

Integrating Tivoli Products

Describes security, data, navigation, reporting, and task integration

Provides extensive samples with a scenario-based approach

Discusses Tivoli product interoperability



Budi Darmawan
Christina L Grimes
Christopher Frost
Hossam A Katory

Karim Badr	Jim Popovitch
Andre Jenie	Piotr A Boetzel
Giorgio Sommariva	Marco Celon
Rodrigo Carvalho Giometti	Roger Turner
Guilherme Steinberger Elias	Hyun Kuk Moon



International Technical Support Organization

Integrating Tivoli Products

December 2009

Archived

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

First Edition (December 2009)

This edition applies to the following product versions:

- ▶ IBM Tivoli Change and Configuration Management Database V7.1.1
- ▶ IBM Tivoli Asset Management for IT V7.1
- ▶ IBM Tivoli Service Request Manager V7.1
- ▶ IBM Tivoli Application Dependency Discovery Manager V7.1.2
- ▶ IBM Tivoli Unified Process Composer V7.1.0
- ▶ IBM Tivoli Provisioning Manager V7.1
- ▶ IBM Tivoli Monitoring V6.2.2
- ▶ IBM Tivoli Composite Application Manager for Transactions V7.1
- ▶ IBM Tivoli Composite Application Manager for Web Resources V6.2
- ▶ IBM Tivoli Netcool/OMNIBus V7.2.1 Fix Pack 3
- ▶ IBM Tivoli Business Service Manager V4.2.0.0
- ▶ IBM Tivoli Network Manager for IP V3.8
- ▶ IBM Tivoli Netcool/Impact V5.1
- ▶ IBM Tivoli Netcool/Webtop V2.2
- ▶ IBM Tivoli Workload Scheduler V8.5
- ▶ IBM Tivoli Usage and Accounting Manager V7.1.2
- ▶ IBM Tivoli Storage Productivity Center

© Copyright International Business Machines Corporation 2009. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	ix
Trademarks	x
Preface	xi
The team who wrote this book	xi
Become a published author	xv
Comments welcome	xvi
Part 1. Introduction	1
Chapter 1. Integration overview	3
1.1 Integration overview	4
1.2 Aspects of integration	5
1.3 Product coverage	6
1.4 Document organization	6
Chapter 2. Integration scenarios	9
2.1 Overview of the scenarios	10
2.2 Common elements	12
2.3 Lab configuration	13
Part 2. Implementation	15
Chapter 3. Product installation overview	17
3.1 Installation overview	18
3.1.1 Agent deployment	18
3.1.2 Resiliency and high availability	19
3.2 IBM Service Management products	19
3.2.1 IBM Service Management	20
3.2.2 IBM Tivoli Application Dependency Discovery Manager	22
3.2.3 IBM Tivoli Provisioning Manager	24
3.3 IBM Tivoli Monitoring family	28
3.4 IBM Tivoli Netcool installation overview	34
3.4.1 Netcool product versions	35
3.4.2 Configuration	35
3.4.3 Installation results	37
3.5 IBM Tivoli Workload Scheduler	39
3.6 IBM Tivoli Usage and Accounting Manager	41
3.7 IBM Tivoli Storage Productivity Center for Data	44

