.0,10.10.10.2:10.10.10.3,
10.10.10.2-10.10.10.9,10.10.10.4:10.10.10.5,
10.10.10.88,,
10.10.10.999,,.
```

To load the scope, you can use the **loadscope** command located at the directory `$COLLATION_HOME/bin`. The syntax for the **loadscope** command is:

```
loadscope.jy [-d] -u <username> -p <password> -s <scopeset> load <scopefile>
```

Parameters in the **loadscope** command are:

- `-d`

  This parameter turns on verbose debug logging.
- `-u username`

  This parameter is the user name to access the TADDM server.
- `-p password`

  This parameter is the password for the user name.

- `-s scopeset`

  This parameter is the scope set to use for loading the scope elements. If the scope set already exists, its contents are replaced by the scope elements contained in the scope file.

- `load`

  This parameter loads the scope elements to the system, appending new elements to existing elements.

- `scopefile`

  This parameter is the file containing the scope elements.

A sample of the **`loadscope`** command is as follows:

```
loadscope.jy -u administrator -p cmdb -s Windows load scopefile
```

For example, the following command loads a file named `/tmp/win.scope` to the existing Windows Servers scope set:

```
$ ./loadscope.jy –u administrator –p taddm –s “Windows Servers” load
/tmp/win.scope
```

Using the command line to export scopes

You might want to export scopes that you have created in TADDM. For example, you might have to move scopes from one TADDM server to another when reconfiguring from a single server architecture to a multiple server architecture.

The output from the scope export command is an XML-formatted file. From `${TADDM_HOME}/dist/sdk/bin` location, use the following command:

```
$./api.sh -u <username> -p <password> find [--depth=5] Scope
```

To import scopes from an XML file, run the following command:

```
./api.sh -u administrator -p collation import scope_file.xml
```

4.3 Access list

The *access list* is a collection of user names, passwords and Simple Network Management Protocol (SNMP) community strings that the TADDM server uses to access and discover the configuration items in the infrastructure. To access and discover network elements, computer systems, applications, and more, enter an appropriate type of access list entry for that configuration item. Access list entries are categorized by device type, computer system type (Windows or non-Windows), application type, and can be restricted to a scope.

When accessing a configuration item, the TADDM server tries access list entries of the related type by starting at the top of the list in the GUI. To increase the discovery speed, you can order the entries from the most general to the most specific.

4.3.1 Adding an access list entry

To add an access list entry, complete the following steps from the Product Console:

1. Select **Discovery** → **Access List**. The Access List page opens.
2. Click **Add**. The Access Details window opens. See Figure 4-2 on page 74.

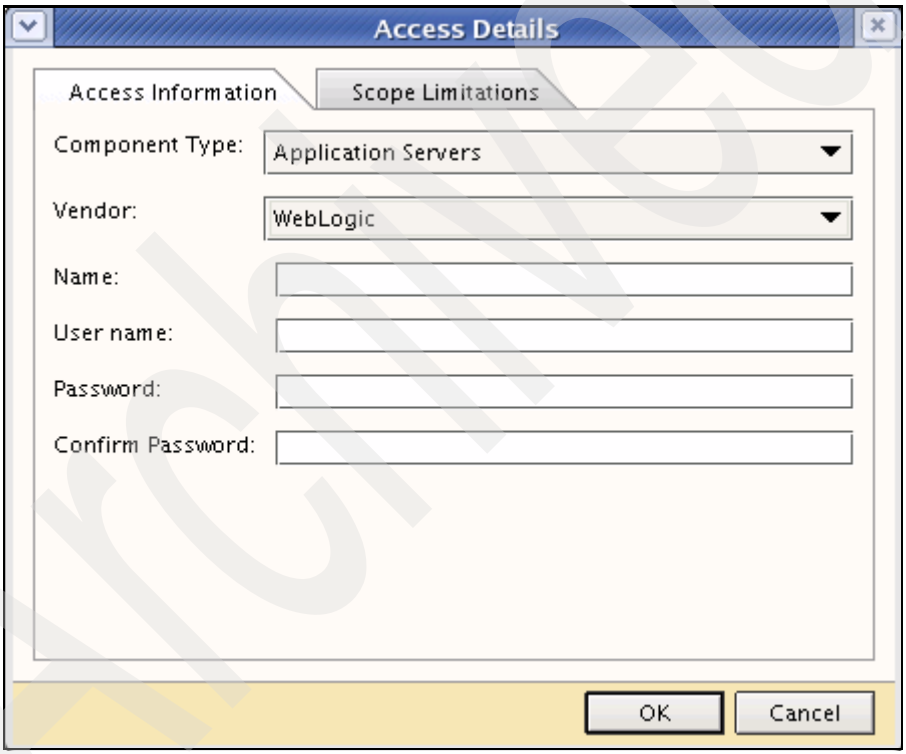

Figure 4-2 Access Details window

3. Click the **Access Information** tab.
4. Select a Component Type.
5. Select a Vendor if applicable.
6. Enter a name for the entry.

7. Enter a user name and password combination, or SNMP community string.
8. Click **Scope Limitations** tab.
9. If you want to use the entry for all scopes, select **Entire scope**.
10. If you want to restrict the entry to a scope, select **Limit to selected scope**, and select a scope set from the list.
11. Click **OK**. A new access list entry appears at the bottom of the list.

4.3.2 Editing an access list entry

To edit an access list entry, complete the following steps from the Product Console:

1. Select **Discovery** → **Access List**. The Access List page opens.
2. Select an access list entry, and click **Edit**. The Access Details window opens.
3. Click **Access Information** tab.
4. Click **Change**. SNMP community string or password field becomes editable.
5. Edit SNMP community string or password information.
6. Click **Scope Limitations** tab.
7. Edit the scope restriction information.
8. Click **OK**. The access list entry appears updated on the list.

4.3.3 Deleting an access list entry

To delete an access list entry, complete the following steps from the Product Console:

1. Select **Discovery** → **Access List**. The Access List page opens.
2. Select an access list entry, and click **Delete**. The Confirm Deletion window opens.
3. Click **Yes.** the scope list appears updated.

4.3.4 Changing the order of the access list entries

To change the order of the access list entries, complete the following steps from the Product Console:

1. Click **Discovery** → **Access List**. The Access List page appears.
2. Select an access list entry.
3. Click **Move Up** or **Move Down** to move the entry until the position you want.

4.4 Discovery profiles

A *discovery profile* is a collection of sensors to be run during discovery. TADDM comes with three default profiles, as described in Table 4-1.

Table 4-1 Default profiles

Level	Type of discovery	What is discovered
Level 1	Credential-less	Basic information about active computer systems: host name, FQDN, OS release level, IP address, and open ports. Network operating systems, such as Cisco and Alteon, are also discovered. On Windows and zLinux hosts, the MAC address is discovered.
Level 2	OS credentials only	Detailed information about all of the operating systems.
Level 3	Full credential	Entire application infrastructure: deployed software components, physical servers, network devices, virtual LANs, and hosts.

Default profiles cannot be edited, but new profiles can be created or cloned from existing profiles. Also, sensors can be cloned and configured according to your requirements.

4.4.1 Creating a discovery profile

To create a discovery profile, complete the following steps from the Product Console:

1. Select **Discovery** → **Discovery Profiles**. The Discovery Profiles page opens.
2. Click **New**. The Create New Profile window opens.
3. Type a profile name and a description.
4. In order to create the new profile as a copy of an existing profile, select a profile from the Clone existing profile list.
5. Click **OK**. A new profile appears on the Discovery Profiles list.

4.4.2 Editing a discovery profile

To edit a nondefault discovery profile complete the following steps from the Product Console:

1. Select **Discovery** → **Discovery Profiles**. The Discovery Profiles page opens.
2. Select a nondefault profile.
3. On the Sensor Configuration page, you can enable and disable sensors as you want.
4. Save the profile configuration by clicking **Save**.

4.4.3 Deleting a discovery profile

To delete a nondefault discovery profile, complete the following steps from the Product Console:

1. Select **Discovery** → **Discovery Profiles**. The Discovery Profiles page opens.
2. Select a nondefault profile, and click **Delete.** The Confirm Deletion window opens.
3. Click **Yes.** The Discovery Profiles list appears updated.

4.5 Anchors and gateways

The TADDM server uses SSH protocol to communicate with the computer systems and components during discovery. However, there are two cases when the server must communicate through a proxy to collect system information:

- When using a firewall between the TADDM server and other sections of your network
- When discovering and collecting information from Windows systems

Anchors are used to discover components across a firewall, and Windows *gateways* are used to discover Windows computer systems. SSH connection must be granted between the TADDM server and anchors or gateways during a discovery that requires anchors or gateways. To ensure this connection, the access list must contain administrator access credentials for the anchors or the gateways, and the related anchors and gateways must be included in discovery scope.

See “Anchor server OS and hardware” on page 19 for anchor and gateway specifications and 3.6, “Anchor and gateway installation” on page 64 for anchor and gateway installation steps.

During the initial Level 2 discovery, a few small Windows Management Interface (WMI) files are pushed onto the Windows gateways and the targeted Windows machine, and WMI restarts, so that WMI is available on the gateway (or gateways) and target machine (or machines).

The following list describes the WMI provider files that are located on the TADDM server in the `${COLLATION_HOME}/dist/lib/ms/gateway` directory and pushed to the Windows gateways and targeted Windows machine `%SystemRoot%\System32\wbem` directory:

- The `TaddmWmi.dll` file is the actual WMI provider that will be registered. It in turn runs `TaddmWmi.exe`.
- The `TaddmWmi.mof` file is specifies what new WMI methods are provided by the provider (`TaddmWmi.dll`).
- The `TaddmWmi.exe` file is called by the `TaddmWmi.dll` WMI provider to run a command.
- The `TaddmPortMap.exe` file is only copied if the endpoint is Win2000. It Provides functionality similar to the netstat tool.
- `The aports.dll` file is only copied if the endpoint is Win2000. Used by TaddmPortMap.

After the files are deployed, WMI on all Windows targets and gateways are restarted. WMI is also restarted after a communication failure. These settings are configurable in the `collation.properties` file, as follows:

- Restart WMI if a WMI error is encountered during AutoDeploy:
  - `com.collation.RestartWmiOnAutoDeploy=true`
  - `com.collation.RestartWmiOnAutoDeploy.1.2.3.4=true`
- Restart WMI if a WMI error is encountered (except during AutoDeploy):
  - `com.collation.RestartWmiOnFailure=true`
  - `com.collation.RestartWmiOnFailure.1.2.3.4=true`

4.5.1 Adding an anchor or gateway

To add an anchor or gateway, complete the following steps from the Product Console:

1. Select **Discovery** → **Anchors and Gateways**. The Anchors and Gateways page opens.
2. Click **Add**. The Add Anchor window opens. See Figure 4-3 on page 79.

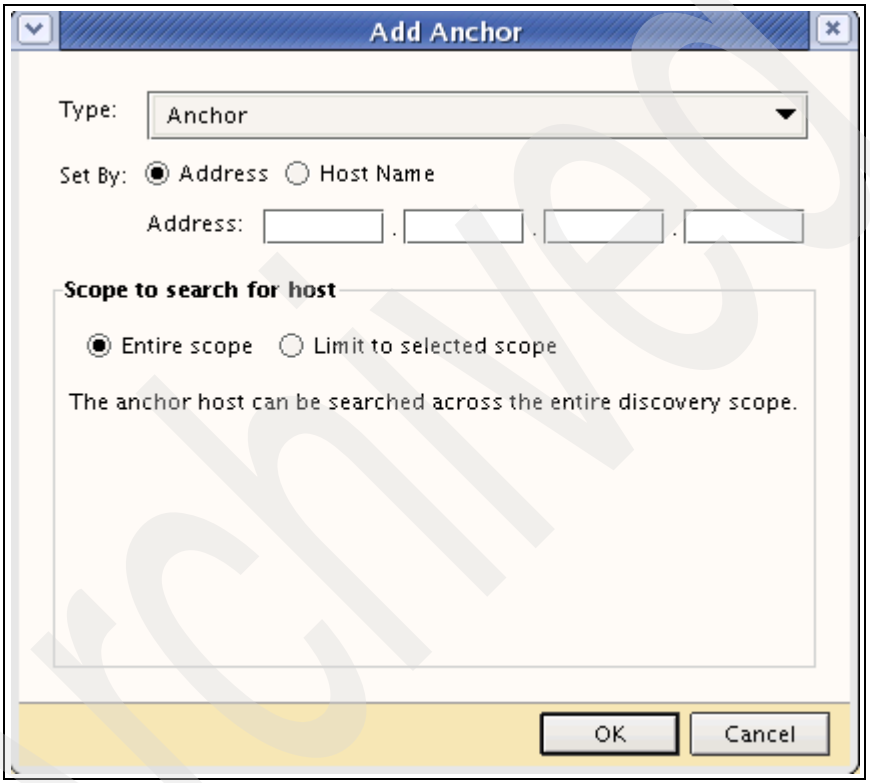

Figure 4-3 Add Anchor window

3. Select the Type: either **Windows Gateway** or **Anchor**.
4. Select Set By: either **Address** for IP address or **Host Name** for host name.
5. Select the scope to search for host:
   - To use the anchor or gateway for all scopes, select **Entire scope**.
   - To restrict the anchor or gateway to a scope, select **Limit to selected scope**, and select a scope set from the list.
6. Click **OK**. A new anchor or gateway appears in Anchors and Gateways list.

4.5.2 Editing an anchor or gateway

To edit an anchor or gateway, complete the following steps from the Product Console:

1. Select **Discovery** → **Anchors and Gateways**. The Anchors and Gateways page opens.
2. Select an anchor or gateway, and click **Edit Scope**. The Add Anchor window opens.
3. Edit the Scope to search for host:
   - To use the anchor or gateway for all scopes, select **Entire scope**.
   - To restrict the anchor or gateway to a scope, select **Limit to selected scope**, and select a scope set from the list.
4. Click **OK**.The Anchors and Gateways list appears updated.

4.5.3 Deleting an anchor or gateway

To delete an anchor or gateway, complete the following steps from the Product Console:

1. Select **Discovery** → **Anchors and Gateways**. The Anchors and Gateways page opens.
2. Select an anchor or gateway, and click **Delete**. The Confirm Deletion window opens.
3. Click **Yes**. The Anchors and Gateways list appears updated.

4.5.4 Setting an anchor port

You can change the anchor port number if the default port is in use. To set an anchor port complete the following steps from the Product Console:

1. Select **Discovery** → **Anchors and Gateways**. The Anchors and Gateways page opens.
2. Select an anchor, and click **Set Anchor Port**. The Edit Port Number window opens.
3. Type the port number for the anchor.
4. Click **OK**. The Anchors and Gateways list appears updated.

4.6 Custom server templates

You can define *custom server templates* to discover and categorize servers that are not supported by TADDM by default. In your infrastructure, you might have in-house developed or customized servers. During discovery these servers will be categorized as unknown servers. By defining custom server templates for unknown servers you can take advantage of tracking, and showing them in topology. Also, you can capture and track the configuration files of the unknown server by specifying them in template. Several primary reasons that custom server templates are important in the use of TADDM include:

- Categorizing running software on a computer system
- Suppressing extraneous software servers from the topology
- Collecting configuration files from all computer systems of a certain type
- Populating a business application with the software that is matched by specific custom server templates

During discovery, a matching custom server template is searched for an unknown server from top to bottom as listed in the GUI. An unknown server can match multiple custom server template, but it is classified as the top one because template-matching is applied from top to bottom in the custom server list and stops at the first match. So, a better approach is to order the custom server templates from the most specific to the most general.

4.6.1 Adding a custom server

To add a custom server, complete the following steps from the Product Console:

1. Select **Discovery** → **Custom Servers**. The Custom Servers page opens.
2. Click **Add**. The Custom Server Details window opens. See Figure 4-4 on page 82.

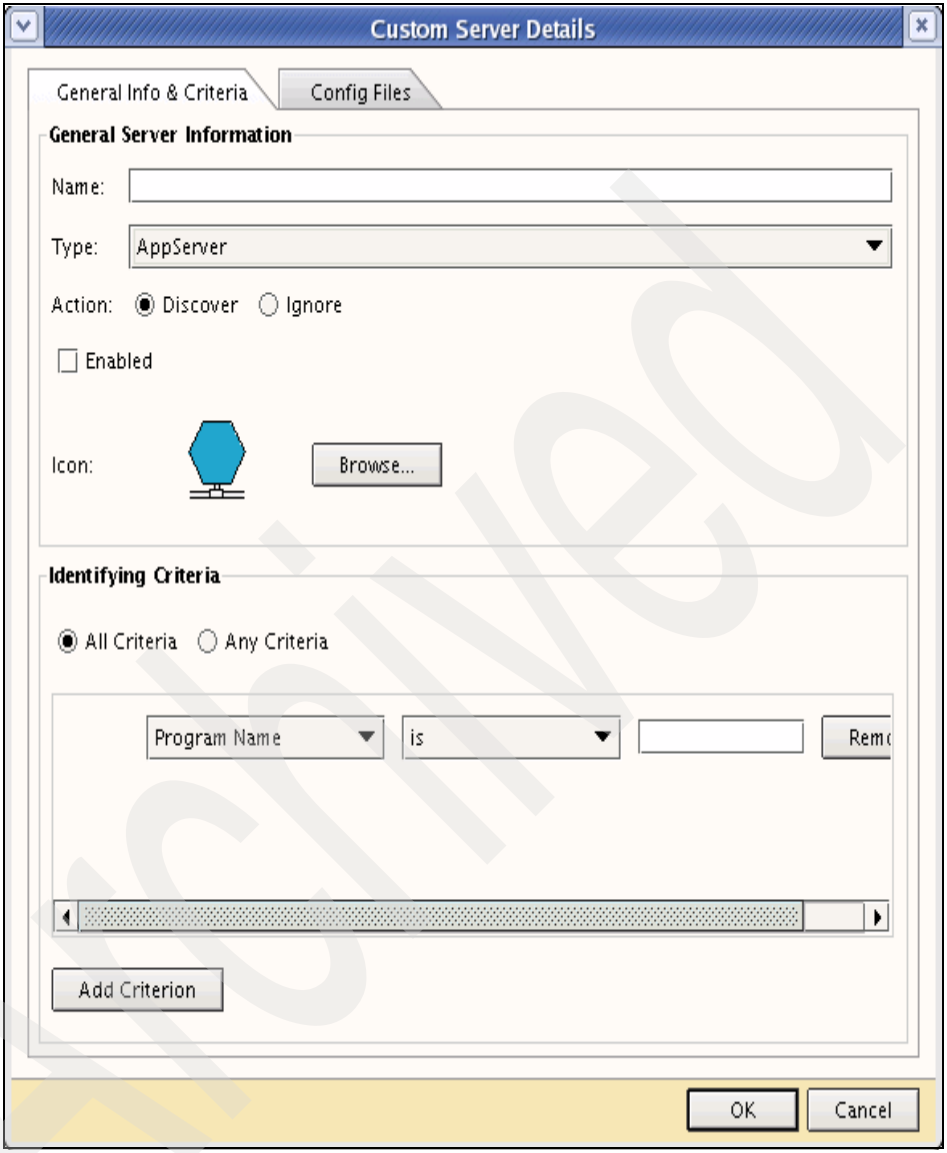

Figure 4-4 Custom Server Details

3. Click **General Info & Criteria** tab.
4. Type a name for the custom server.
5. Select a type.

6. Select one of the following actions:
   - **Discover** means a configuration item matching the custom server is created and information related to the custom server is collected during discovery.
   - **Ignore** means a configuration item will not be created for this custom server.
7. Select the Enabled check box to enable the custom server.
8. Select an icon.
9. Select a connector:
   - **All Criteria** means a process matches the custom server if all of the specified criteria matches.
   - **Any Criteria** means a process matches the custom server if any of the specified criteria matches.
10. Define a criterion by selecting an attribute and an operator, and then typing a value. A criterion can be defined by following the attributes:
    - Program name
    - Argument
    - Environment
    - Port
    - Windows Service Name
11. If you want to compose a complex identifying criteria, click **Add Criterion** and define a new criterion.
12. If you want to remove a criterion, click **Remove**.

13. Click the **Config Files** tab. Perform the following tasks:
    - To add a configuration file:
        i. Click **Add**. The Search Path for Capture File window opens (Figure 4-5).

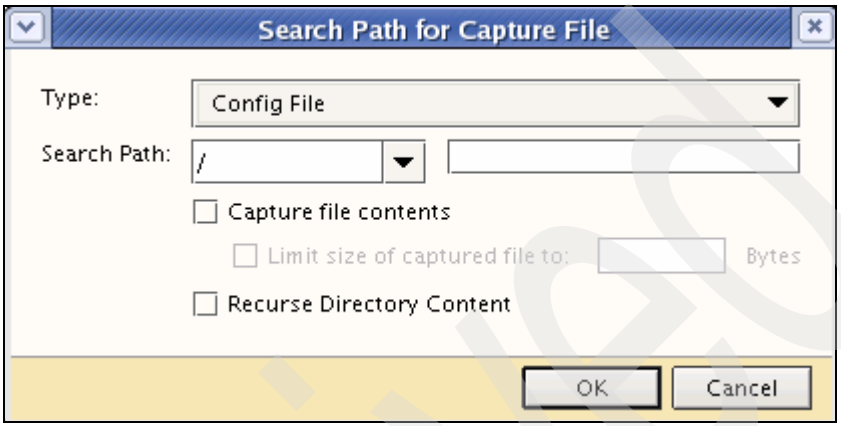

Figure 4-5 Search Path for Capture File

        ii. Select a type.
        iii. Type a search path.
        iv. Select the Capture file contents check box to capture the contents of the file.
        v. Select the Recurse Directory Content check box to recurse the directory structure.
        vi. Click **OK**. A new file appears on the Config Files list.
    - To edit a configuration file:
        i. Select a configuration file and click **Edit**. The Edit Capture File window opens.
        ii. Edit the information related to the configuration file. Click **OK**.
        iii. File appears updated on the Config files list.
    - To delete a configuration file:
        i. Select a configuration file and click **Remove**.
        ii. The Config files list appears updated.
14. Click **OK**. A new custom server appears at the bottom of the Custom Servers list.

4.6.2 Editing a custom server template

To edit a custom server, complete the following steps from the Product Console:

1. Select **Discovery** → **Custom Servers**. The Custom Servers page opens.
2. Select a custom server, and click **Edit**. The Custom Server Details window opens.
3. Click **General Info & Criteria** tab.
4. Edit general server and identifying criteria information.
5. Click **Config Files** tab.
6. Edit configuration file information.
7. Click **OK**. Custom server appears updated on the list.

4.6.3 Copying a custom server template

To copy a custom server, complete the following steps from the Product Console:

1. Select **Discovery** → **Custom Servers**. The Custom Servers page opens.
2. Select a custom server, and click **Copy**. The Set Name window opens.
3. Type a name.
4. Click **OK**. The copy of the selected custom server appears at the bottom of the Custom Servers list.

4.6.4 Deleting a custom server template

To delete a custom server, complete the following steps from the TADDM Product Console:

1. Select **Discovery** → **Custom Servers**. The Custom Servers page opens.
2. Select a custom server, and click **Delete**. The Confirm Deletion window opens.
3. Click **Yes**. The Custom Servers list appears updated.

4.6.5 Changing the order of the custom server templates

To change the order of the custom server templates, complete the following steps from the Product Console:

1. Select **Discovery** → **Custom Servers**. The Custom Servers page opens.
2. Select a custom server.
3. Click **Move Up** or **Move Down** to move the custom server until the position you want.
4. Click **Save**.

4.7 Application templates

You can use *application templates* to associate business applications with custom servers. With application templates, you can create new business applications out of custom servers, or update existing business application servers to include custom servers.

4.7.1 Adding an application template

To add an application template, complete the following steps from the Product Console:

1. Select **Discovery** → **Application Templates**. The Application Templates page opens.
2. Click **Add**. The Create Application Template window opens.
3. Click **General** tab.
4. Type a name.
5. Optionally fill in the Admin Info field.
6. Click **Criteria** tab. See Figure 4-6 on page 87.

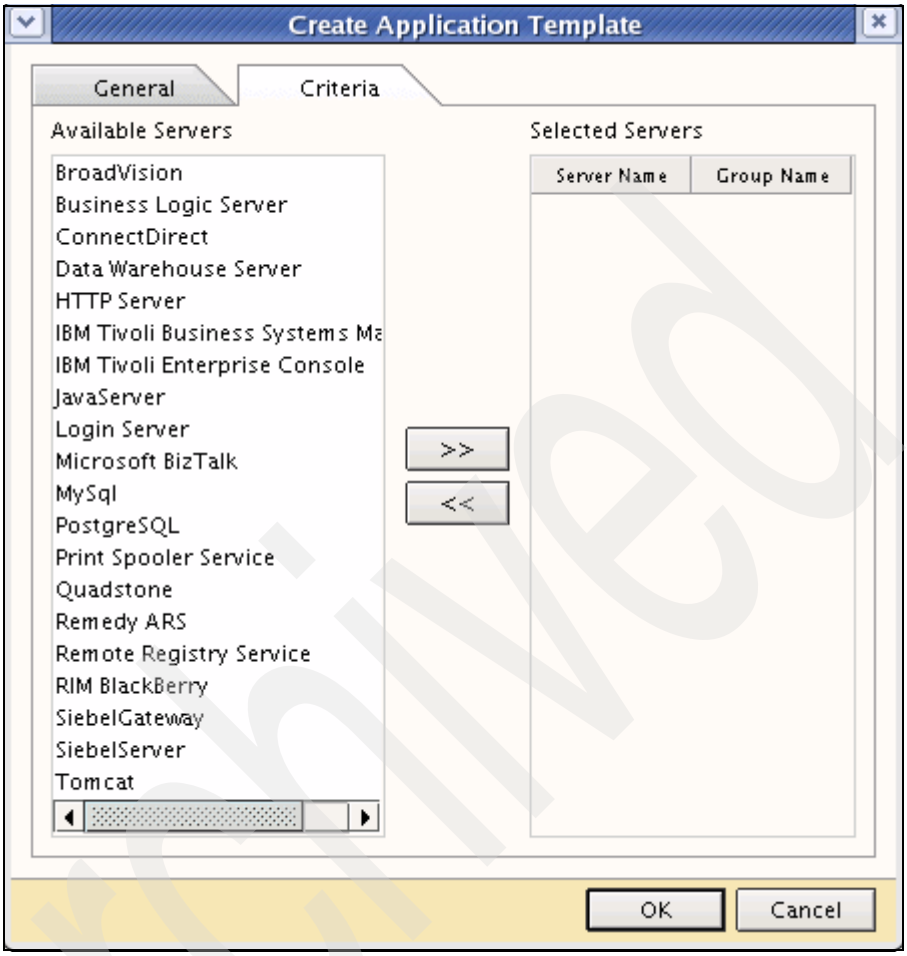

Figure 4-6 Create Application Template

7. To add a custom server, select a custom server on the Available Servers list and click the double right-angle button (**>>**).

Notes:

- Only custom servers that are defined by custom server templates can be added to the application template.
- Custom server templates that are used to ignore applications are not in the list.

8. To remove a custom server, select a custom server on the Selected Servers list and click the double left-angle button (**<<**).
9. Click **OK**. A new application template appears on the Application Templates list.

4.7.2 Editing an application template

To edit an application template, complete the following steps from the Product Console:

1. Select **Discovery** → **Application Templates**. The Application Templates page opens.
2. Select an application template, and click **Edit**. The Edit Application Template window opens.
3. Click the **General** tab.
4. Edit the admin information.
5. Click the **Criteria** tab.
6. Edit criteria information.
7. Click **OK**. Application template appears updated on the list.

4.7.3 Deleting an application template

To delete an application template, complete the following steps from the Product Console:

1. Select **Discovery** → **Application Templates**. The Application Templates page opens.
2. Select an application template, and click **Delete**. The Confirm Deletion window opens.
3. Click **Yes**. The Application Templates list appears updated.

4.8 Application descriptors

You can use *application descriptors* to associate business applications with their components. An application descriptor is an XML file that maps modules or servers (containers) to applications. When the application descriptors are discovered, they are used to automatically associate components to a business application.

The two types of application descriptors are:

- Base application descriptor
- Component application descriptor

You must assign a unique application name in both the base application descriptor and the component application descriptor. This unique name is used to correlate all discovered application descriptors for a specific application.

4.8.1 Base application descriptor

The base application descriptor contains general information about the application, such as the version, contact, and other information.

Because the base application descriptor contains general information, it is not required in order to discover an application.

Note: Business applications are created automatically even without a base application descriptor file as long as a component application descriptor file is provided and contains an app-instance-name tag in the file. The name that is used for the business application is the app-instance-name tag from the component application descriptor file.

Only a single base application descriptor is required for each application. In cases when more than one descriptor is used, the system uses the one with the most recent time stamp.

The base application descriptor can be deployed to the descriptor directory of any one of the components of the application.

Table 4-2 on page 90 describes the structure of the base application descriptor.

Table 4-2 Base application descriptor elements and attributes

Element	Description and attributes	
base-app-descriptor	The root element for the base application descriptor	
app-instance	The element for the application instance information	
	(Required) name	The name of the application instance
	version	The application version
	description	A description of the application instance
	url	The URL pointing to the application
	contact	A contact name or other information for the application (This is not imported into TADDM.)
app-definition	The element for the application definition information.	
	name (Required)	The name of the application definition
	description	A description of the application instance

Example 4-4 is an XML file sample of the base application descriptor.

Example 4-4 Sample base application descriptor file