Chapter 4. Security integration	45
4.1 Background security concepts	46
4.1.1 Lightweight Directory Access Protocol	46
4.1.2 WebSphere federated repositories	48
4.1.3 External authentication	50
4.1.4 Single sign-on	51
4.2 Security setup overview	53
4.2.1 IBM Tivoli Directory Server implementation	54
4.2.2 Security setup considerations	54
4.2.3 Setting up LDAP authentication for federated repositories	55
4.2.4 Setting up single sign-on on multiple WebSphere cells	55
4.3 Integrated Solution Console setup	56
4.3.1 LDAP authentication setup	56
4.3.2 Single sign-on setup	67
4.4 Tivoli Process Automation Engine security setup	73
4.4.1 LDAP configuration in WebSphere Application Server	74
4.4.2 Single sign-on configuration in WebSphere Application Server	84
4.4.3 VMMSYNC configuration	87
4.5 IBM Tivoli Application Dependency Discovery Manager security setup	92
4.6 IBM Tivoli Netcool products LDAP configuration	95
4.6.1 IBM Tivoli Netcool/OMNIBus LDAP configuration	95
4.6.2 Configuring Tivoli Integrated Portal LDAP	103
4.6.3 IBM Tivoli Netcool/Impact LDAP Configuration	114
4.7 IBM Tivoli Monitoring	120
4.7.1 Configuring Tivoli Enterprise Portal Server to authenticate to an LDAP repository	120
4.7.2 Work-around for security problem	126
4.8 IBM Tivoli Storage Productivity Center	127
Chapter 5. Data integration	129
5.1 Data integration overview	130
5.1.1 Common Data Model	130
5.1.2 IDML data	132
5.2 Resource data integration	135
5.2.1 IBM Tivoli Integration Composer	136
5.2.2 Promoting Actual CIs to Authorized CIs	137
5.2.3 IBM Tivoli Provisioning Manager integration	142
5.2.4 IBM Tivoli Monitoring integration	150
5.2.5 IBM Tivoli Business Service Manager integration	155
5.2.6 IBM Tivoli Network Manager for IP integration	163
5.3 Event data integration	176
5.3.1 IBM Tivoli Netcool/OMNIBus and IBM Tivoli Monitoring	177
5.3.2 IBM Tivoli Netcool/OMNIBus and IBM Tivoli Service Request Manager	

integration	194
5.4 Reports integration	218
5.4.1 Tivoli Common Reporting	218
5.4.2 Importing the report package	221
5.4.3 Available reports for Tivoli Common Reporting.	227
5.5 Other data integration	229
Chapter 6. Navigation integration	231
6.1 Navigation integration overview	232
6.2 Building a target URL	232
6.2.1 IBM Tivoli Application Dependency Discovery Manager.	233
6.2.2 Tivoli Process Automation Engine-based products.	235
6.2.3 Tivoli Enterprise Portal	245
6.2.4 Tivoli Integrated Portal	249
6.2.5 IBM Tivoli Netcool/Webtop Active Event List	254
6.2.6 IBM Tivoli Netcool/Impact operator view.	255
6.2.7 IBM Tivoli Storage Productivity Center for Data	255
6.3 Launching out capabilities.	261
6.3.1 IBM Tivoli Application Dependency Discovery Manager.	261
6.3.2 IBM Service Management products	266
6.3.3 IBM Tivoli Monitoring.	277
6.3.4 IBM Tivoli Network Manager for IP	286
6.3.5 IBM Tivoli Business Service Manager.	296
6.3.6 IBM Tivoli Netcool/Webtop Active Event List	309
6.3.7 IBM Tivoli Storage Productivity Center for Data	318
Chapter 7. Self monitoring and management.	321
7.1 Self monitoring overview	322
7.2 IBM Tivoli Provisioning Manager agent.	322
7.2.1 Requirements	323
7.2.2 Installation	323
7.2.3 Configuration.	324
7.3 IBM Tivoli Business Service Manager agent.	326
7.3.1 Requirements	326
7.3.2 Installation and configuration	327
7.4 IBM Tivoli Netcool/OMNIBus monitoring agent	327
7.4.1 Requirements	327
7.4.2 Implementation	328
7.5 IBM Tivoli Application Dependency Discovery Manager.	328
7.5.1 Requirements	329
7.5.2 Implementation	329
7.6 IBM Tivoli Network Manager for IP monitoring	330
7.6.1 Requirements	330