```
<base-app-descriptor>
   <app-instance
         name="Order Management - Staging"
         version="1.5.1"
         description="Order Entry application - staging"
         url="http://orderentry.lab.ibm.com"
         contact="John Public" />
   <app-definition
      name="Order Management"
      description="Order Entry & Tracking application" />
</base-app-descriptor>
```

4.8.2 Component application descriptor

The component application descriptor contains information about a specific application component (server) or module deployed within a server, along with information about the participation of the component within the application.

Components can include database servers, J2EE™ servers, and a module on a server such as Web applications, Enterprise Archives, JSP™ pages, and so on. Each module can have a separate descriptor, or you can include multiple modules (for a single server) in a single descriptor.

The component application descriptor is deployed to the descriptor directory of each of the components (servers) of the application.

Table 4-3 on page 92 describes the structure of the component application descriptor.

Table 4-3 Component application descriptor elements and attributes

Component descriptor element	Description and attributes	
component-app-descriptor	The root element for the component application descriptor	
	app-instance-name	(Required)The name of the application instance
component-descriptor	(Required) The element for the component information	
	type	(Required) A component descriptor can apply to a server in its entirety, or to individual modules within the server. The type attribute specifies this relationship, and can assume either of the following values: ► module ► server
	name	(Required) The name of the component
	functional-group	(Required) The functional group that the component occupies within the application. Functional groups allow for the grouping of similar components within the application. They are used to compare applications to each other.
	marker-module	(Optional) A special type of module definition for J2EE domains. When a module is indicated as a marker module, J2EE-managed servers within the domain that include the marker module are treated as having all of its modules included in the application. You can specify the following values as the marker module: ► true ► false

Example 4-5 on page 93 is a sample XML file of the component application descriptor.

Example 4-5 Example component application descriptor file

```
<component-app-descriptor
   app-instance-name="Order Management-Staging">
   <component-descriptor
      type="module"
      name="/opt/apache13/htdocs/ordermgt/"
      functional-group="Web Tier"
      marker-module="false" />
</component-app-descriptor>
```

4.8.3 Application descriptor locations

The location of the directory containing application descriptors for a particular server is unique to that server, and is determined based on the following criteria.

TADDM looks for a subdirectory named `appdescriptors` in a directory that is determined for each server based on the following order of priority:

1. If set, the `COLL_APP_DESC_DIR` environment variable
2. If specified, the `COLL_APP_DESC_DIR` command line argument
3. The `COLL_APP_DESC_DIR` environment variable and the `COLL_APP_DESC_DIR` command line argument must be part of the server startup environment and argument. If you cannot customize the environment variables or command line of the server, TADDM looks for application descriptors in the default application descriptor directory, as shown in Table 4-4.

Table 4-4 Default application descriptor locations

Server	Default directory and supported modules
WebSphere	► For WAS 5.1: `<WebSphere_home_dir>/appdescriptors` ► For WAS 6.x: `<WebSphere_home_dir>/profiles/<profile_name>/appdescriptors` In this location, <profile_name> is what you configured for your WAS cell. An example is: `/opt/IBM/WebSphere/profiles/default/appdescriptors` Supported modules: J2EE applications, Web, EJB™, and connector modules

Server	Default directory and supported modules
WebLogic	<WebLogic_home_dir>/appdescriptors Supported modules: J2EE applications, Web, EJB, and connector modules
JBoss®	JBoss_home_dir>/appdescriptors Supported modules: J2EE applications, Web, EJB, and connector modules
iPlanet	<iPlanet_home_dir>/appdescriptors Supported modules: Servlets, JSP pages
Apache	<Apache_home_dir>/appdescriptors
Microsoft IIS	<IIS_home_dir>/appdescriptors Supported modules: Virtual hosts
Oracle	<Oracle_home_dir>/instance_name/appdescriptors Supported modules: Users **Important:** You have to create the instance_name for this location.
Sybase/Sybase IQ™	<Sybase_home_dir>/appdescriptors Supported modules: Databases
SQLServer	<SQLServer_home_dir>/appdescriptors Supported modules: Databases
DB2	<DB2_home_dir>/instance_name/appdescriptors
Domino® Server	<Domino_server_home_dir>/appdescriptors
Microsoft Exchange Server	<Domino_server_home_dir>/appdescriptors Microsoft Exchange Server For Microsoft Exchange Server 2003: <exchange_server_home_dir>/appdescriptors Supported modules: Exchange Servers, Exchange Protocol Virtual Servers

Server	Default directory and supported modules
Custom Server	User-supplied through template definition Supported modules: User-supplied through template definition

Veritas Cluster	`VS_home_dir/appdescriptors`
	
	Default locations are:
	► Windows systems
	`C:\Program Files\Veritas\Cluster Server\appdescriptors`
	► UNIX systems
	`/opt/VRTSvcs/appdescriptors`

In the case of managed servers, such as J2EE servers, which are managed by the J2EE domain, the location of the application descriptor directory is at the level of the Admin Server/Domain Manager. The contents specified in that directory are used as the superset of all possible mappings for all managed servers. For each managed server (depending on which modules are discovered as deployed), the application descriptor is processed for inclusion of those modules in the application.

4.9 Security

TADDM server can be configured to use three methods for security:

- File authentication
- LDAP
- WebSphere federated reporistory

This section explains the security configuration for these methods and security mechanism in the existence of a TADDM Enterprise Domain Server.

4.9.1 File authentication

Out of the box, TADDM has file-based authentication. User information is encrypted and saved in a file. For file authentication, user configuration tasks can be performed from Product Console as following:

Create a user

When the file-based registry is used for user management, to create a user, complete the following steps from the Domain Manager:

1. Select **Administration** → **Users**. The Users page opens.
2. Click **Create User**. The Create User window opens. See Figure 4-7 on page 97.

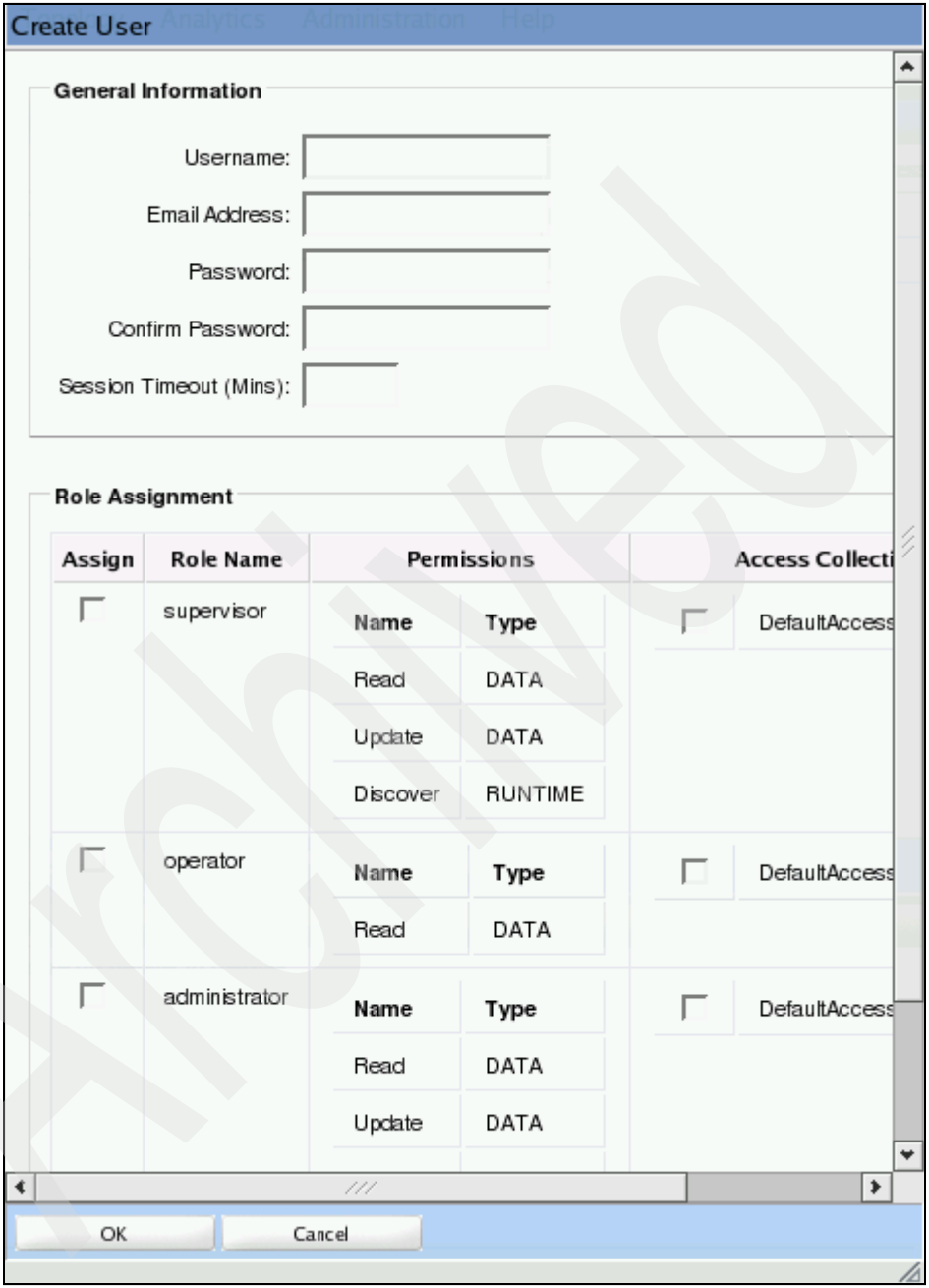

Figure 4-7 Create User window

3. Type the information for the following fields:
   - Username
   - Email Address
   - Password (twice for confirmation)
   - Session Timeout (in minutes)

   Note: For an unlimited session timeout for the Product Console, the session timeout value is -1.

4. Assign roles to the new user. For each role, perform the following steps:
   a. Select the check box of the role.

      Note: TADDM has three default roles:
      - Operator, with read permission
      - Supervisor, with read, update, and discover permissions
      - Administrator, with read, update, discover and admin permissions

   b. Select the check boxes of the access collections for the role to specify the scope of the role.
5. Click **OK**. A new user appears on the Users list.

Edit a user

When the file-based registry is used for user management, to edit a user complete the following steps from the Domain Manager:

1. Select **Administration** → **Users**. The Users page opens.
2. Select a user, and click **Edit**. The Edit window opens.
3. To change the user details:
   a. Edit the user details information and click **Change**.
   b. Click **OK**. User appears updated on the list.
4. To change the password:
   a. Edit the password information and click **Change Password**.
   b. Click **OK**. User appears updated on the list.
5. To change the role assignments:
   a. Select or deselect the check box of a role.
   b. Select or deselect the check boxes of the access collections for the role to specify the scope of the role.
   c. Click **OK**. User appears updated on the list

Delete a user

When the file-based registry is used for user management, to delete a user complete the following steps from the Domain Manager:

1. Select **Administration** → **Users**. The Users page opens.
2. Select a user, and click **Delete**. The Users list appears updated.

4.9.2 Configuring for LDAP

TADDM server can be configured to use an LDAP registry for user authentication. By configuring LDAP, you can use the users that are defined in LDAP registry.

For LDAP configuration, TADDM server requires a user that is defined in LDAP and that is assigned the administrator role in TADDM. TADDM comes with a user named administrator with the role administrator. Before LDAP configuration, perform either one of the following steps:

- Create a user named administrator in the LDAP registry.
- Create a user in the LDAP registry and with the same name in the TADDM server, and then assign the administrator role to the user for all access collections.

To configure TADDM server for LDAP, complete the following steps:

1. Specify the user management module used by this TADDM Server. Possible values are:
   - file

     This value is for a file-based user registry. (This is the default value.)
   - ldap

     This value is for an LDAP user registry.
   - vmm

     This value is for a user registry that uses the federated repositories of WebSphere Application Server.

   For example, in the `$COLLATION_HOME/etc/collation.properties` file:

   ```
   com.collation.security.usermanagementmodule=ldap
   ```
2. Enable the LDAP authentication attribute in the `$COLLATION_HOME/etc/collation.properties` file. For example:

   ```
   com.collation.security.auth.ldapAuthenticationEnabled=true
   ```

3. Set the LDAP host name and port in the `$COLLATION_HOME/etc/collation.properties` file. For example:

   ```
   com.collation.security.auth.ldapHostName=ldap.ibm.com
   com.collation.security.auth.ldapPortNumber=389
   ```

4. Specify the starting point for an LDAP search in the `$COLLATION_HOME/etc/collation.properties` file. For example:

   ```
   com.collation.security.auth.ldapBaseDN=ou=People,dc=ibm,dc=com
   ```

5. Specify the user ID used to authenticate to LDAP if simple authentication is used in the `$COLLATION_HOME/etc/collation.properties` file. For example:

   ```
   com.collation.security.auth.ldapBindDN=uid=ruser,dc=ibm,dc=com
   ```

6. Specify the user password used to authenticate to LDAP if simple authentication is used in the `$COLLATION_HOME/etc/collation.properties` file. For example:

   ```
   com.collation.security.auth.ldapBindPassword=ruser
   ```

7. Specify the name of the class used to represent users in LDAP in the `$COLLATION_HOME/etc/collation.properties` file. For example:

   ```
   com.collation.security.auth.ldapUserObjectClass=person
   ```

8. Specify the name of the attribute used for naming a person in LDAP in the `$COLLATION_HOME/etc/collation.properties` file. For example:

   ```
   com.collation.security.auth.ldapUIDNamingAttribute=cn
   ```

9. Specify the name of the class used to represent user groups in LDAP in the `$COLLATION_HOME/etc/collation.properties` file. For example:

   ```
   com.collation.security.auth.ldapGroupObjectClass=groupofuniquenames
   ```

10. Specify the name of the attribute used for naming a group in LDAP in the `$COLLATION_HOME/etc/collation.properties` file. For example:

    ```
    com.collation.security.auth.ldapGroupNamingAttribute=cn
    ```

11. Specify the name of the attribute used to contain the members of a group in the `$COLLATION_HOME/etc/collation.properties` file. For example:

    ```
    com.collation.security.auth.ldapGroupMemberAttribute=uniquemember
    ```

12. After making changes to the `$COLLATION_HOME/etc/collation.properties` file, stop and start TADDM so that the changes become active.

4.9.3 Configuring for WebSphere federated repositories

TADDM server can be configured to use a WebSphere federated repository for user authentication. By configuring WebSphere federated repository, you can use the users that are defined, and you can enable single sign-on (SSO) with

other Tivoli products such as Change and Configuration Management Database (CCMDB).

For WebSphere federated repository configuration, the TADDM server requires a user that is defined in the federated repository registry which is assigned the administrator role in TADDM. TADDM comes with a user named administrator with the role administrator. Before WebSphere federated repository configuration perform either one of the following steps:

- Create a user named administrator in the federated repository registry.
- Create a user in federated repository registry and in TADDM server with the same name, and assign administrator role to user for all access collections.

To configure TADDM server for WebSphere federated repository, complete the following steps:

1. Stop the TADDM Server.
2. Specify the user management module used by this TADDM Server. Possible values are:
   - file

     This value is for a file-based user registry. (This is the default value.)
   - ldap

     This value is for an LDAP user registry.
   - vmm

     This value is for a user registry that uses the federated repositories of WebSphere Application Server.

   For example, in the `$COLLATION_HOME/etc/collation.properties` file:

   ```
   com.collation.security.usermanagementmodule=vmm
   ```
3. Specify the WebSphere host name and port in the `$COLLATION_HOME/etc/collation.properties` file. For example:

   ```
   com.collation.security.auth.websphereHost=host1.austin.ibm.com
   com.collation.security.auth.webspherePort=9809
   ```
4. Specify the WebSphere administrator user name and password in the `$COLLATION_HOME/etc/collation.properties` file. For example:

   ```
   com.collation.security.auth.VMMAdminUsername=administrator
   com.collation.security.auth.VMMAdminPassword=password
   ```
5. Modify the authentication services `$COLLATION_HOME/etc/ibmessclientauthncfg.properties` file as follows:

   ```
   # This is the URL for the Authentication Service
   authnServiceURL=http://host1.austin.ibm.com:9080/TokenService/services/Trust
   ```

6. Copy two files into the JRE™ as follows:

```
# CRC - Linux jre shown, other jres should work as well

cp dist/lib/websphere/6.1/orb.properties to
dist/external/jdk-1.5.0-Linux-i686/jre/lib

cp dist/lib/websphere/6.1/iwsorbutil.jar to
dist/external/jdk-1.5.0-Linux-i686/jre/lib/ext
```

7. Specify the WebSphere host name and port in the `$COLLATION_HOME/etc/sas.client.props` file. For example:

```
com.ibm.CORBA.securityServerHost=host1.austin.ibm.com
com.ibm.CORBA.securityServerPort=9809
```

8. Specify the WebSphere administrator user name and password in the `$COLLATION_HOME/etc/sas.client.props` file. For example:

```
# RMI/IIOP user identity
com.ibm.CORBA.loginUserid=administrator
com.ibm.CORBA.loginPassword=password
```

9. Encrypt login password in the `$COLLATION_HOME/etc/sas.client.props` file as follows:

   a. Copy `$COLLATION_HOME/etc/sas.client.props` to your WebSphere machine. For example, to `C:\temp\sas.client.props`.

   b. Encrypt the password by using the following command, depending on the operating system you have installed WebSphere.

      - For Linux, Solaris, AIX, and Linux on System z operating systems:

        `PropFilePasswordEncoder.sh`

      - For Windows operating systems:

        `PropFilePasswordEncoder.bat`

      For example:

```
C:\WebSphere\profiles\AppSrv01\bin\PropFilePasswordEncoder
C:\temp\sas.client.props com.ibm.CORBA.loginPassword
```

10. Copy the `sas.client.props` file back to the TADDM Server in the `$COLLATION_HOME/etc` directory.

11. Start the TADDM Server.

For more information, refer to *Tivoli Application Dependency Discovery Manager Administrator's Guide*:

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/cmdb_admin.pdf

4.9.4 Enterprise Domain Server

When you use a TADDM Enterprise Domain Server, the authentication and authorization for the domains are delegated to the enterprise server.

The following list summarizes the security information in a TADDM Enterprise Domain Server:

- For TADDM to function properly, the TADDM Enterprise Domain Server must be running. A TADDM domain delegates security operations to the TADDM Enterprise Domain Server, and this delegation is updated every 2.5 minutes. If 5 minutes pass and this delegation is not updated, the TADDM domain no longer delegates security operations and proceeds as if no TADDM Enterprise Domain Server is present.
- Roles, permissions, and access collections that are stored in the TADDM Server are synchronized from the domain to the TADDM Enterprise Domain Server.
- Users and user to role mappings are not synchronized to the TADDM Enterprise Domain Server.
- If you are using the TADDM file-based registry and a TADDM domain is added to a TADDM Enterprise Domain Server, you must re-create in the TADDM Enterprise Domain Server any users that already exist in a domain, including assigned roles and access that is granted to access collections.
- If you are using a Lightweight Directory Access Protocol (LDAP) or WebSphere Federated Repositories user registry, you must add to the TADDM Enterprise Domain Server the authorization (user to role mappings) for any users that access TADDM.
- Roles and permissions that you created for the domain can be used by the TADDM Enterprise Domain Server after these objects are synchronized from the domain to the Enterprise Domain Server.

5

Discovery

IBM Tivoli Application Dependency Discovery Manager (TADDM) discoveries collect configuration item information of your IT infrastructure, including deployed software components, physical servers, network devices, virtual LAN and host data that are used in a runtime environment. This chapter discusses the most important aspects about TADDM discovery process, including discovery of business applications and business services.

This chapter contains the following topics:

- "Running discoveries" on page 106
- "Creating and managing custom servers" on page 113
- "Using discovery profiles" on page 120
- "Access collections" on page 123
- "Discovering business applications and business services" on page 126
- "The TADDM Enterprise Domain Manager" on page 127

5.1 Running discoveries

TADDM discoveries collect configuration item information of your IT infrastructure, including deployed software components, physical servers, network devices, virtual LAN and host data used in a runtime environment. To optimize the breadth and depth of information that TADDM gathers, some configuration is required within TADDM and in your environment.

For discoveries to run in your environment, you have to provide TADDM with three types of information:

- Scope
- Access credentials (not required for credential-less discoveries)
- Schedule

In the Product Console, either the Discovery tab or the Discovery menu gives you access to the items you have to configure for discoveries.

5.1.1 Defining a scope

Scopes use IP addresses to tell TADDM where to begin discovering the environment. A scope is composed of one or more scope sets.

Creating and managing discovery scopes was discussed in 4.2, “Discovery scopes” on page 69. Refer to that section for more information.

5.1.2 Configuring the access list

With a few exceptions, discoveries require some level of access to acquire the detailed information users want for a complete understanding of the environment. In addition to providing credentials, sometimes you will have to prepare your environment to enable TADDM discoveries to run. For example, the UNIX utility `lsof` (LiSt Open Files) is one of the built-in tools that TADDM uses to perform agentless discoveries. Therefore, `lsof` must be installed on target UNIX servers before you can get detailed discovery results from those servers.

Credentials for TADDM are entered through the GUI. Under the Discovery tab, click **Access List**. By default, several examples of entries are available when you first open the Access List window. You can edit these entries with real data, delete them, or ignore them and move them to the bottom of the list.

Refer to 4.3, “Access list” on page 73 for more details on this topic.

5.1.3 Adding credentials

You add credentials to the Access List by clicking the **Add** button.

Open the Access Details window by clicking **Add** and begin a new entry by opening the pull-down menu to select the **Component Type**. Component types are general categories of target types. It is important to accurately select the type of device for which you are making an entry. The Access Details window is configured to collect appropriate data and TADDM knows how to apply the data for each component type.

5.1.4 Discovery schedules

You can schedule discoveries to ensure that information presented in the Product Console is always current and accurate. In most cases, you should partition your environment into operational groups and perform discoveries on these subsets of your organization. This approach reduces the time to complete a particular discovery, and takes into account that different sections of your environment change at different rates.

If you create a schedule to run a discovery, it binds the current scope to that schedule. Later, if you want to add a new entry to the scope, you must delete the schedule and create a new one.

You can schedule a discovery to perform the following tasks:

- Identify operational groups within your environment. Different sections of your environment are likely to have different rates of change. By identifying operational groups on your network by IP addresses, IP address ranges, and subnets, you can schedule partitions of your infrastructure to have different discovery schedules.
- Check the discovery history to determine how long it typically takes to complete different types of discoveries in your environment. Discovery schedules cannot overlap. The first discovery must complete before a new discovery can begin. If a discovery is scheduled to start before an existing discovery finishes, the new discovery does not start, and an error is logged. Check the discovery history to estimate the typical completion time for different discoveries, so that you can prevent potential schedule overlaps.
- Schedule discoveries based on the operational groups you identified.

Configure most of your scheduled discoveries to refresh a subset of your topology. For example, depending on the size of your environment and your operational requirements, you can schedule a full discovery once every 24 hours, or complete a discovery of the application tier once every six hours.

When you create a discovery schedule, you specify the start time and the frequency of discoveries. You can also define the scope of a particular discovery by selecting the scope elements (subnets, IP addresses, or ranges), components, or views to include in the discovery.

Creating a discovery schedule

To create a new discovery schedule, complete the following steps from the Product Console:

1. Select **Discovery** → **Schedule**.
2. Click **Add** to define a new discovery schedule. The Discovery Schedule window (Figure 5-1) opens. The date and time shown in the window reflect the date and time on the TADDM server.

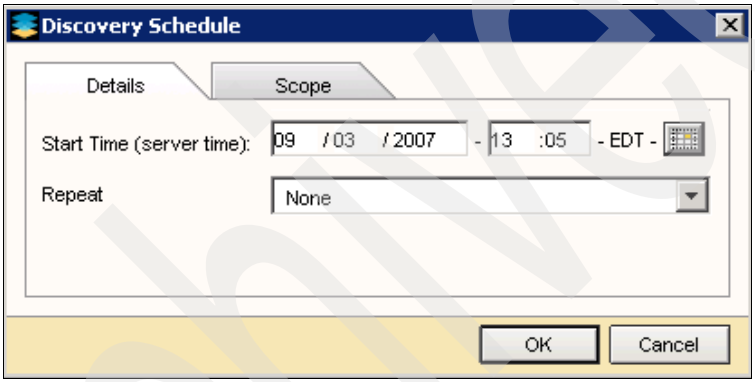

Figure 5-1 Discovery Schedule

3. On the Details tab (Figure 5-2 on page 109), specify the name of the schedule, the date and the time you want discovery to begin, and the repeat frequency (None, Hourly, Daily, Weekly, and Monthly.

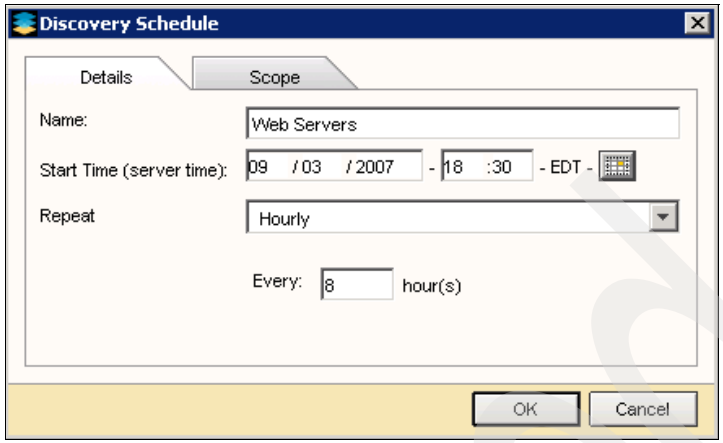

Figure 5-2 Details tab

4. For the schedule scope, TADDM gives you the option to select **Scope Elements**, that is scope sets you have already created, or select **Components**, such as business applications or business services. See Figure 5-3.

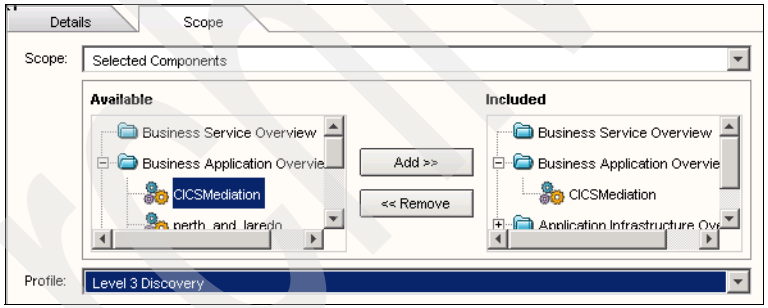

Figure 5-3 Scope tab

5. Select the **Discovery Profile** you want TADDM to use for this discovery. You have the choice of any of the Discovery Profiles you created as well as the default profiles.
6. Click **OK**.

Viewing a discovery schedule

To view a discovery schedule, complete the following steps from the Product Console:

1. Select **Discovery** → **Schedule**.
2. In the **Schedule** window, select the schedule you want to view.
3. Click **Details** to open the Schedule Details window.

For more information, such as deleting or editing a discovery schedule, refer to 7.3, “Discovery schedules” on page 154.

For more information about Discovery Schedule refer to *IBM Tivoli Application Dependency Discovery Manager Capabilities and Best Practices*, SG24-7519

5.1.5 Running a basic discovery

After you set up an initial scope for the discovery and establish an access list for your computing systems, you are ready to run a basic discovery.

To run a discovery, complete the following steps from the Product Console:

1. From the menu bar, click **Discovery Overview**. The Overview pane is displayed.
2. To start the discovery, click **Run Discovery**. The Run Discovery window opens.
3. In the Run Discovery window, complete one of the following steps:
   - To base the discovery on selected scopes that you configured, select **Selected Scope Elements** from the Scope menu, and then select from the tree the scopes to include in the discovery.
   - To base the discovery on selected components, select **Selected Components** from the Scope menu, and then select the components to include in the discovery.
4. From the Profile list, select the discovery profile to use during the discovery.
5. Click **OK** to run the discovery. See Figure 5-4 on page 111.

Note: Optionally, after running a discovery, you may save a version of the discovery for future use and for comparison to other discoveries.

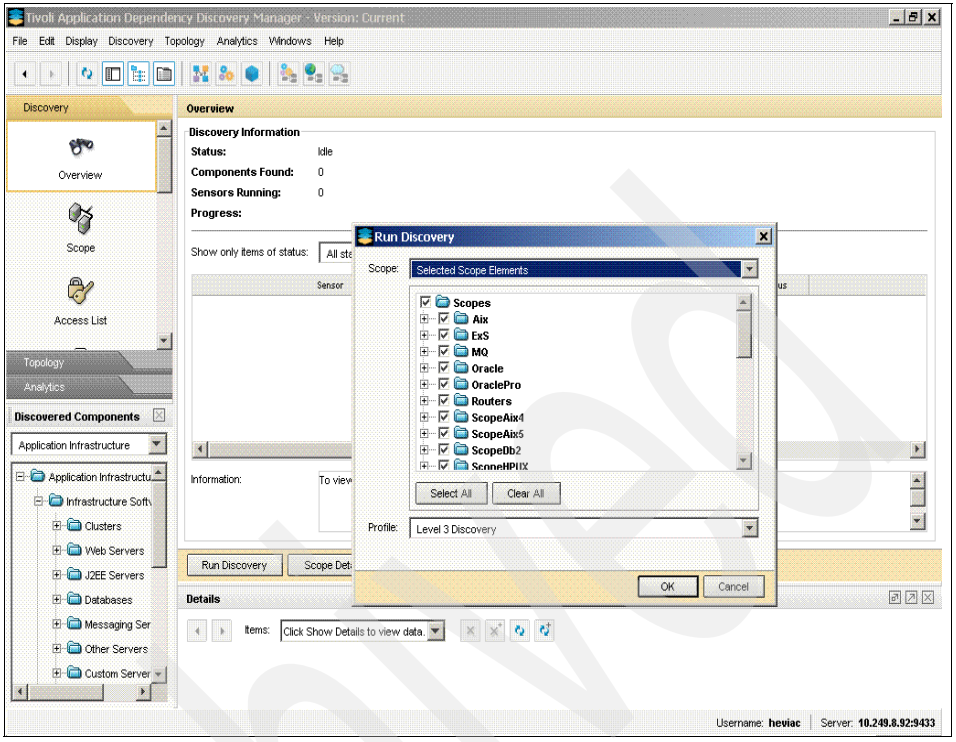

Figure 5-4 Running a discovery

Note: Discoveries cannot overlap. If you have to run a discovery after one is running, then, the first discovery must complete before a new discovery can begin.

5.1.6 Running a discovery from the command line with api.sh

You can also run TADDM discoveries from the command line. Refer to 7.10.2, “Discover command” on page 176 for examples of doing that.

5.1.7 Viewing the discovery history

Each time that a discovery is run, the TADDM Product Console updates the discovery activity and error information that is displayed in the Overview pane. You can view the discovery history, including the associated activity and error information, in the History pane.

To view a discovery history, complete the following steps from the Product Console or Domain Manager:

1. In the Functions pane, click **Discovery History**. The History panel is displayed, as shown in Figure 5-5.

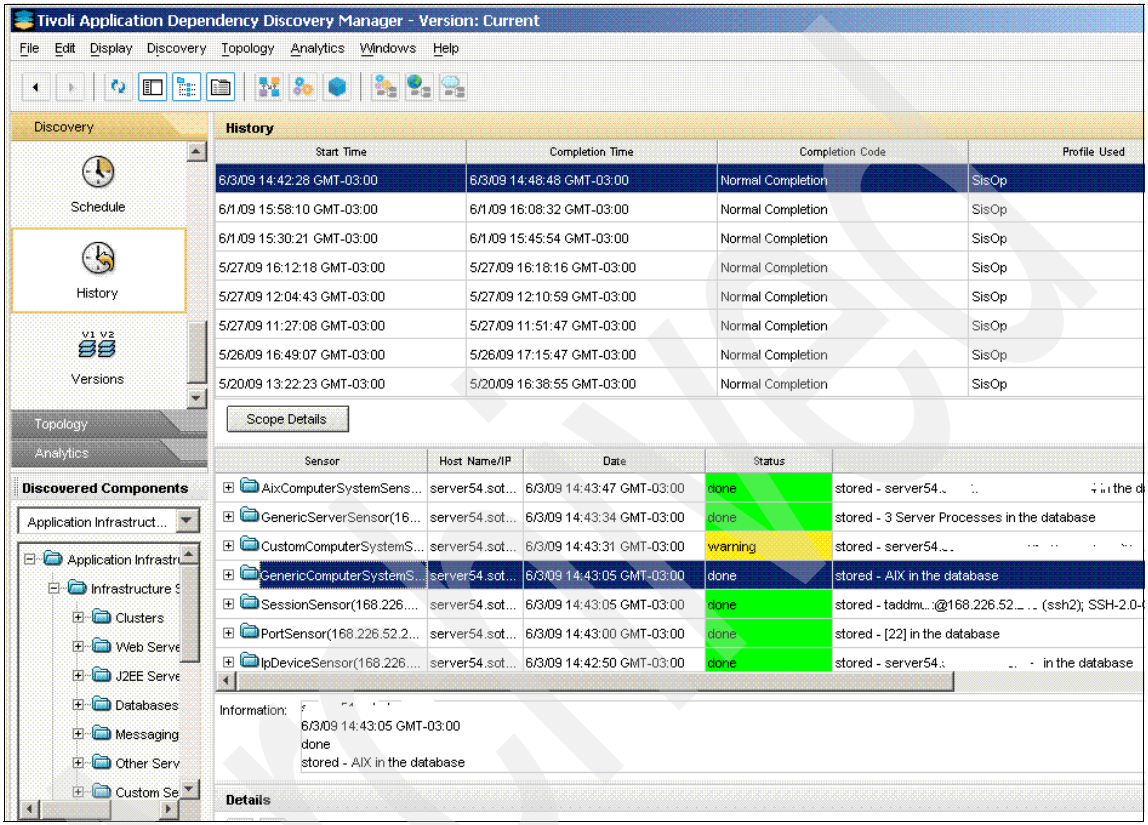

Figure 5-5 History pane

2. To display information about a discovery, select an entry in the table. A second table of data is displayed. This table provides a list of sensors and the host name, IP address, date, status, and description for each sensor.
3. To display the scopes that are included in the discovery, click **Scope Details**. The Scope List window is displayed.
4. To close the Scope List window, click **Cancel**.

5.2 Creating and managing custom servers

You can create custom servers to discover and categorize servers that are not, by default, supported by TADDM. This is an advanced technique for configuring TADDM to discover servers that it does not know about by default.

Your infrastructure might contain software applications and server types, such as custom Java servers, that are not automatically categorized by TADDM. Any server process with a TCP listening port that is not recognized is categorized into an Unknown Server category. Unknown servers are not displayed in the topology and cannot take advantage of most of the functions.

You do, however, get basic information such as the name and runtime data about the unknown server. You can define a custom server to create a template that sets up the membership rules for the custom server.

During a discovery, any unknown server is automatically categorized as a custom server of this type if the runtime information matches the criteria you defined in the template. Any configuration files used by the custom server are also automatically captured if specified in the templates. Custom servers are displayed in the topology, and you can view details about them. Although these details are not as complete as those provided for supported servers, defining custom servers allows all components in your infrastructure to participate in the topology and comparisons. You can manage custom servers in the Custom Servers window.

5.2.1 Identifying unknown server patterns

Before adding a server, run a basic discovery to check for unknown servers.

You can run a report on unknown servers to help you identify patterns to use in the custom server template.

To identify patterns in unknown servers, complete the following steps from the Product Console:

1. In the Functions pane, select **Analytics** → **Inventory**. The Inventory pane is displayed in the workspace.
2. In the Inventory pane, select **Unknown Servers**.
3. To run a report on the unknown servers, click **Run Report**. A message window is displayed.
4. Click **OK**. The Inventory Results pane is displayed.

You should be able to identify a pattern in the configuration of the unknown server, such as the program name, arguments, environment, and port. Use this pattern to create the identifying criteria for the customer server template.

To create a custom server from an unknown server, select an unknown server, and click **Create Custom server** in the Inventory Results pane, as shown in Figure 5-6.

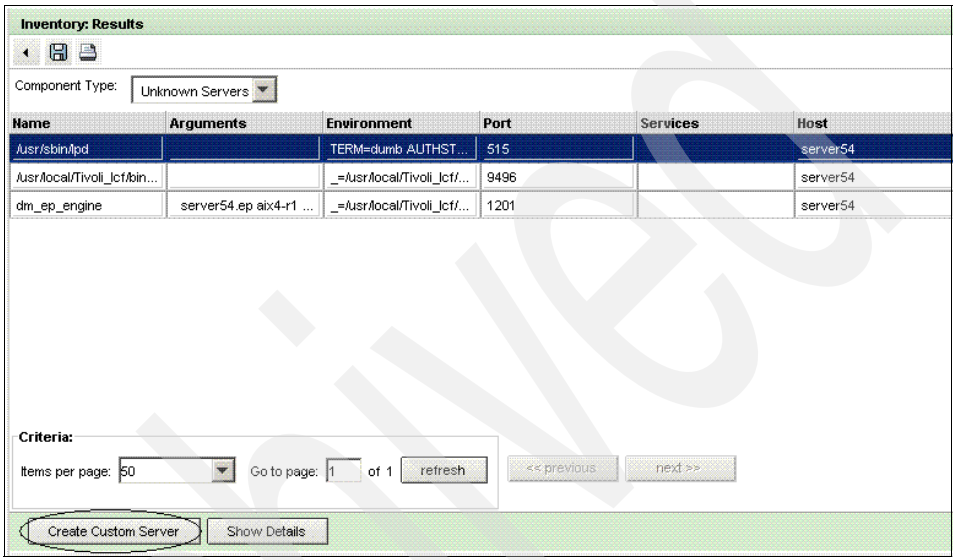

Figure 5-6 Inventory Results

5.2.2 Adding custom servers

A custom server template contains descriptive criteria that is used to assign unknown server processes to the custom server. You specify this criteria when defining the template for the custom server. The following information, associated with running processes, is parsed to match the process to a particular custom server:

- Program name: Name of the executable program
- Window Service name: Name of a window service
- Argument: Arguments passed to the program
- Environment: Environment variables set for the program
- Port: TCP port number on which the process is listening

The custom server general information and criteria details include the name, the type of server, and identifying criteria for the custom server. To view details

about that unknown server, double-click an unknown server in the Topology, and click the **Runtime** tab. You may then use this information to create a search criteria for a custom server using the General Information and the Criteria tab of the Custom Server Details window.

To add a custom server, complete the following steps from the Product Console:

1. In the Functions pane, select **Discovery** → **Custom Servers**. The Custom Servers pane is displayed in the workspace, as shown in Figure 5-7.

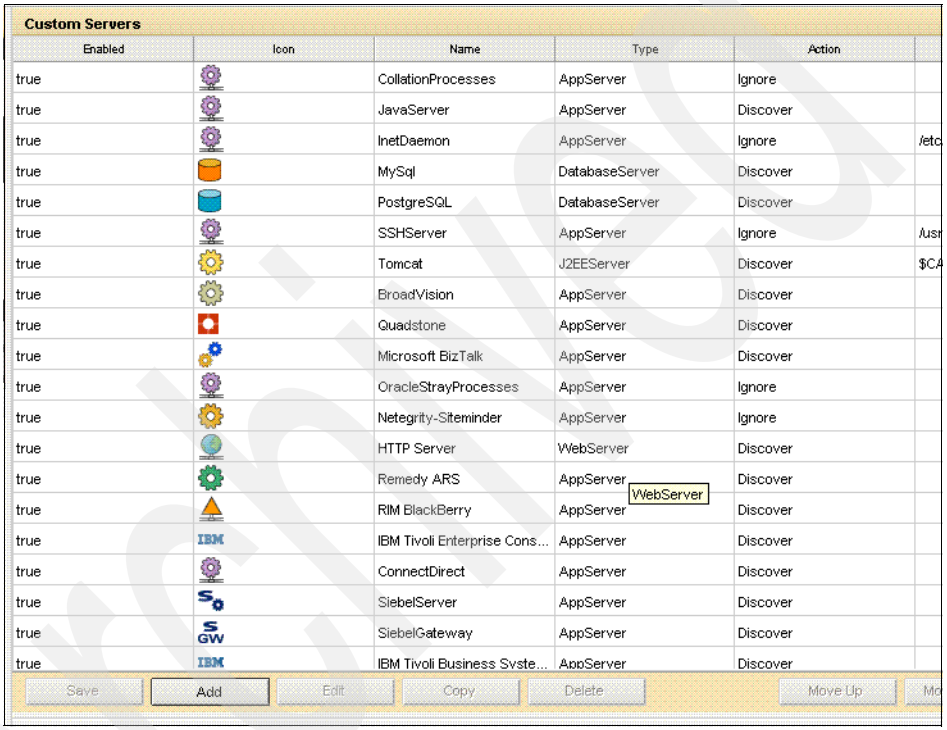

Figure 5-7 Custom Servers

2. Click **Add**. The Custom Server Details notebook is displayed. See Figure 5-8 on page 116.

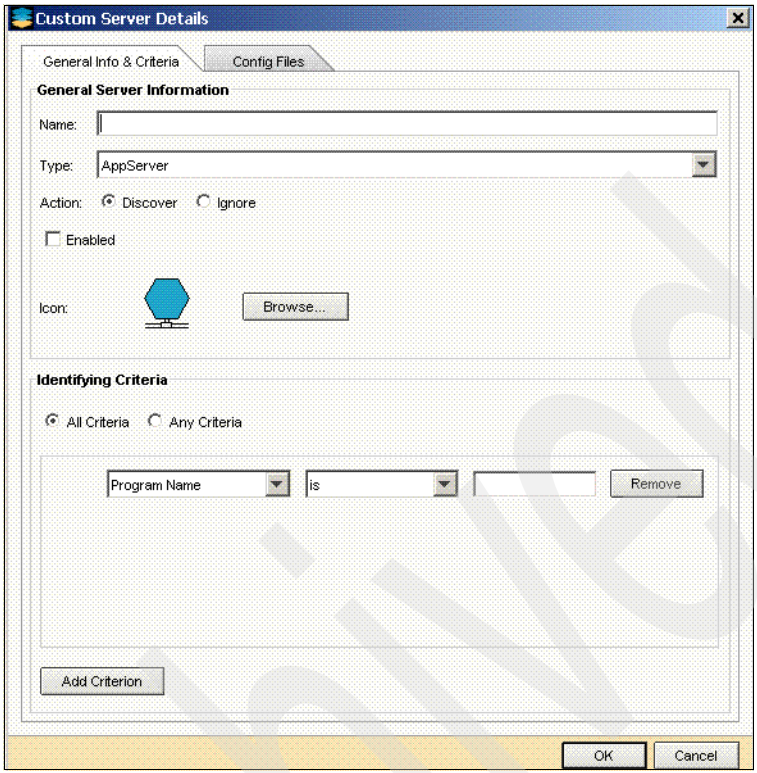

Figure 5-8 Custom Server Details

3. In the Name field, type the name of the custom server.
4. From the Type list, select the type of custom server that you are adding.
5. Under Action, select either of the following options:
   - **Discover**, to discover all instances of the server
   - **Ignore**, to suppress discovery of all instances of the server
6. To enable the custom server definition, select the **Enabled** check box.
7. To select an icon to associate with the custom server, click **Browse** and select the icon that you want to use.
8. Under Identifying Criteria, select either of the following options:
   - **All Criteria**, to match all of the identifying criteria
   - **Any Criteria**, to match any of the identifying criteria

9. Complete the following steps to define the criteria for the custom server:
   a. From the first list, select the criteria type.
   b. From the second list, select the operator.
   c. In the field provided, type the text argument for the criteria type and operator.
10. To remove the identifying criteria, click **Remove** or to define a new criteria, click **Add Criteria**.
11. To add configuration files, click the **Config Files** tab. The Config Files page is displayed.
12. On the Config Files page, click **Add**. The Search Path for Capture File window is displayed.
13. From the Type list, select one of the file types to capture:
   – Config File
   – Software Module
   – Application Descriptor Directory/File
14. From the **Search Path** list, select one of the following search paths for the configuration file:

`/`	This is the root of the file system.
`$PWD`	This is the current working directory of the running program.
`$Home`	This is the home directory of the user ID of the running program.
`C:`	This is the directory on your local computer.
`%ProgramFiles%`	This is the program files directory.
`%SystemRoot%`	This is the system root directory.

15. To capture the contents of the configuration file, click **Capture file contents** and optionally specify the maximum number of bytes of the captured configuration file.
16. To recurse through the directory structure to search for the specified file, click **Recurse Directory Content**.
17. Click **OK** to save the settings for your custom server.

5.2.3 Editing a custom server

To edit a custom server, complete the following steps from the Product Console:

1. In the Functions pane, select **Discovery** → **Custom Servers**. The Custom Servers pane is displayed in the workspace.
2. Select a custom server, and click **Edit**. The Custom Server Details notebook is displayed, with the Name and Type fields disabled. These fields cannot be changed.
3. To change the other fields in the Custom Server Details notebook, refer 5.2.2, “Adding custom servers” on page 114.
4. To refresh the information about the custom server you just changed, run another discovery. To improve the speed of the discovery process, limit the active scope of the discovery to the new component.

5.2.4 Copying a custom server

You may create a new custom server based on an existing one. This is done by copying a server listed in the Custom Servers pane and issuing it a unique name.

To copy a custom server, complete the following steps from the Product Console:

1. In the Functions pane, select **Discovery** → **Custom Servers**. The Custom Servers pane is displayed in the workspace.
2. Select the custom server that you want to copy and click **Copy**. The Set Name window is displayed.
3. In the Name field, type the name for the new custom server.
4. Click **OK** to save the new custom server.

5.2.5 Deleting a custom server

To delete a custom server, complete the following steps from the Product Console:

1. In the Functions pane, select **Discovery** → **Custom Servers**. The Custom Servers pane is displayed in the workspace.
2. Select the custom server that you want to delete and click **Delete**. A message window is displayed.

3. Click **Yes** to delete the custom server.
4. To confirm the deletion, ensure that the custom server is not listed in the Custom Servers pane.

5.2.6 Repositioning custom server entries

You can change the order in which custom servers are listed in the Custom Servers pane. The list order is important because template matching is applied from top to bottom in the custom server list and stops at the first match. For example, a more generic template matches all servers of a specific type and a more specific template matches only servers that have a specific string argument. After a server is matched to a server category, the custom server is removed from the unknown server list. A server cannot be a member of more than one category at the same time, even if the server matches criteria from several custom servers in the list. Changing the order of the list can cause the server process to match to a different custom server.

To reposition entries in the Custom Servers pane, complete the following steps from the Product Console:

1. In the Functions pane, select **Discovery** → **Custom Servers**. The Custom Servers pane is displayed in the workspace.
2. Select the custom server that you want to reposition and perform one of the following steps (see Figure 5-9 on page 120):
   - Click **Move Up**, to move the server up in the entry list.
   - Click **Move Down t**o move the server down in the entry list.

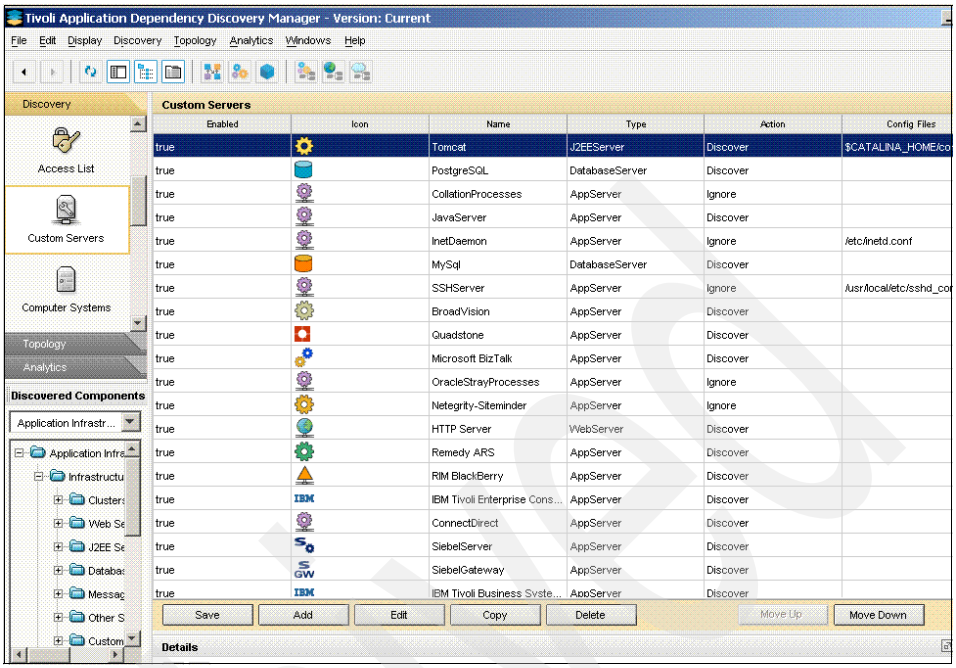

Figure 5-9 Selecting Move Down

5.3 Using discovery profiles

Discovery profiles help you discover your IT environment. TADDM discovers and collects configuration information for the entire application infrastructure, identifying deployed software components, physical servers, network devices, virtual LAN, and host data used in a runtime environment.

By using a discovery profile, you take control of what you discover. For example, you can configure individual sensors, manage multiple configurations of the same sensor, pick the appropriate configuration based on a set of criteria, and manage sets of configurations of different sensors to be applied on a single run. You can also specify a discovery profile's access list, and it is used only during a discovery with this particular profile.

A discovery profile access list works the same as a general access list. When you run a discovery, you must select a profile. If no profile is selected, the discovery is run against the default profile, which is Level 3 discovery.

The default profile can be changed by selecting **Edit** → **Preferences** and choosing another profile.

5.3.1 Creating discovery profiles