7.6.2 Implementation	331
7.7 IBM Tivoli Workload Scheduler agent monitoring	331
7.7.1 Requirements	332
7.7.2 Configuration	332
7.8 IBM Tivoli Netcool/Impact self-monitoring agent	336
Part 3. Scenario walk-through	339
Chapter 8. Operational drill down	341
8.1 Scenario overview	342
8.2 Scenario setup	343
8.2.1 Defining a new dashboard workspace	344
8.2.2 Launching an IBM Tivoli Application Dependency Discovery Manager window within Tivoli Enterprise Portal	347
8.2.3 Launching IBM Tivoli Provisioning Manager	350
8.2.4 Launching IBM Tivoli Network Manager for IP	353
8.3 Sample walk-through	355
8.4 Summary of benefits	363
Chapter 9. Automated application management scenario	365
9.1 Scenario overview	366
9.2 Scenario setup	367
9.2.1 Setting the managed system name	367
9.2.2 Setting up the Common Event Console	369
9.2.3 Navigation from Tivoli Enterprise Portal	377
9.2.4 Tivoli Enterprise Portal to IBM Tivoli Application Dependency Discovery Manager integration specifics	381
9.2.5 IBM Tivoli Provisioning Manager integration specifics	384
9.3 Sample walk-through	387
9.4 Summary of benefits	402
Chapter 10. Executive IBM Tivoli Business Service Manager scenario	403
10.1 Scenario overview	404
10.2 Scenario setup	405
10.3 Sample walk-through	405
10.4 Summary of benefits	410
Chapter 11. Change and release management scenario	413
11.1 Scenario overview	414
11.1.1 Products involved	415
11.1.2 Adoption route	415
11.2 Scenario setup	417
11.3 Sample walk-through	418
11.4 Summary of benefits	438

Abbreviations and acronyms 439

Related publications 443

IBM Redbooks and IBM Redpaper publications. 443

Online resources 446

How to get IBM Redbooks publications 450

Help from IBM 450

Index 451

Archived

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

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®	Lotus®	Service Request Manager®
CICS®	Maximo®	Symphony™
DB2 Universal Database™	Netcool®	Tivoli Enterprise Console®
DB2®	NetView®	Tivoli®
Foundations™	OMEGAMON®	WebSphere®
IBM®	Rational®	z/OS®
IMS™	RDN®	z/VM®
Informix®	Redbooks®	
Lotus Foundations™	Redbooks (logo)  ®	

The following terms are trademarks of other companies:

PostScript, and Portable Document Format (PDF) are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

AMD, the AMD Arrow logo, and combinations thereof, are trademarks of Advanced Micro Devices, Inc.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library, IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Snapshot, and the NetApp logo are trademarks or registered trademarks of NetApp, Inc. in the U.S. and other countries.

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

J2EE, Java, JDBC, JRE, MySQL, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Active Directory, Excel, Internet Explorer, Microsoft, SQL Server, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redbooks® publication provides a broad view of how Tivoli® system management products work together in several common scenarios. You must achieve seamless integration for operational personnel to work with the solution. This integration is necessary to ensure that the product can be used easily by the users.

Product integration contains multiple dimensions. We evaluate the implementation of the following areas:

- ▶ Security integration allows users to be authenticated from a single repository and seamless login to multiple applications. You do not have to log in to multiple applications, only to the first one. User identity and credentials only have to be maintained in one place, therefore, simplifying administration and reducing potential security exposure.
- ▶ Navigation integration allows multiple management applications to work with each other. From a managed resource in one application, a button or menu selection can bring additional information from another integrated application. Users can get information across multiple applications seamlessly.
- ▶ Data integration allows resources or object information to be exchanged between solutions to allow contexts to be established seamlessly.
- ▶ Task integration allows you to use one product and invoke the facility of another product.

We discuss these implementations and also provide sample scenarios about how these integrations work. We based these scenarios on common real-life examples that IT operations often have to deal with.

Finally, we also include additional information about topics, such as agent management, reporting, and product adoption routes.

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, Austin Center.

Budi Darmawan is a Project Leader at the International Technical Support Organization, Austin Center. He writes extensively and teaches IBM classes worldwide on all areas of systems management. Before joining the ITSO 10

years ago, Budi worked in Integrated Technology Services, IBM Indonesia as a Solution Architect and Lead Implementer.