When creating discovery profiles, default profiles, default sensors, and default sensor configurations are not editable. To create discovery profiles, complete the following steps:

1. In the Discovery drawer of the Product Console, click **Discovery Profiles**. See Figure 5-10.

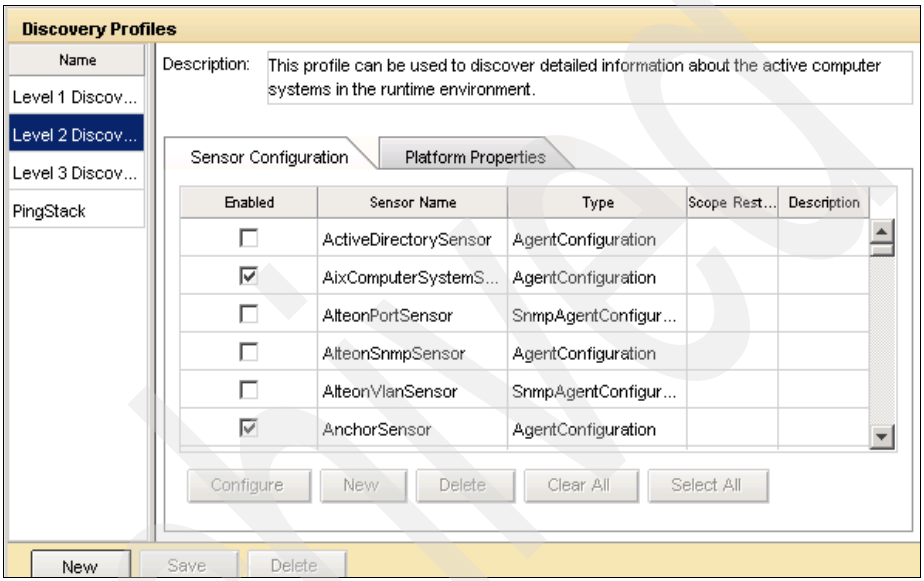

Figure 5-10 Discovery profiles

2. In the Discovery Profiles window, click **New**. See Figure 5-11.

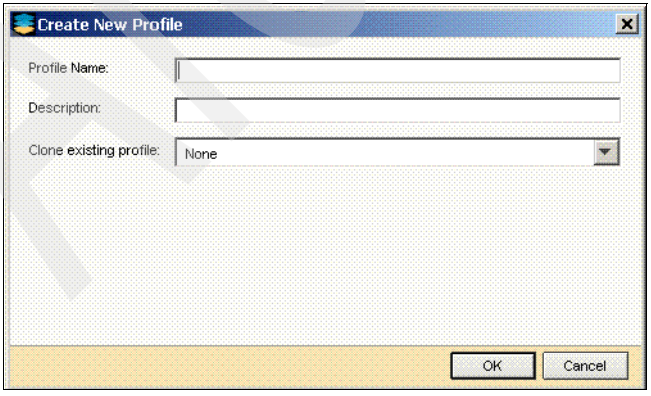

Figure 5-11 Create New Profile

3. Type the profile name. The profile name must be unique.
4. Type a description for the new profile. The description is displayed on the user interface with the Sensor Configuration, Access Control and Platform Properties pages.
5. When you create a new profile, you can use an existing profile as a basis. From the Clone existing profile list, select an existing profile or select **None**. Cloning an existing profile includes the agent configuration, access list, and platform configuration.

   The three levels of default discovery profiles to choose from are:

   – Level 1 discovery

     This profile can be used to perform credential-less discovery. It can be used to discover active computer systems in the runtime environment.

   – Level 2 discovery

     This profile can be used to discover detailed information about the active computer systems in the runtime environment. You can use the Level 2 profile to enable shallow discovery of applications running on a target system using only the system credentials by adding the following property to the `collation.properties` file:

     ```
     com.collation.internalTemplatesEnabled=true
     ```

     If you have this property set to `true`, you receive a CustomAppServer object representing the application running on the target machine. You do not have to provide application credentials to enable this property.

   – Level 3 discovery

     This profile can be used to discover the entire application infrastructure, deployed software components, physical servers, network devices, virtual LAN, and host data used in a runtime environment.
6. Click **OK**. The discovery profile is created and listed with the other existing profiles. See Figure 5-12 on page 123.

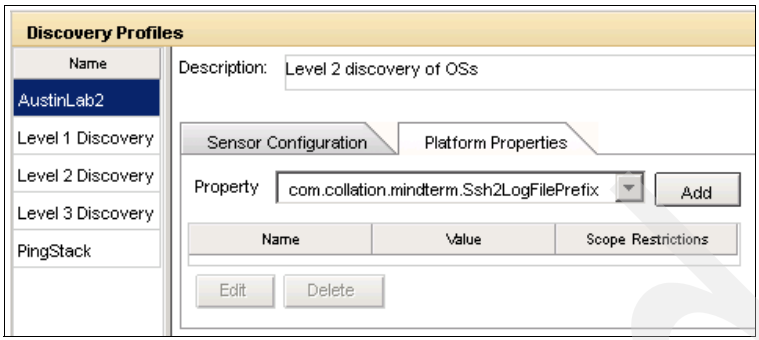

Figure 5-12 New discovery profile

Note: The profiles are listed next to the Sensor Configuration and Platform Properties pages. If you cannot see the profiles, look for a splitter bar beside Sensor Configuration page. Move the splitter bar to see the list of profiles. When you select a profile, the details for the profile are displayed on the Sensor Configuration and Platform Properties pages.

7. On the Sensor Configuration page, select a sensor and you can create, enable, and configure sensors. When you configure a sensor, double-click the value that you want to edit. You can add scope restrictions to a sensor. A scope restriction means that when a discovery is performed using a profile, the sensor runs only on the scope configured with this scope restriction.
8. On the Platform Properties page, you can add, edit, or delete properties for a platform.
9. Click **Save**.

Note: For example, If you want to quickly collect information about how many db2 servers are installed in a data center. You can create a new profile including Level 1 profile, and add DB2 sensor. Database server discoveries usually prerequisite successful discovery of the host system on which they reside.

5.4 Access collections

An *access collection* is a set of configuration items that is managed collectively for security purposes. TADDM does not manage access to configuration items

on an individual basis. Instead, the configuration items are aggregated into sets called access collections.

The security of each access collection is then managed by creating roles and assigning the roles to users as appropriate. Access collections are used to limit the scope of a role. The role applies only to the access collections that you specify when assigning the role to a user.

An access collection called DefaultAccessCollection (containing all configuration items) is created when TADDM is installed. All users have *read* and *update* permissions for this access collection by default, unless data-level security is enabled.

5.4.1 Creating an access collection

To control user access to configuration items, you must create an access collection. Before creating an access collection, run a discovery to ensure that the database of configuration items is up to date.

To create an access collection, perform the following steps from the Product Console:

1. Select **Edit** → **Create Collection**. The Create Collection dialog is displayed. See Figure 5-13.

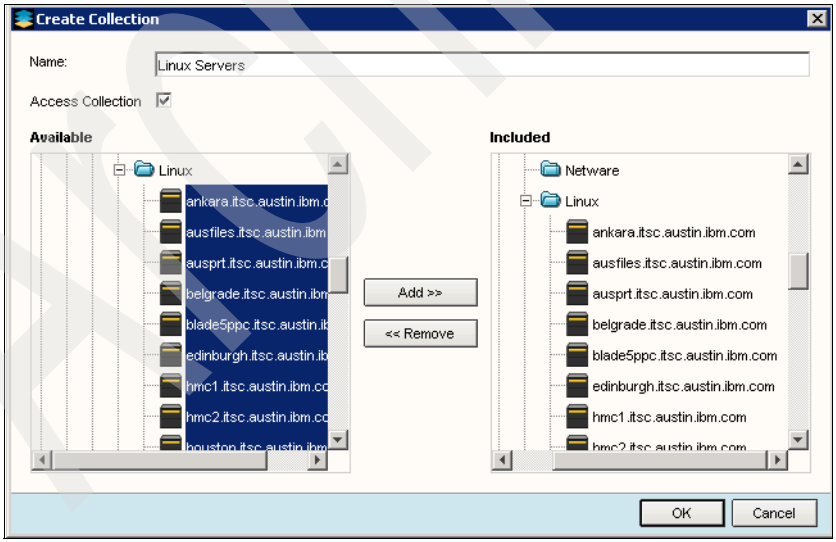

Figure 5-13 Create Collection

2. Type a name for the collection.

3. Select the **Access Collection** check box.
4. Select the configuration items you want to include in the collection, click **Add**.
5. Click **OK**. The access collection is created.

5.4.2 Editing an access collection

You can edit an access collection to change its contents. Before editing an access collection, run a discovery to ensure that the database of configuration items is up to date.

To edit an access collection, perform the following steps in Product Console:

1. In the Discovered Components list, select **Collections**. A list of collections is displayed.
2. Right-click a collection and click **Edit**. The Create Collection dialog is displayed.
3. To modify the list of configuration items included in the collection, select the configuration items, and click **Add** or **Remove**.
4. Click **OK**. Your changes are saved.

5.4.3 Deleting an access collection

You can delete an access collection that is no longer necessary.

Important: Deleting a collection causes it to be disbanded. The configuration items belonging to the collection are not deleted; they simply are no longer aggregated under the collection's name.

To delete an access collection, perform the following steps from the Product Console:

1. In the **Discovered Components** list, select **Collections**. A list of collections is displayed.
2. Right-click a collection and click **Delete**. A confirmation dialog is displayed.
3. Click **Yes**. The access collection is deleted.

5.5 Discovering business applications and business services

Today IT management is not only the support department for the whole enterprise, it is part of the enterprise business service management. Enterprises should not only be concerned about a failed component, they should be more concerned with the impact of that component on the business.

TADDM can help to understand the services that IT provides for the enterprise business. Building a business application definition is the first step in business service management (BSM).

After TADDM discovers or imports all the IT resources from different resources, business applications or business services can be created manually or automatically in TADDM.

Business application is the way to group the different kinds of IT resources into a logical group, this logical group acts together as one unit to provide some kind of service. Business applications can be treated as units for discovery and analytics. Based on the dependencies and relationships discovered by TADDM, the user can better understand their IT environment and utilize them more efficiently.

The top level in the component hierarchy of TADDM is the *business service*. Business services can contain any number of the lower level resources, from business applications to EAR modules in a WebSphere server or specific configuration files on systems.

The purpose of the business service is to consolidate multiple lower-level objects and their relationships in order to perform reporting and analysis considering all related resources.

Business applications can be defined at different levels:

- Container level, for example:
  - WebLogic server
  - Oracle DB
- Deep module level, for example:
  - WAR files defining a module within an application server
  - EAR files, ASP assemblies, and more
- DB schema within a database instance

Business applications may include:

- Components
- Dependencies
  - Inter-component dependencies
  - Dependencies to other business applications
  - Dependencies to other business services
- Admin Information

For more information about Discovery Schedule refer to Redbooks publication *IBM Tivoli Application Dependency Discovery Manager Capabilities and Best Practices*, SG24-7519.

5.6 The TADDM Enterprise Domain Manager

The TADDM Enterprise Domain Manager provides TADDM functionality for an entire enterprise, thus enabling several domains to be managed by a single TADDM Enterprise Domain Server.

5.6.1 TADDM Enterprise Domain Manager overview

The TADDM Enterprise Domain Manager is a Web-based interface that you use to manage several TADDM Domain Managers. The TADDM Enterprise Domain Manager contains the same components as a TADDM Domain Manager for a single domain with the addition of enterprise-specific components.

Table 5-1 lists the enterprise-specific components.

Table 5-1 Enterprise-specific components

Component	Description
TADDM Enterprise Domain Server	The TADDM Enterprise Domain Server communicates with all of the individual servers to collect information about each domain.
TADDM Enterprise Domain Database	The repository for configuration item information collected by the TADDM Enterprise Domain Server. The TADDM Enterprise Domain Database can also contain data loaded by the bulk load program on the TADDM Enterprise Domain Server.
TADDM Enterprise Domain Manager	A graphical application to view and analyze information about domains within your enterprise.

A TADDM Enterprise Domain Server contains information obtained from a collection of TADDM Domain Managers. Several TADDM Domain Managers can be located throughout your enterprise, with each one containing detailed information about a specific domain. The TADDM Enterprise Domain Database can operate in the following two modes:

- Deep mode

  This mode is the default. The TADDM Enterprise Domain Database synchronizes all of the information contained in single domains. Thus, when this information is retrieved in an enterprise environment, it is retrieved completely from the TADDM Enterprise Domain Database. There is no runtime access to the single domain to retrieve deep information when TADDM Enterprise Domain Database operates by default in this mode. The TADDM Enterprise Domain Server operates by default in this mode, as the `COLLATION_HOME/etc/domainquery` file on the TADDM Enterprise Domain Server contains the text SYNC_ALL_ATTRS. When the domainquery file begins with SYNC_ALL_ATTRS, any remaining contents of the file are ignored and deep synchronization is performed.

- Shallow mode

  The TADDM Enterprise Domain Database stores a small set of top-level information contained in the TADDM Domain Database. However, TADDM Enterprise Domain Database can query each TADDM Domain Database for more detailed information as needed. This reduces the amount of information that needs to be held at the TADDM Enterprise Domain Database, enabling it to hold more objects and process more transactions than a TADDM Domain Database. The default top-level information that is locally stored in the TADDM Enterprise Domain Database is specified in the `COLLATION_HOME/etc/domainquery.shallow` file on the TADDM Enterprise Domain Server. This file can be customized.

5.6.2 Exploring the TADDM Enterprise Domain Manager Console

You can perform most of the same operations with the TADDM Enterprise Domain Manager as you can with the TADDM Domain Manager for a single domain.

The only two differences between the tabs on the Functions pane of the TADDM Enterprise Domain Manager and those of the TADDM Domain Manager are the Discovery tab (which is only in the TADDM Domain Manager) and the Domain Management tab (which is only in the TADDM Enterprise Domain Manager).

Starting the TADDM Enterprise Domain Manager

When you select TADDM Enterprise Domain Database during installation, the Domain Manager link on the TADDM Launch page launches the TADDM Enterprise Domain Manager rather than the TADDM Domain Manager for an individual domain. When a domain is added to the TADDM Enterprise Domain Manager, TADDM Domain Manager for the added domain is accessible from the TADDM Enterprise Domain Manager using the **Launch** button as described in “Exploring the Domain Summary pane” on page 129.

To start the TADDM Enterprise Domain Manager, complete the following steps:

1. Open a Web browser and enter the Web address of the system where you installed the TADDM product. For example, enter something similar to the following Web address format:

   `http://system.company.com:9430`

   The TADDM Launch page is displayed.
2. Click the link for the **Domain Manager**. The TADDM Enterprise Domain Manager login window is displayed.
3. Enter the user name and password of the default administrator and click **Login**. The TADDM Enterprise Domain Manager is displayed.

Accessing domain summary information

To access the domain summary information for all domains in your enterprise, in the Functions pane, click **Domain Management** → **Domain Summary**.

Exploring the Domain Summary pane

The Domain Summary pane contains a table with the following fields:

- Domain: Name of this domain
- Host Name: Name of the host for this domain
- Last Synchronized: Time of the last synchronization for this domain
- Domain Status: Status of the host

The Distributed Domain Summary section contains the following buttons:

- Add: Adds a domain to your enterprise
- Edit: Edits the selected domain in your enterprise
- Delete: Deletes the selected domain from your enterprise
- Refresh: Updates the Domain Summary table information for the selected domain
- Launch: Starts a Product Console for a domain in your enterprise

- Schedule: Schedules synchronization of the TADDM Domain Databases to the TADDM Enterprise Domain Database
- Inventory: Opens the Inventory Summary pane for the selected domain

Adding or changing a domain

You can add a new domain to your enterprise or change an existing domain.

Note: To add a domain or change an existing domain, you must login to the TADDM Enterprise Domain Manager with a user that has admin runtime permission.

The Add Domains pane is shown in Figure 5-14.

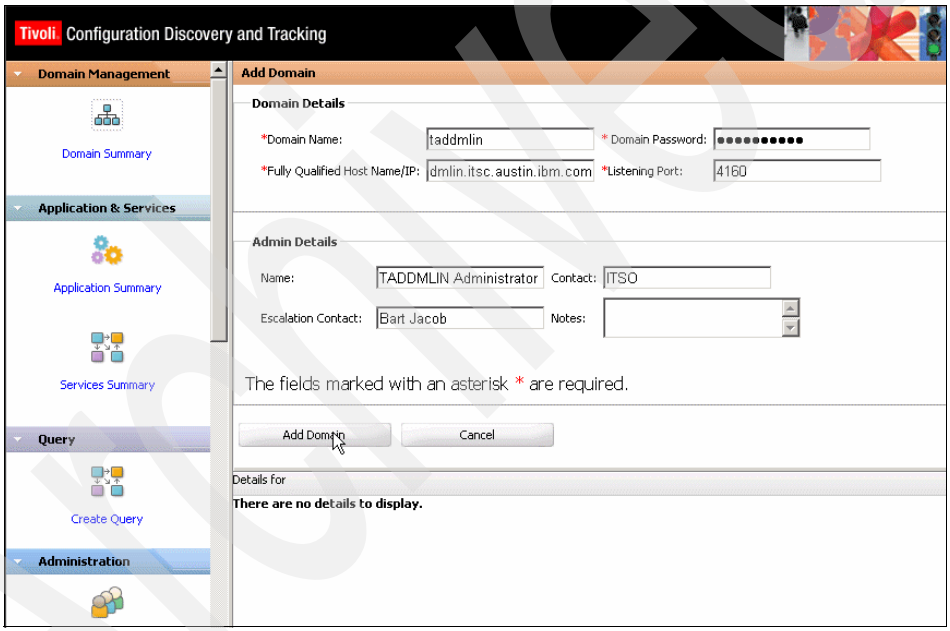

Figure 5-14 Add Domain

The Add Domains and Edit Domains panes contain the following sections:

- Domain Details

  Use this section to enter information describing the domain that you are adding or changing.

- Admin Details

  Use this section to enter information about the contacts for this domain.

The Domain Details section of the Add Domain and Edit Domain panes contains the following fields:

- Domain Name

  (Required): This field is for the name of the domain.

- Domain Password

  (Required): This field indicates the password to use to log into the TADDM Domain Server. Use the SSL passphrase of the domain. To obtain the passphrase, use the value of the `com.collation.sslpassphrase` property from the domain server in the following file:

  `COLLATION_HOME/dist/etc/collation.properties`

- Fully Qualified Host Name/IP

  (Required): This field is for the fully qualified host name or IP address of the TADDM Domain Server.

- Listening Port

  (Required): This field indicates the listening port of the TADDM Domain Database. Use the unicast discovery port of the domain.

  To obtain the unicast discovery port, use the value of the `com.collation.jini.unicastdiscoveryport` property from the domain server in the `COLLATION_HOME/dist/etc/collation.properties` file. The default value is `4160`. In the Edit Domain pane, the fields are completed with current values.

The Admin Details section of the Add Domain and Edit Domain panes contains the following fields:

- Name: The name of the domain administrator
- Contact: The contact for the domain
- Escalation Contact: The name of the escalation contact for the domain
- Notes: User notes about the domain

The Add Domain and the Edit Domain panes contain the following buttons:

- Add Domain: (Add Domain pane only) Adds this domain
- Save Changes: (Edit Domain pane only) Saves the changed information
- Cancel: Returns to the Domain Summary pane without saving any information
- Test Connection (Edit Domain pane only): Tests that TADDM Enterprise Domain Manager can contact the Domain using the specified Domain Details

Adding a domain to your enterprise

To add a domain to your enterprise, complete the following steps:

1. In the Domain Summary pane, click **Add**. The Add Domain pane is displayed.
2. In the Add Domain pane, enter the domain information.
3. To apply the information you entered, click **Add Domain**.

Changing a domain in your enterprise

To change the domain information for an existing domain, complete the following steps:

1. In the Domain Summary pane, select the domain to be changed and click **Edit**. The Edit Domain pane opens.
2. In the Edit Domain pane, update the domain information.
3. To ensure that the TADDM Enterprise Domain Manager can contact the Domain using the specified Domain Details, click **Test Connection**.
4. If you changed the information of a domain, click **Save**.

Deleting a domain

You can delete a domain from your enterprise. To delete a domain, you must log in to the TADDM Enterprise Domain Manager as a user that has the admin runtime permission. When a domain is deleted from an TADDM Enterprise Domain Database, any access collections that need to be deleted must be manually deleted using either the API or api.sh. If the domain that you are deleting has authorization policies for access collections that were synchronized to the TADDM Enterprise Domain Database, you must manually remove access to these access collections using the TADDM Enterprise Domain Manager.

To delete a domain from your enterprise, complete the following steps:

1. In the Domain Summary pane, select a domain.
2. To delete the domain, click **Delete**. A prompt is displayed to confirm that you want to delete the selected domain.
3. In the prompt, click **OK**. The domain is deleted from your enterprise and removed from the Domain Summary table.

Displaying enterprise inventory information

To display an inventory summary for the domains in your enterprise, complete the following steps:

1. In the Domain Summary pane, select the domain for which you want to display an inventory summary.
2. Click **Inventory** in the Domain Summary pane. The Inventory Summary pane is displayed.

Displaying topology information

You can display topology information for all domains in your environment by clicking the items listed under Topology in the Functions pane. The TADDM Enterprise Domain Manager offers a graphical interface that displays components and business applications within an interconnected graph. You can also display a hierarchical view of the discovery data. The Topology tab in the TADDM Enterprise Domain Manager operates in the same way as it does in the TADDM Domain Manager. However, in the enterprise environment the components in your enterprise can be in more than one domain and the topology information will reflect this.

6

Problem Determination

This chapter focuses on the problem determination of Tivoli Application Dependency Discovery Manager V7.1. Problem determination is a critical concept to understand and to successfully use, deploy and support any version of Tivoli Application Dependency Discovery Manager.

This chapter contains the following topics:

- "Log files" on page 136
- "Memory issues" on page 140
- "Name resolution issues" on page 141
- "Access and discovery issues" on page 141
- "Relationships" on page 147

6.1 Log files

Log files provide assistance in troubleshooting issues with discovery problems and with functions of TADDM. These log files are located in a central location of the `$COLLATION_HOME/log` directory.

The following log files are typically the most useful:

- error.log
- local-anchor*.log
- tomcat.log
- services/DiscoverManager.log
- services/TopologyManager.log

Table 6-1 lists all TADDM log files.

Table 6-1 TADDM log files

File	Description
bulkload.log	Messages about the bulk load program
cdm.log	Messages about the Web portal
control.log	Messages about the starting, stopping, and status of the TADDM Server
discover.log	Messages from the Discover jini service
discover-admin.log	Messages from the DiscoverAdmin jini service
error.log	Error messages from TADDM services
events-core.log	Messages from the Events Core jini service
local-anchor*.log	Messages from J2EE application server sensors, such as WebSphere and WebLogic
login.log	Audit trail for user logins
l2.log	Messages from the topology builder process
proxy.log	Messages from the Proxy jini service
tomcat.log	Messages about application server activity
topology.log	Messages from the Topology jini service
services/ApiServer.log	Messages about TADDM APIs

File	Description
services/ChangeManager.log	Messages about the processing of change events after discovery completes
services/ClientProxy.log	Messages about the GUI
services/DiscoverManager.log	Messages about sensor activity
services/DiscoverObserver.log	Messages about the movement of completed work items from the discover manager to the topology manager
services/MonitorStateManager.log	Messages about the processing of discovery and change events
services/ProcessFlowManager.log	Messages about the event processing engine for discovery
services/ReportsServer.log	Messages about the processing of reports
services/TopologyManager.log	Messages about the interface between the data store and all other components
services/ViewManager.log	Messages about the building of the topology graphs and navigation trees for configuration items (CIs)

In TADDM, you can set logging levels either globally or locally for each Java Virtual Machine (JVM). You can also use split logging, which creates a separate log file for each sensor, and dynamic logging, which enables you to change logging levels without restarting the TADDM Server. The log level can be applied selectively.

JVMs for which local logging can be set are:

- Discover
- DiscoverAdmin
- EventsCore
- Proxy
- Topology

In `$COLLATION_HOME/etc/collation.properties`, JVM log levels are set with the following property:

```
com.collation.log.level.vm.<vm_name>=<log_level>
```

The `vm_name` is the name of the JVM; and `logging_level` is the level that you want to set for that JVM.

If the log level is not in the properties file, add it. The valid log levels are:

- FATAL
- ERROR
- WARN
- INFO (default)
- DEBUG
- TRACE

Be careful when using DEBUG, because it can affect performance. So use it in cases of active troubleshooting situations. Be sure to allocate extra space for topology logs when long term troubleshooting is required. To increase these values modify the properties in the `collation.properties` file, shown in Example 6-1.

Example 6-1 collation.properties

```
# file size of a rollover log file.
com.collation.log.filesize=20MB
#Number of logfiles before rollover
com.collaton.log.filecount=5
```

In the example, the parameters delete the oldest data, when all the space is allocated by the values of these properties.

Do not set the logging level to TRACE without a request from IBM Support. It can expose passwords and generate large volumes of log messages.

6.1.1 SplitSensor logging

SplitSensor logging makes troubleshooting much easier and improves both support and the customer's ability to read and use the logs to diagnose problems. To set this feature, you set the following parameter in the `collation.properties` file:

```
com.collation.discover.engine.SplitSensorLog=true
```

When split logging is used, the sensor log files are put into the following directory structure, where `runid` includes the date and time of the discovery, and the log file name (`sensorName-IP`) includes the sensor name and the IP address of the system that is discovered:

```
$COLLATION_HOME/log/sensors/runid/sensorName-IP.log
```

A sample log file path and name is:

```
$COLLATION_HOME/log/sensors/20070621131259/SessionSensor-10.199.21.104.log
```

6.1.2 Dynamic logging

When this logging is used, the changes that are made and saved to the logging levels do not require a restart of the TADDM server. The changes take affect in about 60 seconds. The only file requiring the TADDM to be restarted is the `tomcat.log` file. The changes for dynamic logging are performed in the file `collation.properities`, with the following parameter:

`com.collation.deploy.dynamic.logging.enabled=true.`

6.1.3 Extra debugging

The extra debugging feature increases the output in the log files and is set in the `collation.properities` file with a `true` or `false` setting. It also can cause space issues, so ensure that there is adequate space before setting this value. The feature is similar to other LOG4J loggers in how it accepts the log level settings. This deeper setting (for more detailed debugging) enables you to see looping issues. It shows, for example, verification of sessions that normal debugging does not. See Example 6-2. This level of logging provides a much lower level of granularity.

Example 6-2 Extra debugging

```
(i.e. 2008-07-28 11:30:03,477 DiscoverManager [DiscoverWorker-15]
SessionSensor-130.186.22.77 DEBUG session.AbstractSessionClient -
execute(write sys$output "AbstractSessionClient verifying session"):
failure [extra debugging]
```

6.1.4 Collecting data for IBM

After you have the data collected, you have to ensure the following items accompany the data:

- Operating system type and version of the TADDM server and database machine of the domain machine, ECMDB machine, gateway and anchor
- Operating system type and version of the target machine, gateway and anchor
- Type of database and version
- Version of TADDM, patch lever and e-fixes on the TADDM server, domain machine, ECMDB machine, gateway and anchor

- Memory of the TADDM server, database machine, domain machine, ECMDB machine, target machine. gateway and anchor
- Processor of the TADDM server, database machine, domain machine, ECMDB machine, target machine, gateway and anchor

6.2 Memory issues

Memory issues typically cause out-of-memory conditions. There are several ways to look into this type of problem. First check for `core.XXX.XXX.XXX.dmp` files found in the log directory.

Next, look in the `$COLLATION_HOME/external/gigaspaces-4.1/bin` for javacores and heap dump files. Remember, that when diagnosing out-of-memory conditions, you must have log files with the same date and time stamps as the javacore and heap dump file.

After you have located the javacore file, first determine whether an out-of-memory condition exists by opening the javacore; it is typically at the top of the file. If you find an out-of-memory condition, determine the service name. You can search that file for service name. After you have the service name, go to `collation.properties` file and increase the *jvmargs* for that service name. Now this is where the tricky part comes in. *Do not* automatically change the jvmargs heap size. Before taking this action, the heap dump should be analyzed to determine whether increasing the heap is the correct action. The following example, shows how to increase it:

```
com.collation.Topology.jvmargs=-Xmx2048M
```

This parameter is in the JVM vendor section of the `collation.properites` file. If it does not exist, you can add it. Keep in mind that modifying `jvmargs` above 1.5 GB can make heap dumps difficult or impossible to analyze, so be very careful and make sure to completely investigate and assess the situation before increasing these values.

> Important: The `-Xms` value should always follow the `-Xmx` value and should always be smaller or equal to the `-Xmx` value.

6.3 Name resolution issues

Name resolution is very important because networked environments are dependent on being able to resolve to names and IP addresses in order to send and receive data. Therefore, be sure that the following entry is always in the `/etc/hosts` file of all TADDM machines and targets:

```
127.0.0.1  localhost loopback
```

Although having the entry in the file is a good practice for networking purposes, another reason for having the entry is to prevent subsequent issues with Product Console logins or other issues with TADDM.

Make sure the forward and reverse name resolution is working by IP address and host name. Host name is especially important if DNS is being used. Two commands to check DNS resolution are:

```
ping -s <hostname>
nslookup <hostname>
```

6.3.1 Full Qualified Domain Names (FQDN)

Fully Qualified Domain Names (FQDN) are locators that identify the server within the organization that the system belongs to. For TADDM to know about the host names, set the following parameters in the `collation.properties` file:

```
com.collation.platform.os.disableDNSLookup
com.collaton.platform.os.disableRemoteHostDNSLookup
```

If you want to see the host names and not the IP addresses, set the values to `true`, so the server can resolve the FQDN of the local and remote systems.

6.4 Access and discovery issues

Discovery is done by using sensors with TADDM to determine how hosts and applications are configured with the environment. These sensors use secure network connections, encrypted access credentials, and native host utilities to perform the discovery process.

6.4.1 Testing the connection

To test the secure connection accessibility through SSH, use the **testssh.py** script, which is located in the `$COLLATION_HOME/support/bin` directory. The command is run against a specific host, from the TADDM server to check if a discovery will run from a secure shell. The syntax for the command is:

- Before TADDM V7.1:

  ```
  testssh.py <IP> "<some command>"
  ```

- TADDM V7.1 and later:

  ```
  testssh.py <IP> -u <TADDM user> -p <password> "<some command>"
  ```

For example:

```
testssh.py 10.10.15.15 -u administrator -p collation "lsof -nP -i -C
```

6.4.2 WebSphere discovery

Several tools exist within the `$COLLATION_HOME/support/bin` directory to test connections, accessibility, and environments. One of the tools, **testwasconnection.sh**, addresses WebSphere (WAS) issues. This script specifically is designed to test the JMX™ connect to verify credentials and connectivity to WAS. Example 6-3 shows how to use this script.

Example 6-3 testwasconnection.sh usage information

```
General    - This script tests connectivity to a WAS installation
#
#  Usage Info -
#  There are 3 sets of options that can be supplied to the script.
#   (1) host version
#   (2) host version port connectorType
#   (3) host version port connectorType user password trust-store
trust-passphra
se
#       key-store key-passphrase
#
#   Parms:
#    host          - required, the host containing the WAS installation
#    version       - required, the WAS version, valid values are 5.x,
6.0, 6.1.
#                    if you don't know the version, use 5.x
#    port          - optional, the port for the connection. Default is
8880
```

```
#    connectorType - optional, valid values are SOAP or RMI. Default is
SOAP
#    user          - optional, user id to access WAS
#    password      - optional, password for user id
#    trust-store   - optional, fully qualified location of trust store
#    trust-passphrase - optional, passphrase for trust-store.
#    key-store     - optional, fully qualified location of key store
#    key-passphrase - optional, passphrase for key-store
```

Example 6-4 has samples of using the script to debug problems.

Example 6-4 Examples

```
#   ./testwasconnection.sh yourhost.com 5.x
#   ./testwasconnection.sh yourhost.com 6.0 8880 SOAP
#   ./testwasconnection.sh yourhost.com 6.1 8880 SOAP wasadmin foobar
#       /opt/WebSphere/AppServer/profiles/default/etc/DummyClientTrustFile.jks
#       WebAS
#       /opt/WebSphere/AppServer/profiles/default/etc/DummyClientKeyFile.jks
#       WebAS
```

6.4.3 StackScan

The StackScan sensor requires and uses raw sockets from the operating system to send packets to endpoints to do the stack analysis. Use of these sockets is not a function or command of the operating system. Therefore, StackScan is given direct access in the *sudoers file*. The sudoers file is restricted to either *all* or to specific operating system commands.

Note: StackScan runs on all anchors and requires sudo on each one, and the TADDM server is an anchor by default.

Configuring sudo access control on UNIX/Linux

The StackScan sensor requires sudo access control to collect discovery information. For Windows operating systems, sudo access control is not necessary. To configure sudo access, complete the following steps for the TADDM server and anchor hosts:

1. From a command prompt window, use the **su** command to switch to root authority on the local host.
2. Type the **visudo** command.
3. Type the following line in the `/usr/local/etc/sudoers` or `/etc/sudoers` file:

   `<`*TADDM_USER*`>ALL=(ALL) NOPASSWD:ALL`

In the command line, <TADDM_USER> is the non-root user ID that is used by the Configuration Discovery and Tracking server.

Note: This is only required on anchor servers, including the TADDM server.

6.4.4 Database connectivity

You can check for connectivity to the database in the following three ways:

- Run a query on the CDT database. You can find this script in the .../dist/bin directory. For example:

  ```
  dbquery.sh_'select_count(*)_from_compsys'
  ```

- Use the **testjdbc.py** script, located in the .../dist/support/bin directory of the CDT Server.

  This script checks the JDBC connection, independent of any collation.properties settings.

  The syntax is as follows:

  ```
  testjdbc_{o?racle?|s?ybase?|d?b2}_{ip|host}_port_{user}_{password}
  {oracleSID|sybabe_db|DB_name}
  ```

  For example, connecting to a DB2 database on frwlnx06 port 50005 user and password ctginst1 database name is ECMDB:

  ```
  dist/support/bin/testjdbc.jy_d_frwlnx06_50005_ctginst1_ctginst1_ECMDB.
  ```

- Check for a connection to the instance on the DB server itself after sourcing the CDT user environment, which includes the database user environment.

  Use the database native commands to connect to the database, such as **sqlplus** or **db2** commands.

6.4.5 Mapping

The **lsof** command is used to map sockets, pipes, and files that are opened by a process. TADDM uses this for mapping during discoveries. This approach helps TADDM in identifying what is associated to a process. Therefore, when template-matching is performed, TADDM knows what process is running. When overriding the **lsof** command to a specific IP address is necessary, TADDM does have a way to accomplish this in the collation.properties. For example:

```
com.collation.discover.agent.command.lsof.SunOS.10.10.10.15=lsof
```

6.4.6 Scopes

When large scopes are performed during a discovery, database deadlocks are more likely to occur, and performance decreases. Database deadlocks cause the database to roll back the deadlocked transaction. Then, there is a wait time before a retry is attempted, and the limit is governed by the following parameter:

`com.collation.discover.osbserver.topopumpstorageattempts`

However, to control database deadlocks, use the following parameter, in the `collation.properties` file:

`com.collation.discover.topopumpcount`

This parameter has a default value of `16`.

Important: Be careful when changing this value. Any time you consider changing this value, support should be involved.

6.4.7 Database configurations

Database configuration parameters are located in the `collation.properties` files. Any time host names, users, ports, and other database configuration parameters are changed, they will have to be modified accordingly. These parameters determine how you connect to the database. See the database configuration parameters in Example 6-5.

Example 6-5 Database configuration parameters

```
com.collation.unixuser=cmdbusr1
com.collation.unixgroup=503
com.collation.db.sid=cmdb
com.collation.db.port=50000
com.collation.db.type=db2
com.collation.db.server=josephine.tivlab.austin.ibm.com
```

6.4.8 Expired password

If the TADDM administrator password expires, use these steps to restore it:

1. Stop the TADDM server.
2. Set the operating system clock back one month.
3. Start the TADDM server.

4. In the Domain Manager, set the expiration date to 12/31/2020 for the administrator accounts.
5. Stop the TADDM server.
6. Start the TADDM server.
7. Go to directory `<taddm_install>/dist/support/bin` and run the **`encryptprops.sh`** script.

   This re-encrypts the password that was typed in plain text in the `collation.properities` file.

Change a password

To change a password, perform the following steps:

1. In the `collation.properties` file, find the password that has to be changed and type it in plain text.
2. Save the file.
3. Go to directory `<taddm_install>/dist/support/bin` and run the **`encryptprops.sh`** script.

   This re-encrypts the password that was typed in plain text in the `collation.properities` file.

6.4.9 rmi.clientproxy.server.hostname parameter

When bringing up the TADDM Product Console with HTTP or HTTPS, the TADDM server downloads the Java Network Launch Protocol (JNLP) file. While downloading the `jnlp` file, the server tells the client which IP address that the client should use to contact the server. By default, the server does a reverse-lookup on its host name and sends that IP address to the client. If the server's host name resolves to the loopback address of 127.0.0.1 (sometimes in Linux the default is the `/etc/hosts` file), then the client tries to connect to itself. Therefore, the purpose for the `com.collation.clientproxy.rmi.server.hostname` parameter is to prevent the Product Console from connecting to itself. It allows FQDN lookup.

6.4.10 sslpassphrase

Another parameter used in the realm is `sslpassphrase`, which is used specifically by Jetty HTTPS and GUI HTTPS server.

Note: Jetty project is hosted by the Eclipse Foundation:

http://www.eclipse.org

It provides an HTTP server, HTTP client, and `javax.servlet` container:

http://java.sun.com/javaee/5/docs/api/javax/servlet/package-summary.html

This parameter is set in the `collation.properties` file by the following parameter:

```
# SSL passphrase used by Jetty HTTPS server and GUI HTTPS server
com.collation.sslpassphrase=
```

This can affect TADDM's ability to start if it is not set correctly, and will throw errors in the `tomcat.log` file, similar to Example 6-6.

The setting of `sslpassphrase` is in file `collation.properties` file.

Example 6-6 tomcat.log

```
The tomcat.log shows error "SEVERE: Error initializing endpoint
java.io.IOException: Keystore was tampered with, or password was incorrect".
The installer sets the password for the keystore (serverkeys) to the
sslpassphrase. This error means that either the serverkeys file has been
tampered with or the sslpassphrase has been changed.
```

6.5 Relationships

The two types of relationships within TADDM are implicit relationships and explicit relationships:

- Implicit relationships are defined by the Common Object Model. Any relationship that TADDM builds during discovery or with the UI is an implicit relationship.

  Tip: if you can see the relationship in the TADDM UI, it is an implicit relationship.

- Explicit relationships are built outside of TADDM. These can come from other products like CCMDB, they can be imported through DLA, or they can be created through the API.

If query TADDM through the API, most of the information you should have will be implicitly defined in the data associated with CIs. You can use the API's **find** command to query data for individual CIs with a (depth of one) to get implicit relationships. The depth of one returns the relationships, but none of the information about the specific CIs, which you likely already have from the previous queries.

You may also have certain explicit relationships that you want to include in your integration. Use a **find** command to query those explicit relationships, as shown in Example 6-7.

Example 6-7 Explicit relationships

```
rs = find(query.toString(), 1, null, permissions)
where query.toString() returns the following query:
select * from Relationship where (source == &lt;guid&gt; or target ==
&lt;guid&gt;) and not generated
```

Important: The *find relationships API query* should be used with caution. It returns graphs of fully populated, related CI objects, and can therefore fetch and return unexpectedly large amounts of data.

7

Administration

In this chapter, we describe the administration tasks for IBM Tivoli Application Dependency Discovery Manager (TADDM) v7.1. We explain various ways to perform administration of TADDM, features, and interactions with other systems.

This chapter contains the following topics:

- "Manually starting and stopping the TADDM server" on page 150
- "Updating the database passwords" on page 153
- "Discovery schedules" on page 154
- "Synchronization schedules" on page 157
- "Versions" on page 159
- "Manual component creation" on page 161
- "Manual dependency creation" on page 163
- "Business applications and business services" on page 166
- "Roles and permissions" on page 170
- "Application programming interface" on page 173

7.1 Manually starting and stopping the TADDM server

You can start and stop the TADDM server manually by using the **control** command, which can also be used to test the server status.

For AIX, Linux, Linux on System z, and Solaris operating systems, the **control** command is located in the following path:

```
$COLLATION_HOME/bin/control
```

For Windows operating systems, the **control** command is located in the following path:

```
%COLLATION_HOME%\bin\control.bat
```

Only the non-root user that was defined during the installation process can run the **control** command. This section describes the usage of this command.

7.1.1 Starting the TADDM server

Note: A local or remote database server must be started and running before the TADDM server code is started. The TADDM server cannot initialize or run properly if the database is not available.

To manually start the TADDM Server, complete the following steps:

1. Log in as the non-root user that was defined during the installation process.
2. Open a command prompt window.
3. Go to the directory `$COLLATION_HOME/bin`.
4. Use one of the following commands to run the start script:
   - For AIX, Linux, Linux on System z, and Solaris operating systems:

     ```
     ./control start
     ```
   - For Windows operating systems:

     ```
     startServer.bat
     ```

Note: With the **COLLATION_HOME%\bin\control.bat start** command, the TADDM server remains running as long as the user is logged on. When the user logs out, the TADDM server stops because that user owns the process. However, the TADDM server can also be started as a service by using the **COLLATION_HOME%\bin\startServer.bat** command issued by the *run-as* user that was specified during installation. With this command, the TADDM server remains when the run-as user logs off the system.

7.1.2 Restarting the TADDM server

To manually restart the TADDM Server, complete the following steps:

1. Log in as the non-root user that was defined during the installation process.
2. Open a command prompt window.
3. Go to the directory `$COLLATION_HOME/bin`.
4. Use one of the following commands to run the start script:
   - For AIX, Linux, Linux on System z, and Solaris operating systems:

     ```
     ./control restart
     ```
   - For Windows operating systems:

     ```
     control.bat restart
     ```

7.1.3 Stopping the TADDM server

To manually stop the TADDM Server, complete the following steps:

1. Log in as the non-root user that was defined during the installation process.
2. Open a command prompt window.
3. Go to the directory `$COLLATION_HOME/bin`.
4. Use one of the following commands to run the start script:
   - For AIX, Linux, Linux on System z, and Solaris operating systems:

     ```
     ./control stop
     ```
   - For Windows operating systems:

     ```
     control.bat stop
     ```

7.1.4 Testing the TADDM server status

To test the TADDM server status, complete the following steps:

1. Log in as the non-root user that was defined during the installation process.
2. Open a command prompt window.
3. Go to the directory `$COLLATION_HOME/bin`.
4. Use one of the following commands to run the start script:
   - For AIX, Linux, Linux on System z, and Solaris operating systems:

     ```
     ./control status
     ```

   - For Windows operating systems:

     ```
     control.bat status
     ```

The TADDM server status is displayed as in Example 7-1.

Example 7-1 The TADDM server status