Andre Jenie is a Senior IT Specialist in Singapore. He has more than 10 years of experience in IBM Tivoli enterprise management products and solutions. He holds a degree in Mathematics from Universitas Padjadjaran, Bandung, Indonesia. His areas of expertise include Tivoli Service Availability and Performance Monitoring (SAPM), ISM products, and Tivoli Provisioning Manager. He has written extensively for the IBM Tivoli Monitoring monthly newsletter for the Tivoli technical team in Asia Pacific, and he has written many technical documents in iRAM. He is now a member of the Tivoli Global Response Team (GRT) Asia Pacific.

Christina L Grimes is an Advisory Software Engineer for the IBM Software Group working from Indianapolis. She has eight years of experience in IBM Tivoli enterprise management. Her areas of expertise include client support for IBM Tivoli Netcool/OMNIBus and IBM Tivoli Netcool/OMNIBus Probes and Gateways. She is an IBM Certified Deployment Professional for Tivoli Netcool® Core. She is currently the Netcool and TBSM Level 2 Support US Team Lead.

Christopher Frost is an IT Specialist for IBM Software Group in Australia. He has three years of experience with Tivoli Software and has a Bachelor of Computer Science from Monash University in Melbourne. His areas of expertise include performance and availability monitoring, and he is Information Technology Infrastructure Library (ITIL®) Foundation-certified.

Giorgio Sommariva is an Advisory Operations Architect for Global Technology Services from Segrate, Italy. He has seven years of experience in the enterprise system management field. His areas of expertise include system monitoring, performance monitoring, service level monitoring, event management, configuration management, and service management. He is an ITIL Foundation Certified Professional, and he is now focused on network management and business service management solutions.

Guilherme Steinberger Elias is a Senior IT Specialist working from Hortolandia, Brazil. He has 13 years of experience in the system management field and has been working with Tivoli products for 11 years. He holds a Masters degree in Software Engineering and the titles of Network Specialist and Project Management from Unicamp University. He is an IBM Certified IT Specialist and is an IBM Certified Deployment Professional for Tivoli Monitoring 6 and ITIL Foundation Certified. His current areas of expertise include performance management, event management, asset management, and development. He currently works in Integrated Technology Delivery as an IT Architect focused on ISM solutions.

Hossam A Katory is a Staff Software Engineer in IBM Software Group, Cairo Lab in Egypt. He has over five years of experience in the Software Globalization and Bidirectional Languages Support field in the Tivoli, WebSphere®, and Lotus® families. He holds a BSc degree in Computer Engineering from AASTMT. His areas of expertise include Tivoli Maximo®, TAMIT, IBM Tivoli Change and Configuration Management Database, Release Process Manager, Tivoli Application Dependency Discovery Manager, and Lotus Foundations™.

Hyun Kuk Moon is an IBM Certified Professional IT Specialist for Software Group in IBM Korea. He joined IBM in 2000. He has been working in Tivoli Software for nine years. His areas of expertise include IBM Tivoli Monitoring, Tivoli Application Dependency Discovery Manager, Tivoli Provisioning Manager, Tivoli License Compliance Manager, and Tivoli Usage and Accounting Manager. He is ITIL-certified and CobiT-certified.

Jim Popovitch is a Software Developer with IBM Software Group, Tivoli Managing Engineer, Customer Solutions, in Atlanta, GA, IBM U.S.

Karim Badr is a Software Engineer in the IBM Software Group, Cairo Lab in Egypt. He is an Electronic Engineer and completed his advanced diploma in Software Engineering. He has been with IBM since 2005. Currently, he provides globalization support in Tivoli products and provides services in the region. His areas of expertise include Maximo Asset Management, IBM Maximo Asset Management for IT, IBM Tivoli Change and Configuration Management Database, Tivoli Application Dependency Discovery Manager, and Tivoli Provisioning Manager.

Marco Celon is an IT Specialist in IBM Software Group. He has over 10 years experience in the service provider industry in various areas ranging from performance management to customer care systems. He is an IBM Certified Solution Advisor, an IBM Certified Deployment Professional, and an IBM Certified Service-Oriented Architecture (SOA) Associate. His current focus is on performance and automation products from IBM Tivoli. He joined IBM in 2006 and is based in Rome, Italy.