```
[admin@TADDM bin]$ ./control status
Discover: Started
DbInit: Started
GigaSpaces: Started
Tomcat: Started
Topology: Started
DiscoverAdmin: Started
Proxy: Started
EventsCore: Started

TADDM: Running
```

Note: You can see the TADDM server status information also on the TADDM Web GUI under the Administrator Console section as shown in Figure 7-1 on page 153.

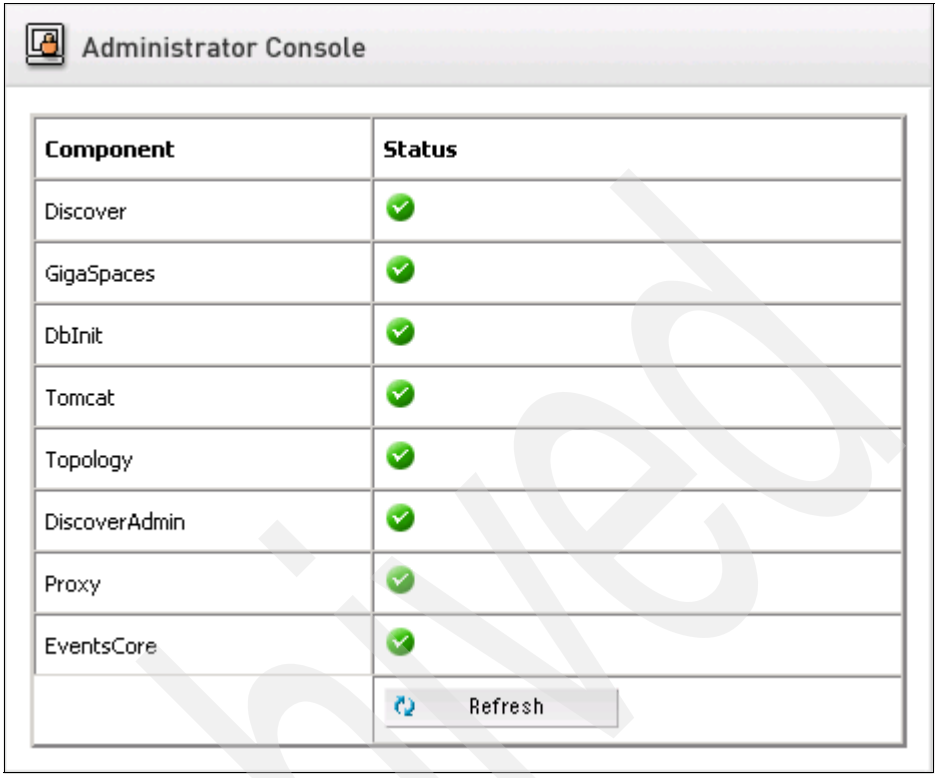

Figure 7-1 The TADDM server status

7.2 Updating the database passwords

The database passwords for both the database user and archive user are stored in the `$COLLATION_HOME/etc/collation.properties` file.

To update the database passwords, complete the following steps:

1. Log in as the non-root user that was defined during the installation process.
2. Open a command prompt window.
3. Go to the `$COLLATION_HOME/etc` directory.
4. Modify the `collation.properties` file and update the following parameters:
   - `com.collation.db.password`
   - `com.collation.db.archive.password`

   Replace the encrypted password string with the new password in clear text.

5. Save the file.
6. Go to the directory `$COLLATION_HOME/support/bin`.
7. To encrypt the passwords, run **`encryptprops`** script giving as a parameter the `$COLLATION_HOME` directory:
   - For AIX, Linux, Linux on System z, and Solaris operating systems:

     ```
     ./encryptprops.sh $COLLATION_HOME
     ```

   - For Windows operating systems:

     ```
     encryptprops.bat %COLLATION_HOME%
     ```
8. Check the `collation.properties` file to ensure that the passwords are no longer plain text.
9. Go to the `$COLLATION_HOME/bin` directory.
10. Restart the TADDM server:
    - For AIX, Linux, Linux on System z, and Solaris operating systems:

      ```
      ./control restart
      ```

    - For Windows operating systems:

      ```
      control.bat restart
      ```

7.3 Discovery schedules

You can initiate TADDM discoveries on demand, based on a schedule, or through an API call. Scheduling is a methodical way to keep the TADDM database up-to-date.

7.3.1 Adding a discovery schedule

To add a discovery schedule, complete the following steps from the Product Console:

1. Select **Discovery** → **Schedule**. The Schedule page opens.
2. Click **Add**. The Discovery Schedule window opens. See Figure 7-2 on page 155.

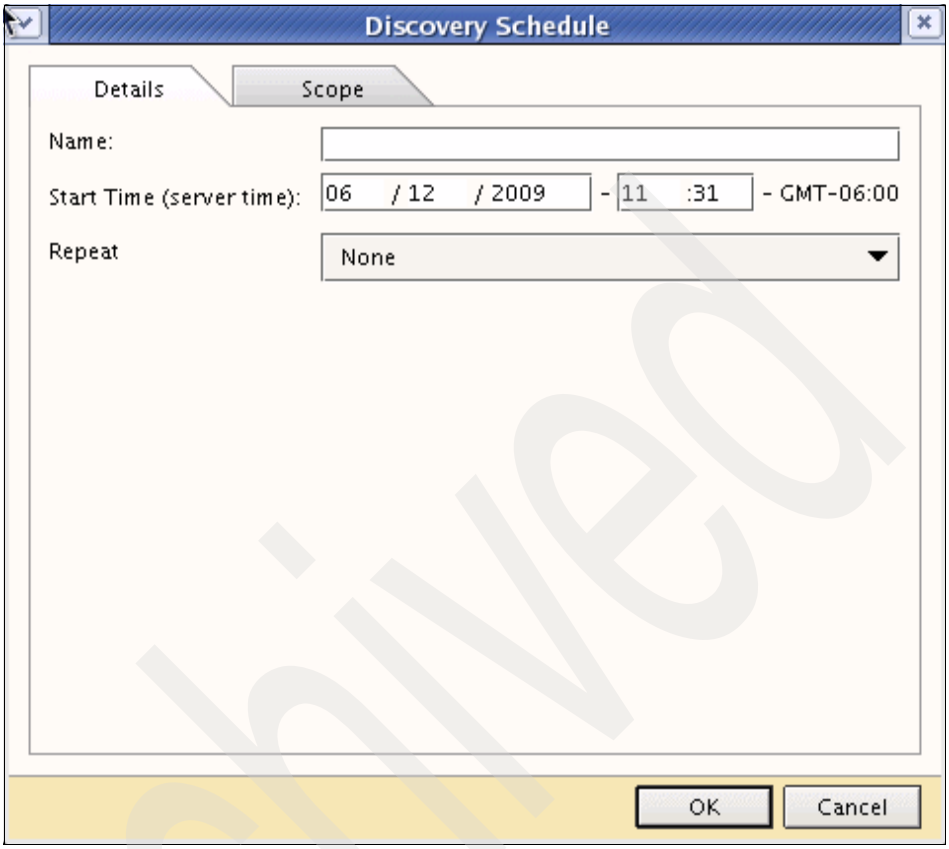

Figure 7-2 Discovery Schedule window

3. Click **Details** tab.
4. Enter a Name for the discovery schedule.
5. Specify a start time.
6. From the Repeat list, select the frequency that you want the discovery schedule to run.
7. In the Every field, type the numeric value for the time interval.
8. Click the **Scope** tab.
9. Select a Scope and a Profile for discovery.
10. Click **OK**. New discovery schedule appears on the Schedule list.

7.3.2 Viewing discovery schedule details

To view the details of a discovery schedule, complete the following steps from the Product Console:

1. Select **Discovery** → **Schedule**. The Schedule page opens.
2. Select a discovery schedule.
3. Click **Details**.
4. The Schedule Details window opens, as shown in Figure 7-3.

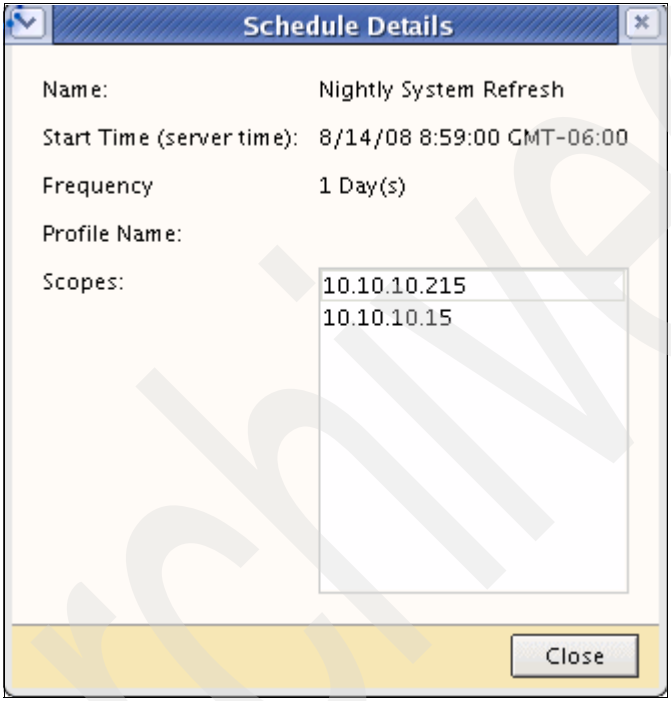

Figure 7-3 Schedule Details window

5. Close the Schedule Details window by clicking **Close**.

7.3.3 Deleting a discovery schedule

To delete a discovery schedule, complete the following steps from the Product Console:

1. Select **Discovery** → **Schedule**. The Schedule page opens.
2. Select a discovery schedule.
3. Click **Delete**.The Confirm Deletion window opens.
4. Click **Yes**. Schedule list appears updated.

7.4 Synchronization schedules

Synchronization is the method to update the TADDM Enterprise Domain Server with new domain data information. Like discoveries, only one synchronization can run at a given time for a domain. Also, synchronization fails if a discovery is running on the domain. Synchronizations can be full or incremental. The following list describes the full and incremental synchronizations:

- Full synchronization:

  Synchronization deletes all the configuration items that belong to a domain in the TADDM Enterprise Domain Server and synchronizes them again.

- Incremental synchronization:

  Only configuration items that have changed on the domain are synchronized to the TADDM Enterprise Domain Server.

Synchronization for a domain is always a full synchronization, and scheduled synchronizations are always incremental.

Note: You can use the TADDM Product Console to clear the entire topology for a domain. However, if the domain is part of a TADDM Enterprise Domain Server, you must perform full-synchronization in the TADDM Enterprise Domain Server to clear the domain data. The TADDM Enterprise Domain Server is not automatically aware that the topology has been cleared.

Synchronizations can be run on demand or they can be scheduled.

7.4.1 Adding a synchronization schedule

To add a synchronization schedule, complete the following steps from the Enterprise Domain Manager:

1. Select **Domain Management** → **Domain Summary**. The Domain Summary page opens. See Figure 7-4 on page 158.
2. Select a domain.

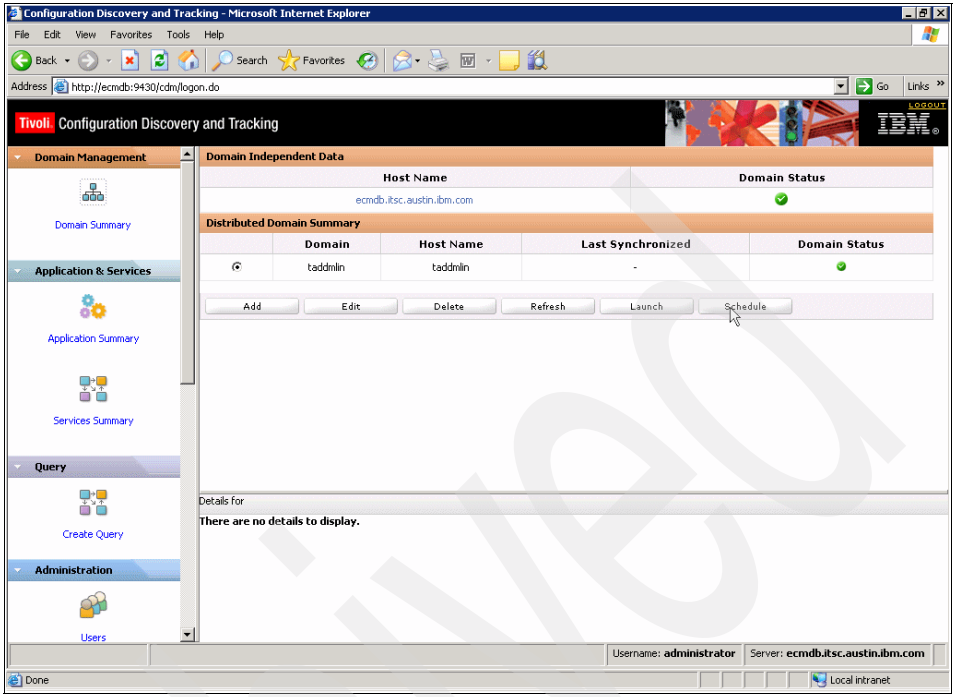

Figure 7-4 Domain Summary page

3. Click **Schedule**. The Synchronize Domain page opens.
4. In the Scheduled Synchronization section, click **Add**. The Schedule Period window opens.
5. Enter a name for the synchronization schedule.
6. Specify a start time.
7. From the Repeat list, select the frequency that you want the synchronization schedule to run.
8. In the Every field, type the numeric value for the time interval.
9. Click **OK**. New synchronization schedule appears on the Scheduled Synchronization list.

Note: To start the synchronization immediately click **Start,** or to stop the synchronization click **Stop**.

7.4.2 Viewing synchronization details

To view the details of a synchronization schedule, complete the following steps from the Enterprise Domain Manager:

1. Select **Domain Management** → **Domain Summary**. The Domain Summary page opens.
2. Select a domain.
3. Click **Schedule**. The Synchronize Domain page opens.
4. In Last Synchronization Times section, click **View Sync Details**.
5. Synchronization status is displayed for the specified domain.

7.4.3 Deleting a synchronization schedule

To delete a synchronization schedule, complete the following steps from the Enterprise Domain Manager:

1. Select **Domain Management** → **Domain Summary**. The Domain Summary page opens.
2. Select a domain.
3. Click **Schedule**. The Synchronize Domain page opens.
4. In the Scheduled Synchronization section, select a synchronization schedule.
5. Click **Delete**.
6. Scheduled Synchronization list appears updated.

7.5 Versions

A *version* is a snapshot of the current infrastructure. Versions are read-only views of the entire topology. Analytic reports support comparisons between versions. The discovered data for the new version is in the TADDM database and is stored under the archive user schema, which is the archive user that is specified during the TADDM installation.

The reasons for creating versions in TADDM are:

- Disaster recovery

  Create a version before any major IT infrastructure change, such as a new IT asset added into your current IT environment or application deployment, from development and testing environment to production environment. You can

use the configuration information in TADDM in case any recovery is necessary after a major IT infrastructure change.

- IT infrastructure changing

  If you have any IT environment structure changing, such as adding or removing a group of hardware or software resources or re-deploying an application to different environments, then the topology is different after the change. Before the change, create a version that saves the topology data for later usage.

- Audit reports

  If you need security audit and compliance reports, you can create versions for the audit and compliance check.

Versions are different from change history. The following list contains the main differences between versions and the change history:

- Comparison shows only the differences between two components, not all the intermediate configuration changes.
- Versions are in the TADDM database until you delete them.
- Change History for a component is deleted along with the component.
- Clearing the topology clears all components and the corresponding change history for those components, but it does not delete versions.

7.5.1 Adding a version

To add a version, complete the following steps from the Product Console:

1. Select **Discovery** → **Versions**. The Versions page opens.
2. Click **Create**. The Create Version window, as shown in Figure 7-5, opens.

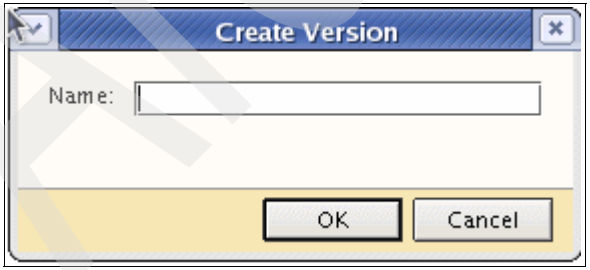

Figure 7-5 Create Version window

3. Enter a name for version.
4. Click **OK**. A new version appears on the Versions list.

7.5.2 Viewing a version

To view a version, complete the following steps from the Product Console:

1. Select **Discovery** → **Versions**. The Versions page opens.
2. Select a version.
3. Click **View**. A new Product Console for the version appears.

7.5.3 Deleting a version

To delete a version complete the following steps from the Product Console:

1. Select **Discovery** → **Versions**. The Versions page opens.
2. Select a version.
3. Click **Remove**.The Confirm Deletion window opens.
4. Click **Yes**. Versions list is updated.

7.6 Manual component creation

In addition to automatically discovering components through the provided discovery or custom server templates, you can manually add and identify components to an infrastructure topology. Adding components manually is very useful when creating a custom server template for the component is not possible. For example, you might have to add a specialized server, such as a mainframe system, that does not get automatically discovered.

7.6.1 Adding a component

To add a component manually, complete the following steps from the Product Console:

1. From the menu bar select **Edit** → **Create Component** or from the toolbar click **Create a new component** icon. The Create Component Wizard window opens. See Figure 7-6 on page 162.

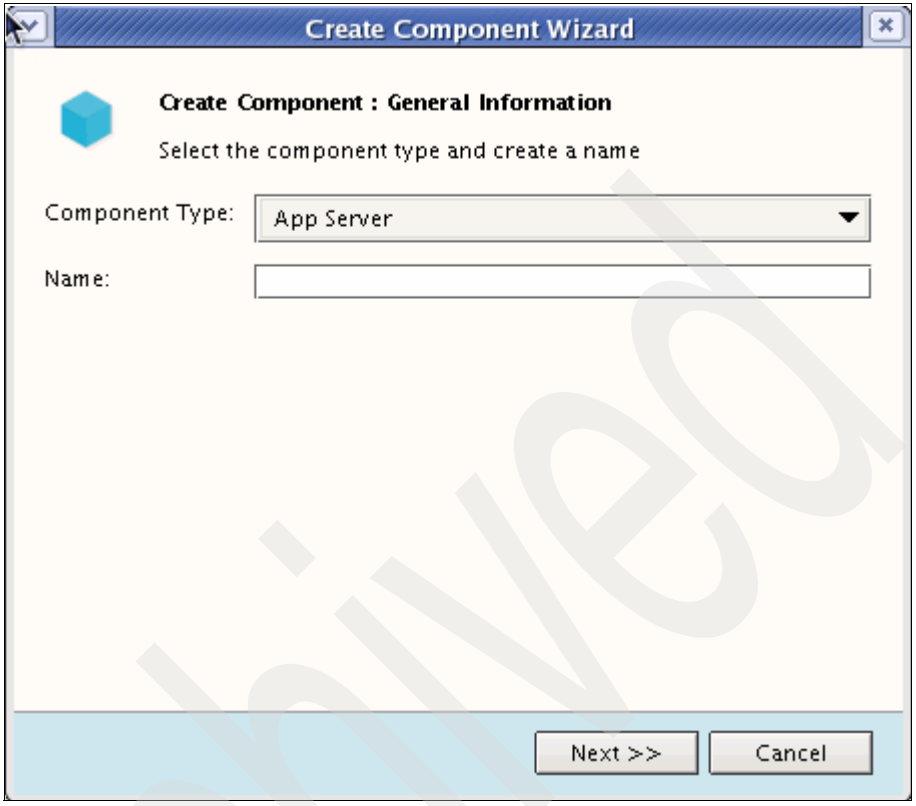

Figure 7-6 Create Component Wizard window

2. On the General Information page, select a component type:
   - App Server
   - Cluster
   - Computer System
   - Database
   - Firewall
   - J2EE Server
   - Legacy system
   - Load Balancer
   - Router
   - Service
   - Switch
   - Web Server
3. Type a name.
4. Click **Next**. The Admin Information page opens.

5. Optionally fill in the Admin Information fields.
6. Click **Finish**.
7. Reload the view. New component appears on the related component type list.

7.6.2 Editing a component

To edit a component, manually complete the following steps from the Product Console:

1. Select **Discovered Components** → **List/Search** → **<related component type>**. The component type list opens.
2. Select a component.
3. Right-click, and select **Edit.** The Edit Component window opens.
4. Edit the Admin Info fields.
5. Click **OK**. Component appears updated on the list.

7.6.3 Deleting a component

To delete a component, manually complete the following steps from the Product Console:

1. Select **Discovered Components** → **List/Search** → **<related component type>**. The component type list opens.
2. Select a component.
3. Right-click, and select **Delete.** The Delete Items window opens.
4. Click **OK**.
5. Reload the view. Component list appears updated.

7.7 Manual dependency creation

You can create a dependency for when the server does not automatically discover a dependency relationship that you know exists between components. For example, the server might not capture dependencies such as FTP sessions that are running when the discovery is run. During a subsequent discovery, dependencies that you create manually are maintained.

7.7.1 Adding a dependency

To add a dependency, manually complete the following steps from the Product Console:

1. Select **Discovered Components** → **List/Search** → **<related component type>**. The component type list opens.

2. Select a component.

   Note: The component can be either the dependent or provider component.

3. Right-click, and select **Component Dependencies.** The Dependencies List window opens. See Figure 7-7.

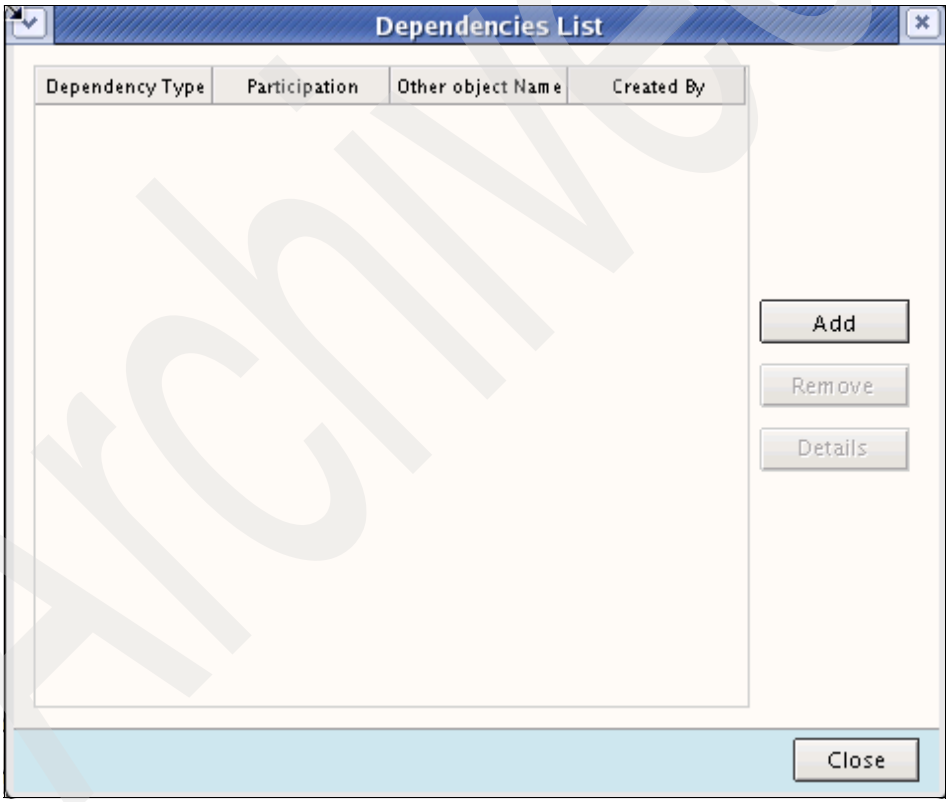

Figure 7-7 Dependencies List window

4. Click **Add**. The Add Dependency window opens, as shown in Figure 7-8 on page 165.

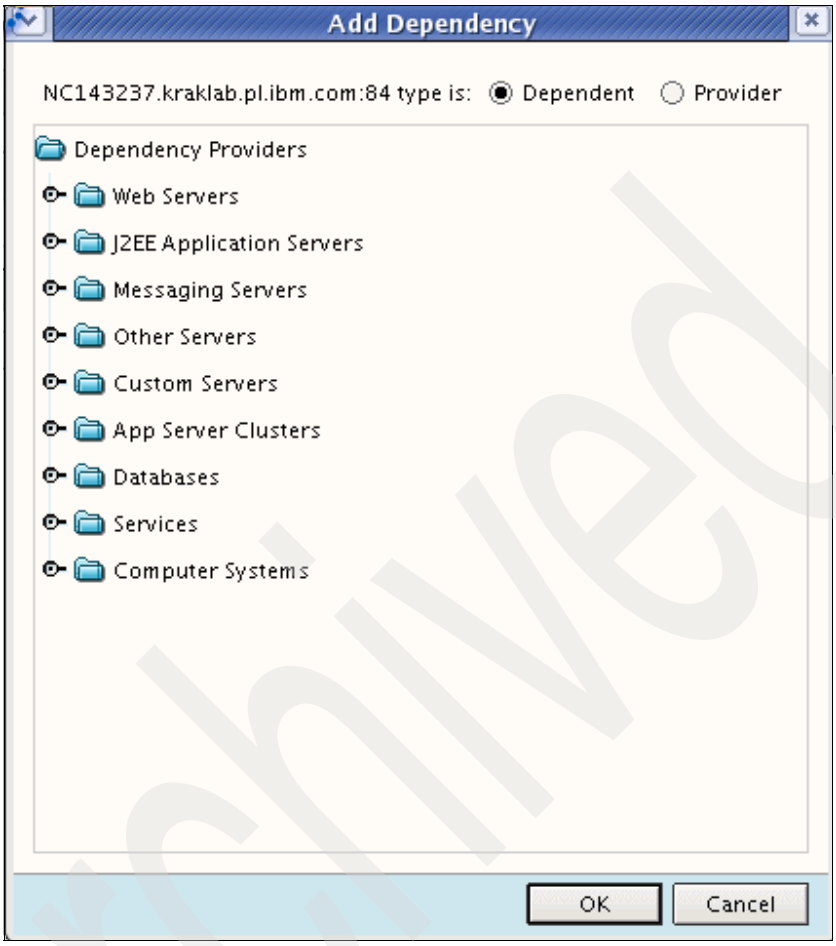

Figure 7-8 Add Dependency window

5. Select the dependency type from one of the following choices:
   - Select **Dependent** to make the selected component dependent on another component.
   - Select **Provider** to make the selected component a provider of information to another component.
6. Select the other component in the dependency relationship with the selected component.
7. Click **OK**. New dependency appears on the Dependencies list.
8. Click **Close**.
9. Reload the view. Component details become updated.

7.7.2 Viewing dependency details

To view the dependency details of a component, complete the following steps from the Product Console:

1. Select **Discovered Components** → **List/Search** → **<related component type>**. The component type list opens.
2. Select a component.
3. Right-click, and select **Component Dependencies.** The Dependencies List window opens.
4. Click **Close** to close the Dependencies List window.

7.7.3 Deleting a dependency

To delete a dependency manually, complete the following steps from the Product Console:

1. Select **Discovered Components** → **List/Search** → **<related component type>**. The component type list opens.
2. Select a component.

   Note: Component can be either the dependent or provider component.

3. Right-click and select **Component Dependencies.** The Dependencies List window opens.
4. Select a dependency, and click **Remove.** The Confirm Deletion window opens.
5. Click **Yes**. The Dependencies List opens updated.
6. Click **Close** to close the Dependencies List window.

7.8 Business applications and business services

A *business application* is a collection of components that is typically deployed and assigned a version number as a unit. Business applications typically represent business functions supported by the organization, for example Order Entry or Billing.

A *business service* is a collection of components integrated across multiple business applications and that delivers functionality for a specific customer task,

typically through a Web page. For example, Order Entry is a business service that can be delivered by integrating components from the Order Management, Inventory Management, and Billing business applications.

You can also create a business service to organize components servicing a particular organization or users in a geographic location. For example, you can create a business service that includes all databases and application servers used by the finance organization in a company.

You can simplify your infrastructure by combining large collections of individual components into groups. You can use these groups to view infrastructure resources as they are used by each business application or business service. Business applications and business services ease reporting and analysis of your infrastructure.

After you select the components that make up a business application or a business service, all component level dependencies are propagated to the application or the service topology, automatically highlighting all the application-level interdependencies.

You can subscribe business application configuration items to business services, but you cannot subscribe business service configuration items to business applications.

Business applications have an option to create a new functional group. Functional groups are useful when performing comparisons between two applications, because TADDM compares functional group instances with the same name. For example, Apache Web servers are compared only with other Apache Web servers. Functional groups also make it easier to manually create business applications. By selecting a specific functional group, a filtered list of components is automatically provided that can be subscribed to a business application.

7.8.1 Adding a business application or a business service

To add a business application or a business service, complete the following steps from the Product Console:

1. From the menu bar, select **Edit** → **Create Business Application** to add a business application, or select **Edit** → **Create Business Service** to add a business service. The Create Business Application Wizard or the Create Business Service Wizard window opens.
2. On the General Information page, type a name.
3. Optionally, type a description and a URL.

4. Click **Next**.The Create Business Application Components page opens. See Figure 7-9.

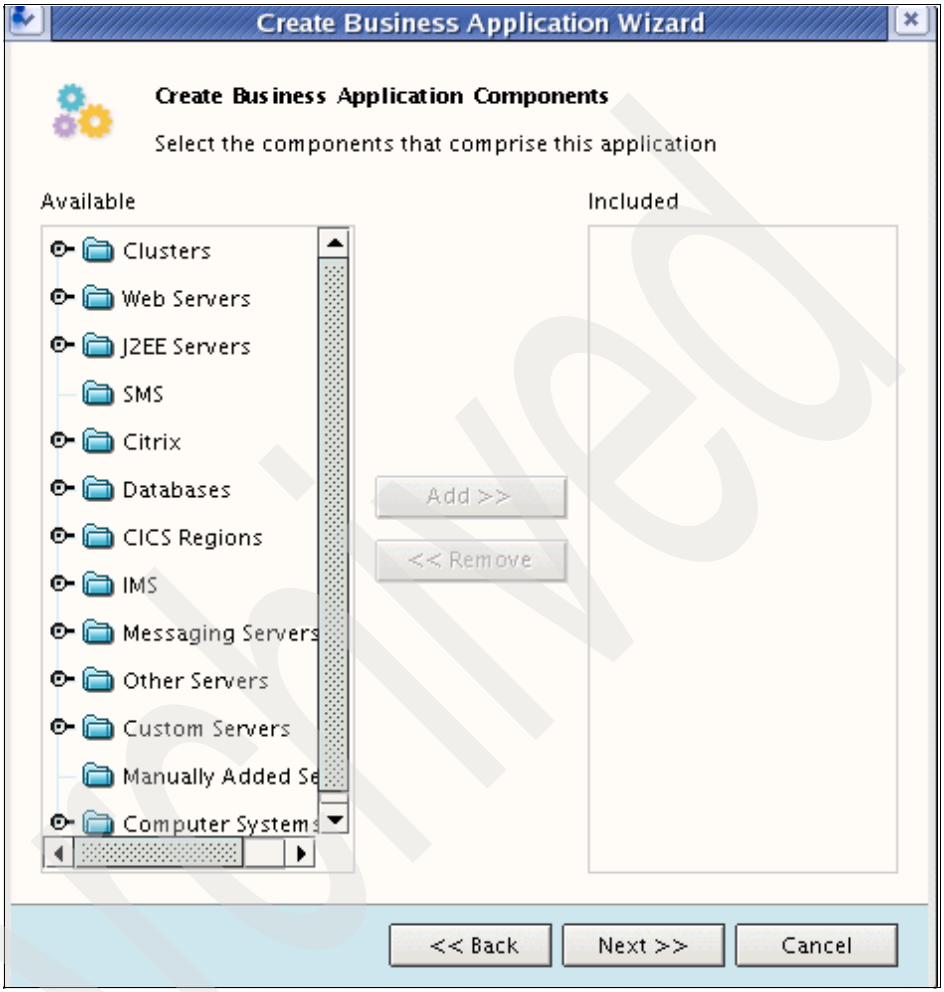

Figure 7-9 Create Business Application Wizard window

5. To add a component, select it on the Available list and click **Add**.
   To remove a component, select it on the Included list and click **Remove**.
6. Click **Next**.The Admin Information page opens.
7. Optionally, fill in the Admin Information fields.
8. Click **Finish**. A new business application or a business service appears on the **Discovered Components** → **Business Application** or **Discovered Components** → **Business Service** list.

7.8.2 Viewing business application or business service details

To view the details of a business application or a business service complete the following steps from the Product Console:

1. Select **Discovered Components** → **Business Application** for business applications, or **Discovered Components** → **Business Service** for business services. The Business Application or the Business Service list opens.
2. Select a business application or a business service.
3. Right-click and select **Show Details**.
4. The business application or business service details appear on the Details page.

7.8.3 Viewing business application or business service topology

To view the topology of a business application or a business service, complete the following steps from the Product Console:

1. Select **Discovered Components** → **Business Application** for business applications, or select **Discovered Components** → **Business Service** for business services. The Business Application or the Business Service list opens.
2. Select a business application or a business service.
3. Right-click and select **Show Software Topology** for the software topology or select **Show Physical Topology** for the physical topology. The business application or business service topology appears on the Business Applications page.

7.8.4 Editing a business application or a business service

To edit a business application or a business service, complete the following steps from the Product Console:

1. Select **Discovered Components** → **Business Application** for business applications, or **Discovered Components** → **Business Service** for business services. The Business Application or the Business Service list opens.
2. Select a business application or a business service.
3. Right-click and select **Edit.** The Edit Business Application Wizard or the Edit Business Service Wizard window opens.
4. On the General Information page, edit the description and the URL.
5. Click **Next**.The Create Business Application Components page opens.

6. Edit the business application or business service components. To add a component, select a component on the Available list and click **Add**.
7. To remove a component, select a component on the Included list and click **Remove**.
8. Click **Next**.The Admin Information page opens.
9. Edit the Admin Information fields.
10. Click **Finish**. Business application or business service is updated on the list.

7.8.5 Deleting a business application or a business service

To delete a business application or a business service, complete the following steps from the Product Console:

1. Select **Discovered Components** → **Business Application** for business applications, or **Discovered Components** → **Business Service** for business services. The Business Application or the Business Service list opens.
2. Select a business application or a business service.
3. Right-click and select **Delete.** The Delete Items window opens.
4. Click **OK**.
5. Reload the view. Business application or business service list is updated.

7.9 Roles and permissions

A *permission* authorizes the user to perform an action or access a specific configuration item. A *role* is a set of permissions that can be assigned to a user. Permissions can be:

- Read

  The user can view information about a configuration item.

- Update

  The user can view and modify information about a configuration item.

- Discover

  The user can initiate a discovery, create and update discovery scope objects, or create new objects from the Product Console Edit menu.

- Admin

  The user can create or update users, roles, and permissions. The user can also configure authorization policy with the authorization manager.

Permissions are aggregated into roles, and users are granted permissions by assigning them roles that have those permissions. Permissions are classified as data-level or method-level as follows:

- Data-level

  Read and update permissions

- Method-level

  Discover and admin permissions

Note: TADDM has three default roles:

- Operator: with read permission
- Supervisor: with read, update, and discover permissions
- Administrator: with read, update, discover, and admin permissions

When you assign a role to a user, you must specify one or more access collections for that role. This limits the scope of the role to only those access collections that are appropriate for that user.

7.9.1 Adding a role

To add a role, complete the following steps from the Domain Manager:

1. Select **Administration** → **Roles**. The Roles page opens.
2. Click **Create Role**. The Create Role window opens. See Figure 7-10 on page 172.

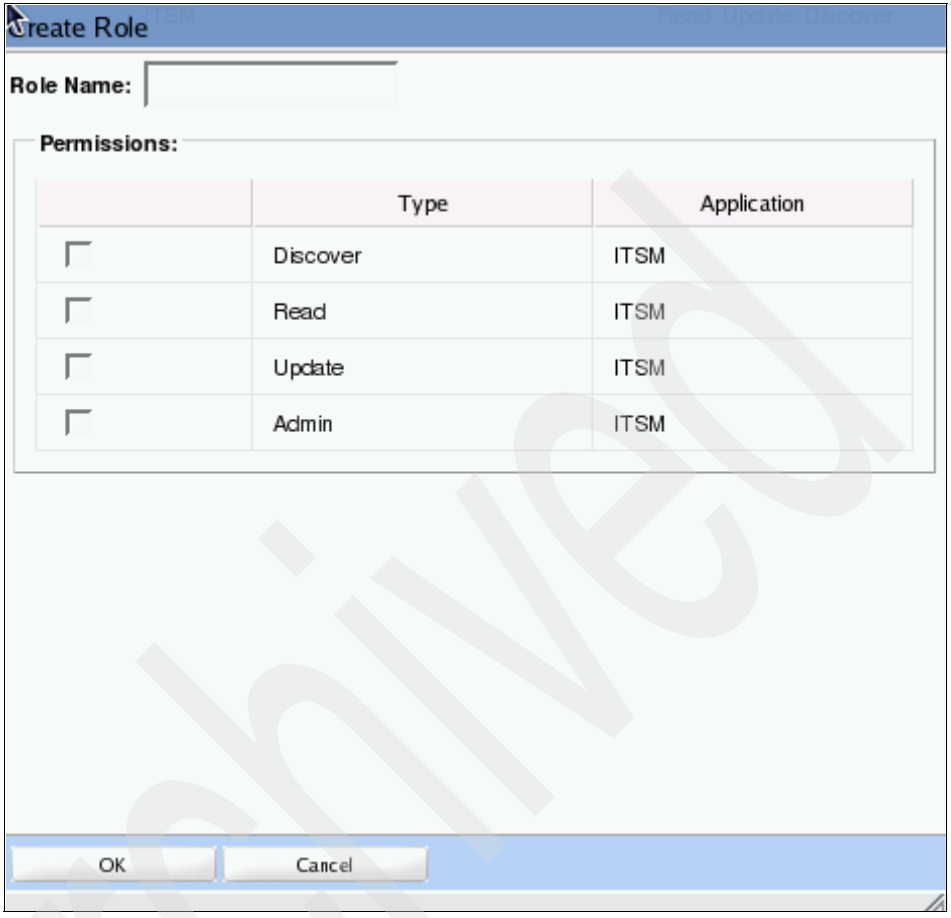

Figure 7-10 Create Role window

3. Enter a Role Name.
4. Select the Permissions check boxes to specify the permissions of the role.
5. Click **OK**. New role appears on the Roles list.

7.9.2 Deleting a role

To delete a role, complete the following steps from the Domain Manager:

1. Select **Administration** → **Roles**. The Roles page opens.
2. Select a role, and click **Delete**.
3. The Roles list appears updated.

7.10 Application programming interface

You can use the application programming interface (API) to access features of TADDM from the command line, enabling scripting, simple customizing, and scheduling. This section describes the commands.

For AIX, Linux, Linux on System z, and Solaris operating systems, the API is located in the `$COLLATION_HOME/sdk/bin/api.sh` path.

For Windows operating systems, the API is located in the following path:

```
%COLLATION_HOME%\sdk\bin\api.bat
```

The syntax for the API is:

```
api.sh -u|--u user -p|--p password [-H|--host host] [-P|--port port]
COMMAND COMMAND-PARAMETERS
```

The API parameters are:

- `-u|--u user`

  The user who is running the API command

- `-p|--p password`

  The password that authenticates the user

- `-H|--host host`

  The TADDM server host name, which by default is localhost

- `-P|--port port`

  The TADDM server port, which by default is 9530

- *`COMMAND`*

  You can use the API to issue the following commands:

  - find
  - discover
  - topology
  - changes
  - version
  - delete
  - import
  - export
  - naming

- *`COMMAND-PARAMETERS`*

  Parameters vary depending on the specific command.

7.10.1 Find command

The **find** command finds a set of topology objects and returns an XML representation.

Command syntax

The command syntax is shown in Example 7-2.

Example 7-2 Syntax: find command

```
api.sh -u|--u user -p|--p password [-H|--host host] [-P|--port port]
find [--depth depth] [--indent num-spaces] [--changetype type [--from
from-date [--end end-date] ] [-o --outfile local file to write to [-x
--maxfilesize size]] [-s--suppress list of classes to suppress] root

api.sh -u|--u user -p|--p password [-H|--host host] [-P|--port port]
find [--depth depth] [--indent num-spaces] --guid object-guid

api.sh -u|--u user -p|--p password [-H|--host host] [-P|--port port]
find [--depth depth] [--indent num-spaces] [--mssguid mss-guid |
--mssname mss-name] mql-query
```

Command parameters

Command parameters include:

- `-u|--u user`

  The user that is running the API command

- `-p|--p password`

  The password that authenticates the user

- `-H|--host host`

  The TADDM server host name, which by default is localhost

- `-P|--port port`

  The TADDM server port, which by default is 9530

- `find`

  Runs the **find** command

- `--depth depth`

  The level of the result tree to construct

- `--indent num-spaces`

  The indentation to use for the resulting XML output

- `--changetype type`

  The type of change, from among the following values:

  - 0: Created
  - 1: Updated
  - 2: Deleted
  - 3: Creates and updates
  - 4: All changes

- `--from from-date`

  The beginning date of the change parameter, using the following format:

  `mm/dd/yy hh:mm:ss AM|PM`

- `--end end-date`

  The end date of the change parameter, using the following format:

  `mm/dd/yy hh:mm:ss AM|PM`

- `-x |--maxfilesize size`

  The outfile can be wrapped into several smaller files by specifying the maximum file size in bytes. The output is split into several files under the maximum file size when possible.

- `-o|--outfile <local file to write to>`

  The name of the file to redirect the output of the find command to

- `-s|--suppress <list of classes to suppress>`

  The list of classes to be omitted from the find results

  The classes are model object name classes, such as ComputerSystem, OperatingSystem, and so on.

- `--guid object-guid`

  The GUID of the object for which the find command is being executed.

- `--mssguid mss-guid`

  The GUID of the Management Software System

- `--mssname mss-name`

  The name of the Management Software System

- `mql-query`

  The query specified using the Model Query Language (MQL), for example:

  `SELECT attributes FROM object type [WHERE expression]`

  You can use long or short names for the object types in this argument.

- root

  The model object to serve as the root for the resulting XML output. You can use long or short names for the object types in this argument.

Command examples

Examples of using the **find** command are:

- Save the result of find computer systems to the file `cs_output.xml` file:

  ```
  api.sh -u user -p password -H host -P port find -o cs_output.xml
  ComputerSystem
  ```

- Save the result of find computer systems to the file `cs_output.xml` file, each with a maximum file size of 1000 bytes:

  ```
  api.sh -u user -p password -H host -P port find -o cs_output.xml -x
  1000 ComputerSystem
  ```

- Find all computer systems where the result does not include OperatingSystem objects:

  ```
  api.sh -u user -p password -H host -P port find -s OperatingSystem
  ComputerSystem
  ```

7.10.2 Discover command

The **discover** command starts or stops a discovery run.

Command syntax

The command syntax is shown in Example 7-3.

Example 7-3 Syntax: discover command

```
api.sh -u|--u user -p|--p password [-H|--host host] [-P|--port port]
[-l logfile_name] discover start [--name run-name] scope-element1
scope-element2 ... scope-elementn

api.sh -u|--u user -p|--p password [-H|--host host] [-P|--port port]
[-l logfile_name] discover abort|status [-profile profile name]
scope-element1 scope-element2 ... scope-elementn
```

Command parameters

Command parameters include:

- `-u|--u user`

  The user that is running the API command

- `-p|--p password`

  The password that authenticates the user

- `-H|--host host`

  The TADDM server host name, which by default is localhost

- `-P|--port port`

  The TADDM server port, which by default is 9530

- `-l logfile_name`

  The location and name of the log file, which by default is:

  `$COLLATION_HOME/sdk/log/api-client.log`

- discover

  Runs the **`discover`** command

- `start scope-element1 scope-element2 ... scope-elementn`

  Starts a discover by using the specified scope elements

- `--name run-name`

  The user-assigned name of the discovery run

- `abort`

  Stops a running discovery on the specified host

- `status`

  Returns the discovery status on the specified host, from among the following values:

  - Running
  - Idle
  - Waiting
  - Aborted

- `profile scope-element1 scope-element2 ... scope-elementn`

  Uses the profile specified by <`profile name`> for the discovery

Command examples

Examples of using the **discover** command are:

- Discover the subnet 10.10.10.0/24:

```
api.sh -u user -p password -H host discover start
“10.10.10.0/255.255.255.0”
```

- Get the discover status:

```
api.sh -u user -p password -H host -P port discover status
```

7.10.3 Topology command

The **topology** command clears or rebuilds the topology.

Command syntax

The command syntax is shown in Example 7-4.

Example 7-4 Syntax: topology command

```
api.sh -u|--u user -p|--p password [-H|--host host] [-P|--port port]
[-l logfile_name] topology clear|rebuild
```

Command parameters

Command parameters include:

- `-u|--u user`

  The user that is running the API command

- `-p|--p password`

  The password that authenticates the user

- `-H|--host host`

  The TADDM server host name, which by default is localhost

- `-P|--port port`

  The TADDM server port, which by default is 9530

- `-l logfile_name`

  The location and name of the log file, which by default is:

  `$COLLATION_HOME/sdk/log/api-client.log`

- `topology`

  Runs the **topology** command

- `clear`

  Clears the existing topology

- `rebuild`

  Rebuilds the topology

Command examples

Examples of using the **`topology`** command are:

- Clear the topology:

```
api.sh -u user -p password -H host -P port topology clear
```

- Clear the existing topology:

```
api.sh -u user -p password -H host -P port topology rebuild
```

7.10.4 Changes command

The **`changes`** command retrieves the changes for a topology object.

Command syntax

The command syntax is shown in Example 7-5.

Example 7-5 Syntax: changes command

```
api.sh -u|--u user -p|--p password [-H|--host host] [-P|--port port]
[-l logfile_name] changes guid from-date [to-date]
```

Command parameters

Command parameters include:

- `-u|--u user`

  The user that is running the API command

- `-p|--p password`

  The password that authenticates the user

- `-H|--host host`

  The TADDM server host name, which by default is localhost

- `-P|--port port`

  The TADDM server port, which by default is 9530

- `-l logfile_name`

  The location and name of the log file, which by default is:

  `$COLLATION_HOME/sdk/log/api-client.log`

- `changes`

  Runs the **changes** command

- `guid`

  The GUID of the object for which you want to determine changes

- `from-date`

  The beginning date of the change command, using the following format:

  `mm/dd/yy hh:mm:ss AM|PM`

- `to-date`

  The end date of the change command, using the following format:

  `mm/dd/yy hh:mm:ss AM|PM`

Command example

An example of using the **changes** command includes finding all changes on an object that occurred between two specific dates, as follows:

```
api.sh -u $user -p password -H host -P port changes
10A5794E86C53A0BBB10F262055CB3EA "06/06/05 12:00:00 AM" "06/08/05
12:00:00 AM"
```

7.10.5 Version command

The **version** command manages versions in the TADDM.

Command syntax

The command syntax is shown in Example 7-6.

Example 7-6 Syntax: version command

```
api.sh -u|--u user -p|--p password [-H|--host host] [-P|--port port]
[-l logfile_name] version [-c|--create version-name
version-description] [-e|--createempty version-name
version-description] [-d|--delete version-id-or-name]

api.sh -u|--u user -p|--p password [-H|--host host] [-P|--port port]
[-l logfile_name] version getall
```

Command parameters

Command parameters include:

- `-u|--u user`

  The user that is running the API command

- `-p|--p password`

  The password that authenticates the user

- `-H|--host host`

  The TADDM server host name, which by default is localhost

- `-P|--port port`

  The TADDM server port, which by default is 9530

- `-l logfile_name`

  The location and name of the log file, which by default is:

  `$COLLATION_HOME/sdk/log/api-client.log`

- `version`

  Runs the **`version`** command

- `-c|--create version-name version-description`

  Creates a new version using the supplied name

- `-e|--createempty version-name version-description`

  Creates an empty new version using the supplied name

- `-d|--delete version-id-or-name`

  Deletes the specified version

Command examples

Examples of using the **`version`** command are:

- Create a version:

```
api.sh -u user -p password -H host -P port version -create
"version1.0" "This is the initial version"
```

- Delete a version:

```
api.sh -u user -p password -H host -P port version -delete
"version1.0"
```

7.10.6 Delete command

The **delete** command removes an object from the TADDM.

Command syntax

The command syntax is shown in Example 7-7.

Example 7-7 Syntax: delete command

```
api.sh -u|--u user -p|--p password [-H|--host host] [-P|--port port]
[-l logfile_name] delete guid
```

Command parameters

Command parameters include:

- `-u|--u user`

  The user that is running the API command

- `-p|--p password`

  The password that authenticates the user

- `-H|--host host`

  The TADDM server host name, which by default is localhost

- `-P|--port port`

  The TADDM server port, which by default is 9530

- `-l logfile_name`

  The location and name of the log file, which by default is:

  `$COLLATION_HOME/sdk/log/api-client.log`

- `delete`

  Runs the **delete** command

- `guid`

  The GUID of the object to delete

Command example

An example of using the **delete** command includes deleting an object with the specified GUID, as follows:

```
api.sh -u user -p password -H host -P port delete
10A5794E86C53A0BBB10F262055CB3EA
```

7.10.7 Import command

The **import** command imports data into the TADDM.

Command syntax

The command syntax is shown in Example 7-8.

Example 7-8 Syntax: import command

```
api.sh -u|--u user -p|--p password [-H|--host host] [-P|--port port]
[-l logfile_name] import [-T|--topo] [--timeout time] [--mssguid
mss-guid|--mssname mss-name] [--maxfilesize size]
local-directory-to-read-data-from
```

Command parameters

Command parameters include:

- `-u|--u user`

  The user that is running the API command

- `-p|--p password`

  The password that authenticates the user

- `-H|--host host`

  The TADDM server host name, which by default is localhost

- `-P|--port port`

  The TADDM server port, which by default is 9530

- `-l logfile_name`

  The location and name of the log file, which by default is:

  `$COLLATION_HOME/sdk/log/api-client.log`

- `import`

  Runs the **import** command

- `-T | --topo`

  Rebuilds the topology after the import operation is complete

- `--timeout time`

  The timeout value, which is useful for very large file imports (in seconds)

- `--mssguid mss-guid`

  The GUID of the Management Software System with which the imported data is to be associated

- `--mssname mss-name`

  The name of the Management Software System with which the imported data is to be associated

- `local-directory-to-read-data-from`

  The name of the directory from which the data is to be imported

Command example

An example of using the **`import`** command includes importing data into the TADDM, as follows:

```
api.sh -u user -p password -H host import directory/
```

The command attempts to import all files in the specified directory. If the command encounters an invalid XML file, an exception is thrown but the command continues importing until all files have been read.

7.10.8 Export command

The **`export`** command exports data for top-level model objects in the TADDM.

Command syntax

The command syntax is shown in Example 7-9.

Example 7-9 Syntax: export command

```
api.sh -u|--u user -p|--p password [-H|--host host] [-P|--port port]
[-l logfile_name] export [--mssguid mss-guid|--mssname mss-name]
[--maxfilesize size] local-directory-to-write-data
```

Command parameters

Command parameters include:

- `-u|--u user`

  The user that is running the API command

- `-p|--p password`

  The password that authenticates the user

- `-H|--host host`

  The TADDM server host name, which by default is localhost

- `-P|--port port`

  The TADDM server port, which by default is 9530

- `-l logfile_name`

  The location and name of the log file, which by default is:

  `$COLLATION_HOME/sdk/log/api-client.log`

- `export`

  Runs the **`export`** command

- `--mssguid mss-guid`

  The GUID of the Management Software System. Only data associated with the specified MSS is exported.

- `--mssname mss-name`

  The name of the Management Software System. Only data associated with the specified MSS is exported.

- `--maxfilesize size`

  The maximum size of the exported files (in bytes)

- `local-directory-to-write-data`

  The name of the directory to which the data is to be exported

Command example

An example of using the **`export`** command includes exporting top-level model objects to the specified directory, as follows:

```
api.sh -u user -p password -H host export directory/
```

7.10.9 Naming command

The **`naming`** command returns one or more GUIDs associated with a configuration item (CI). Only the GUIDs of top level CIs in the XML file are returned.

Command syntax

The command syntax is shown in Example 7-10.

Example 7-10 Syntax: naming command

```
api.sh -u|--u user -p|--p password [-H host naming sample.xml ]
```

Command paramaters

Command parameters include:

- `-u|--u user`

  The user that is running the API command

- `-p|--p password`

  The password that authenticates the user

- `-H|--host host`

  The TADDM server host name, which by default is localhost

- `-P|--port port`

  The TADDM server port, which by default is 9530

- `-l logfile_name`

  The location and name of the log file, which by default is:

  `$COLLATION_HOME/sdk/log/api-client.log`

- `naming`

  Runs the **`naming`** command

- `-f`

  The location and name of the XML file containing the configuration item (model object)

Command example

An example of using the **`naming`** command includes displaying the one or more GUIDs for CIs in the XML file, as follows:

```
api.sh -u user -p password -H host naming sample.xml
```

A

Sample certification test questions

In this appendix, we provide sample questions that are representative of the ones you encounter on the actual certification test. We recommend you take this sample test after studying the chapters in this book.

This appendix contains:

- "Questions" on page 188
- "Answers" on page 193

Questions

The following questions can assist you in studying for the certification test:

1. Which of the following statements are true regarding the Tivoli Application Dependency Discovery Manager (TADDM) Domain Database (CMDB)? (Choose two.)
   a. The CMDB holds the discovered information about configuration items in CIM Database structure.
   b. An existing DB2 database located on another system or Oracle database located on another system can support a CMDB.
   c. The TADDM installation wizard does not install an Oracle database.
   d. The TADDM installation wizard can be used to install an Oracle database.
   e. An existing DB2 database located on another system or Oracle database located on another system cannot support a CMDB.
2. Where would you plan to use federated repositories?
   a. When authenticating against Tivoli Provisioning Manager components.
   b. To take advantage of the user and group management capabilities that it provides and to enable single sign-on (SSO) between Tivoli applications.
   c. When combining data from multiple Domain databases.
   d. When authenticating against file base repositories.
3. What does the term data federation refer to?
   a. An architecture where a relational database management system (RDBMS) enables access to heterogeneous data sources.
   b. An architecture where a relational database management system (RDBMS) enables access to security portals.
   c. An architecture where a Web-based application captures data feeds from a single data source.
   d. An architecture where a Web-based application sends data feeds to a single data source.

4. What types of user repositories does TADDM support? (Choose 3.)
   a. TADDM file-based repositories
   b. LDAP repositories.
   c. Federated repositories functionality of IBM WebSphere Application Server.
   d. XML based encrypted data stores.
   e. WMI based encrypted data stores.
   f. WQL based encrypted data stores.
5. Which of the following choices are valid installation scenarios? (Choose 3.)
   a. A simple installation with the installation of a DB2 database
   b. A simple installation without the installation of a DB2 database
   c. An advanced installation with a remote DB2 database. The DB2 database is installed prior to the installation of the TADDM Server.
   d. An advanced installation with a remote Oracle database. The Oracle database is installed automatically with the installation of the TADDM Server.
   e. A simple installation with the installation of a Sybase database
6. Where are the installation log files stored?
   a. <TADDM Installation directory>/installLogs
   b. <TADDM Installation directory>/Logs
   c. <TADDM Installation directory>/tmp/Logs
   d. <TADDM Installation directory>/etc/installLogs
7. On a UNIX TADDM server, which command can be used to determine a successful installation of TADDM?
   a. service collation status
   b. service taddm status
   c. service cdt status
   d. service inventory status
8. Which file contains a configuration setting to adjust the time allowed to start a new anchor server?
   a. collation.properties
   b. anchor.properties
   c. collation.settings
   d. anchor.settings

9. Which configuration parameter is used to control SSH sensor timeout?
   a. com.collation.discover.agent.SshAgent.timeout
   b. com.collation.discover.agent.SshSensor.timeout
   c. com.collation.discover.agent.Ssh.timeout
   d. com.collation.discover.agent.SshSession.timeout
10. Which user is allowed to run the bulk load program?
   a. The user used to start the server
   b. Any user
   c. nobody
   d. Any user in the staff group
11. On a UNIX TADDM server, which command is used to populate the CMDB from the output produced by the Discovery Library Adapter?
   a. bulkload.sh
   b. loaddla.sh
   c. loadidml.sh
   d. importdla.sh
12. Which type of discovery does not require credentials?
   a. Level 1
   b. Level 2
   c. Level 3
   d. Custom Level 2
13. Which of the following commands does the GenericServerSensor run? (Choose 3.)
   a. lsof -nP -i
   b. ps axww *
   c. netstat.exe -nao
   d. test.jy
   e. ln -s $COLLATION_HOME /tmp

14. During the discovery process, if a session cannot be established using SSH or WMI, what will happen?
    a. The discovery process will end.
    b. A SNMP sensor will be run.
    c. A JMX sensor will be run.
    d. The discovery process will be started again.
15. Which command can be used to retrieve the template definitions in XML output?
    a. api.sh -u <username> -p <password> find --depth=5 Template
    b. jdk.sh -u <username> -p <password> find --depth=5 Template
    c. loadidml.sh -u <username> -p <password> find --depth=5 Template
    d. template.jy -u <username> -p <password> find --depth=5 Template
16. Which command can be used to load scope sets from the a text file?
    a. loadscope.jy
    b. loadscopesets.jy
    c. bulkload.jy
    d. api.sh
17. What type of entries can exist in a scope definition? (Choose 3.)
    a. An IP range, which includes all IP addresses between the start and end
    b. A subnet defined by an IP address and a netmask
    c. A host name, which is an individual device
    d. A list of IP address and ports numbers, which related to the Nmap configuration
    e. Subnets defined by IP addresses, netmasks and SNMP OIDs
    f. A list of discovery profiles labels
18. A property in a configuration file improves readability of the logs by separating the logging into per-sensor log files. What is the correct setting of the property?
    a. com.collation.discover.engine.SplitSensorLog=true
    b. com.collation.discover.engine.SplitSensorLog=false
    c. com.collation.discover.engine.SplitLogSensor=true
    d. com.collation.discover.engine.SplitLogSensor=false

19. Which sensor is invoked first?
    a. PortScanSensor
    b. IPDeviceSensor
    c. IPRangeSensor
    d. GenericComputerSensor
20. Which sensor is enabled by default in all discovery profiles?
    a. StackScanSensor
    b. ConfidenceThresholdSensor
    c. ISSStackScanSensor
    d. StackSensor
21. Which is a valid authentication type that can be specified in an access list?
    a. password
    b. passphrase
    c. private key
    d. base64
22. What will be the results of the **`api.sh -u user -p password -H gazoo discover start 10.10.10.0/255.255.255.0`** command?
    a. Discover the subnet 10.10.10.0/24 and the “gazoo” host.
    b. Discover the subnet 10.10.10.0/24.
    c. Discover the first 24 address in the subnet 10.10.10.0 and the “gazoo” host.
    d. Discover the first 24 addresses in the subnet 10.10.10.0.

Answers

The correct answers to the sample questions in this appendix are:

1. b, c
2. b
3. a
4. a, b, c
5. a, b, c
6. a
7. a
8. a
9. a
10. a
11. c
12. a
13. a, b, c
14. b
15. a
16. a
17. a, b, c
18. a
19. c
20. a
21. a
22. b

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 196. Note that some of the documents referenced here may be available in softcopy only.

- *Deployment Guide Series: IBM Tivoli Application Dependency Discovery Manager V7.1*, SG24-7616
- *IBM Tivoli Application Dependency Discovery Manager Capabilities and Best Practices*, SG24-7519

Online resources

These Web sites are also relevant as further information sources:

- IBM Professional Certification Program:

  http://www.ibm.com/certify/index.shtml
- Test 000-011 objectives:

  http://www.ibm.com/certify/tests/obj011.shtml
- IBM Tivoli Application Dependency and Discovery Manager V7.1 resources:

  http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/index.jsp?topic=/com.ibm.taddm.doc_7.1/cmdb_welcome.html
- OPAL Web site:

  http://www.ibm.com/software/brandcatalog/portal/opal
- TADDM Discovery Library Adapter Developer's Guide:

  http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/cmdb_dladevguide.pdf

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

access collection 123
  creating 124
  DefaultAccessCollection 124
  deleting 125
  editing 125
active troubleshooting situations 138
adding a domain 130, 132
  Admin Details 130
    Contact 131
    Escalation Contact 131
    Name 131
    Notes 131
  Domain Details 130
    Domain Name 131
    Domain Password 131
    Fully Qualified Host Name/IP 131
    Listening Port 131
addmPortMap. 78
Admin Details section 130
agentless discoveries 106
AL32UTF8 character set 55
anchors and gateways 77
  adding 79
  deleting 80
  editing 80
  setting an anchor port 80
Apache 12
aports.dll 78
application descriptors 88
  base application descriptor 89
  component application descriptor 91
  locations 93
    Apache 94
    Custom Server 95
    DB2 94
    Domino Server 94
    iPlanet 94
    JBoss 94
    Microsoft Exchange Server 94
    Microsoft IIS 94
    Oracle 94
    SQLServer 94
    Sybase/Sybase IQ 94
    Veritas Cluster 95
    WebLogic 94
    WebSphere 93
application programming interface (API) 173
application templates 86
  adding 86
  deleting 88
  editing 88
audit trail 136

B

base application descriptor 89
Bitvise 20
BSM (business service management) 126
bulkload 28
business applications 13, 89, 126, 166
  adding 167
  Admin Information 127
  components 127
  definition 126
    Container level 126
    DB schema 126
    Deep module level 126
  deleting 170
  dependencies 127
    dependencies to other business applications 127
    dependencies to other business services 127
    Inter-component 127
  viewing 169
business services 126, 166
  adding 167

C

certificate set up 38
certification
  benefits 3
  checklist 5
  IBM Professional Certification Program 2
  process 7
change a password 146

changes command 179
changing a domain 130
collation.properties 138, 153
collecting data for IBM 139
com.collation.db.archive.password 153
com.collation.db.password 153
com.collation.discover.topopumpcount 145
com.collation.jini.unicastdiscoveryport 131
com.collation.sslpassphrase 131
Common Data Model (CDM) 41
Common Information Model (CIM) 41
Component Dependencies 164
Component Type 162
ComputerSystem sub-model 42
configuration 67
  Access list 73
    adding an access list entry 74
    changing the order 75
    deleting an access list entry 75
  discovery scopes 69
    Adding a scope 70
    Adding a scope set 70
    Deleting a scope 71
    Deleting a scope set 70
    Editing a scope 71
    Loading a scope set from a file 71
    loadscope command 71
    loadscope command options 72
  performance tuning for databases 68
  Query optimizer 69
  RUNSTATS command 68
configuration item (CI) 12–13
configure sudo access 143
control command 150
control command. 150
control restart 154
courses 8
custom server templates 81, 114
  adding a custom server 81
  changing the order 86
  copying 85
  deleting 85
  editing 85
custom servers 113, 119
  adding 114
  creating 114
  Custom Servers window 113
  deleting 118
  displayed in the topology, 113
  editing 118
  managing 113
  repositioning entries 119
CustomAppServer object 122
Cygwin SSH 20

D

database buffer pool size 13
database deadlocks 145
database server discoveries 123
Datacenter 19
DB2 server on separate system 49
deep discovery 34
default passphrase 38
deleting a domain 132
disaster recovery 159
Discover Engine 30
Discover Observer 30
discovery components
  Discover Engine 30
  Discover Observer 30
  Java Space 30
  Process Flow Manager 30
  sensor 30
  Topology Builder 31
discovery history 107, 112
  viewing 111
discovery profiles 76, 120–121
  creating 121
  deleting 77
  editing 77
  level 1 76
  level 2 76
  level 3 76
  types
    Credential-less 76
    Full credential 76
    OS credentials only 76
discovery schedules 107
  adding a discovery schedule 154
  creating 108
    binding the current scope 107
    Components 109
    Details tab 108
    Schedule 108
    Scope Elements 109
  deleting 156
  overlap 111

running a basic discovery 110
synchronization schedules 157
adding 157
deleting 159
Full synchronization 157
Incremental synchronization 157
viewing synchronization details 159
viewing 110
viewing discovery schedule 156
displaying topology information 133
Distributed Management Task Force (DMTF) 41
DNS (Domain Name System) 41
Domain Details section 131
Domain Manager 127
domainquery.shallow file 128

E

EAR files 126
editing a domain 132
Save Changes 131
Test Connection 131
encryptprops script 154
Enterprise Domain Manager 127
Console 128
database 128
modes 128
Deep Mode 128
shallow mode 128
Distributed Domain Summary section 129
Domain Summary 129
overview 127
starting 129
enterprise inventory information 133
Enterprise JavaBeans (EJBs) 42
export command 184

F

Find command 174
findRelationships API query 148
firewall zone 20
Fully Qualified Domain Names (FQDN) 141

G

gateways 77

H

heap dump 140
heap dump files 140

I

IBM Certification Agreement 6
IBM Common Data Model (CDM)
configurations 42
dependencies 42
containment 42
service 42
service dependencies 42
transactional 42
signature 42
IBM WebSphere Application Server Federated Repositories 27
import command 183
Information Technology Infrastructure Library (ITIL) 13
Inventory Results page 114
IP routing table 42
iPlanet 12

J

javacore 140
JDBC (Java Database Connectivity) 42
jdbc connection 144
JDBC connection pools 42
Jetty HTTPS server 146
jini service 136
JMS (Java Message Service) 42
JMS topic queue 42
jnlp file 146
JSPs (JavaServer Pages) 42
jvmargs heapsize 140

L

large scopes 145
LDAP user registry 60
Lightweight Directory Access Protocol (LDAP) 103
loadscope 72
localhost 173
LOG4J loggers 139
logging
log files 136
error.log 136
local-anchor*.log 136
services/DiscoverManager.log 136
services/TopologyManager.log 136

tomcat.log 136
valid log levels 138
DEBUG 138
ERROR 138
FATAL 138
INFO (default) 138
TRACE 138
WARN 138
logical connections 42
long term troubleshooting 138
lsof (LiSt Open Files) utility 22, 106, 144

M

ManagedElement 45
manual component creation 161
adding a component 161
deleting a component 163
editing a component 163
manual dependency creation 163
adding a dependency 164
dependency types
Dependent 165
Provider 165
memory issues 140
Microsoft Windows Server 2003 18
ModelObject 45

N

name resolution 141
naming command 185
Network Mapper (Nmap) 21, 65
NFS (Network File System) 41
Nmap 21
non-ASCII characters 57

O

Open Source Nmap application 21
OpenSSH 32
Oracle DB 126
out-of-memory conditions 140

P

performance impact 138
performance tuning 68
permissions 170
Admin 170
Discover 170
Read 170
Update 170
ping -s 141
platform release 14
port 22 65
PortScan seed 35
primary user ID 49
Process Flow Manager 30
production environment 48–49
Prometric 6

R

raw sockets 143
Recurse Directory Content 117
Red Hat Enterprise Linux 5.0 16
Redbooks Web site 196
Contact us xv
return on investment (ROI) 5
rmi.clientproxy.server.hostname 146
roles 171
adding 171
Administrator 171
default roles 171
deleting 172
Operator 171
Supervisor 171
running discoveries 106
scope 106
RUNSTATS command 68
runtime relationships 42

S

sample questions 187
search path for capture file 117
secondary user ID 49
secure network connections 141
Secure Shell (SSH) 20
Security 96
configuring for LDAP 99
file authentication 96
configuring for WebSphere federated repositories 100
create a user 96
delete a user 99
edit a user 98
sensor configuration 122
Server Equivalent (SE) 12
Server Message Block 24

service name 140
services 126
setupAix.bin 62
setupLinux.bin 62
setupLinux390.bin 63
setupSolarisSparc.bin 63
setupWin32.exe 63
SnmpMib2Sensor seed 36
Solaris 10 SPARC 16
Solaris 9 SPARC 16
split logging feature 137
SplitSensor logging 138
sshd daemons 65
sslpassphrase 146
StackScan sensor 143
starting the TADDM server manually 150
Subnet 71
sudo access 32, 65
sudoers 143
SUSE Linux Enterprise Server 18
synchronization 157

T

TADDM
  Common Data Model 41
  discovery
    controlling 39
    discovering MySQL database 38
    discovering OpenVMS 39
    discovering WebSphere 37
    Level 1 discovery 31
    Level 2 discovery 32
    Level 3 discovery 34
    other discovery profiles 39
    sensor flow 34
  discovery components 29
    Discover Engine 30
    Discover Observer 30
    Java Space 30
    Process Flow Manager 30
    Sensor 30
    Topology Builder 31
  planning 11
    anchor server OS and hardware 19
    browser support 20
    Directory Services integration 27
    Discovery Library Adapters 28
    federating with eCMDB 25
    hardware prerequisites 12
    LDAP configuration 28
    maximum number of configuration items (CI) 27
    operating system prerequisites 13
    requirements analysis 22
      application environment 25
      firewall considerations 23
      network environment 23
      server environment 22
    sensors and discovery 29
    sizing 25
    supported database versions 22
    Windows gateway and operating system 21
  Product Console 71, 85
  publications 8
  resources recommended for study 8
TADDM agent-free discovery 29
TADDM Domain Server 55
TADDM Enterprise Domain Server 103, 127, 157
TADDM information center 56
TADDM installation
  advanced installation 55
    tasks 48
  anchor and gateway installation 64
    anchor considerations 65
    gateway considerations 66
  Oracle database 55
  overview 48
  prerequisite tasks 49
    using a local DB2 database 49
    using a remote DB2 database 50
  Setup WebSphere Federation Server 58
  silent flag 62
  silent installation 62
    record option 62
  simple installation 50
    tasks 48
  specifying a location for the server 52
  TADDM Domain Server 55
  TADDM Enterprise Domain Server 55
TADDM installation overview 48
TADDM Product Console 111, 119, 157
TADDM relationships 147
  explicit 147
  implicit 147
TADDM server 12–13
  Domain 12
  restarting 151

stopping 151
testing the status 152
TADDM troubleshooting
access and discovery issues 141
change a password 146
collecting data for IBM 139
database connectivity 144
expired password 145
log files 136
mapping 144
memory issues 140
name resolution issues 141
relationships 147
SplitSensor logging 138
sslpassphrase 146
TADDM versions
adding a version 160
deleting a version 161
reasons to create
Audit reports 160
Disaster recovery 159
IT infrastructure changing 160
viewing a version 161
TaddmPortMap.exe 78
TaddmWmi.dll 78
TaddmWmi.exe 78
TaddmWmi.mof 78
TCP listening port 113
test questions 187
test the secure connection 142
testwasconnection.sh 142
the archive user 49
Tivoli Certification benefits 5
Tivoli environment 5
Topology Builder 30–31
topology command 178
topology object 179

U

unicast discovery 131
unicast discovery port 131
unknown server category 113
unknown server patterns 113
unknown servers 113
unzip utility 21
unzip verification for AIX 21
Use anonymous binding 61
user registry 48

V

valid log levels 138
version command 180

W

WAR files 126
Web-based courses 8
WebLogic 12
WebLogic server 126
WebSphere (WAS) 142
WebSphere discovery 142
WebSphere Federated Repositories user registry 103
WebSphere Federation Server 48
Windows gateway 66
Windows protocols 24
WinSSHD 66

(0.2”spine)
0.17”<->0.473”
90<->249 pages

Certification Study Guide Series: IBM Tivoli Application Dependency Discovery Manager V7.1

Helps you achieve TADDM V7.1 certification

Explains the certification path and prerequisites

Introduces sample test questions

This IBM Redbooks publication is a study guide for IBM Tivoli Application Dependency Discovery Manager (TADDM) V7.1 and is aimed at individuals who want to get an IBM Professional Certification for this product.

The IBM Tivoli Application Dependency Discovery Manager V7.1 Professional Certification, offered through the Professional Certification Program from IBM, is designed to validate the skills required of technical professionals who work in the implementation of the IBM Tivoli Application Dependency Discovery Manager V7.1 product.

This book provides a combination of theory and practical experience needed for a general understanding of the subject matter. It also provides sample questions that will help in the evaluation of personal progress and provide familiarity with the types of questions that you will encounter in the exam.

This publication does not replace practical experience, nor is it designed to be a stand-alone guide for any subject. Instead, it is an effective tool that, when combined with educational activities and experience, can be an extremely useful preparation guide for the exam.

For your convenience, we structure the chapters based on the sections of Test 000-011: IBM Tivoli Application Dependency and Discovery Manager V7.1 Implementation, such as Planning, Installation, and so on, so studying each chapter will help you prepare for one section of the exam.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7764-00 ISBN 0738433241