Piotr A Boetzel is an IT Specialist at IBM Global Technology Services in Warsaw, Poland. He joined IBM four years ago and has been working on systems monitoring using the following products: Tivoli Monitoring, Tivoli Enterprise Console®, Netcool OMNIBus, Webtop, and IBM Director. He works for local clients and also for Global Systems Management Architecture that publishes Tivoli standards and tools. He holds several certifications: IBM Certified Advanced Deployment Professional - Tivoli Fault Management Solutions 2008, IBM Certified Deployment Professional - Tivoli Netcool Core V3.0, ITIL and others. He holds a Masters degree in Telecommunication from Warsaw University of Technology.

Rodrigo Carvalho Giometti is an IT Specialist currently working in Integrated Technology Delivery at IBM Brazil, Hortolandia. He joined IBM five years ago. He supports several clients' applications and monitors the critical parts of their businesses using the following products: IBM Tivoli Monitoring, IBM Tivoli Enterprise Console, and IBM Tivoli NetView. His areas of expertise also include Configuration and Event and Performance Management. He is ITIL-certified and has nine years of experience in monitoring and integrating IT solutions. He graduated in Designing and Developing Web Solutions from Anhembi Morumbi University.

Roger Turner is in the IBM Software Group. He is a Tivoli Application Dependency Discovery Manager, IBM Tivoli Change and Configuration Management Database, Tivoli Service Request Manager, IBM Maximo Asset Management for IT, and IBM Tivoli Business Service Manager Implementation Specialist and a Systems Management Specialist: IBM Maximo in Durham, N.C.



Figure 1 Roger Turner, Andre Jenie, Giorgio Sommariva, Karim Badr, Christina Grimes, Budi Darmawan, Marco Celon, and Guilherme Elias



Figure 2 Piotr Boetzel, Rodrigo Carvalho Giometti, Jim Popovitch, Christopher Frost, Moon Hyunkuk, and Hossam Katory

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

Bart Jacob, Tamikia Barrow, and Margaret A Ticknor
International Technical Support Organization

Conrad Johnson, Sandra Tipton, and Michael Kaczmarski
IBM Software Group Integration Lab

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 IBM Redbooks publications 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



Part 1

Introduction

In this part, we introduce this integration project and the scenarios that we cover.

Integration overview

In this chapter, we provide an overview of the integration of Tivoli products.

We discuss the following topics:

- ▶ 1.1, "Integration overview" on page 4
- ▶ 1.2, "Aspects of integration" on page 5
- ▶ 1.3, "Product coverage" on page 6
- ▶ 1.4, "Document organization" on page 6

1.1 Integration overview

When using unintegrated point products from multiple vendors together, you might encounter the following problems. Users must manually switch between product consoles, translating and transferring context and data, which is slow and error prone. Users must build, apply, and maintain their own integration capabilities. Users must learn multiple disparate user interface paradigms. Credentials for the same users must be maintained in multiple registries. A user must log in separately to each product console. Data from multiple product databases must be extracted and combined manually to produce useful reports. Data is modeled differently and uses different identifiers for the same artifacts, so it cannot be combined easily. Multiple logs in disparate formats and locations must be scrutinized to diagnose problems. By contrast, when multiple Tivoli products are used together, these problems are addressed. Based on this concern, Tivoli development started an integration initiative that provides a guideline about how to converge the products to a common set of rules to allow the products to work together. IBM is implementing this initiative with each product release to enhance the overall integration. The integration initiatives cover the following tracks:

- ▶ Security integration initiatives

Security integration enables Tivoli products to integrate on security aspects, such as authentication and single sign-on, shared user registry support, centralized user account management, consistent authorization, audit log consolidation, and compliance reporting.

- ▶ Navigation integration initiatives

Navigation initiatives allow seamless user interface transition from different Tivoli products when the context is needed. This seamless integration involves integrated user interface and launch in context abilities.

- ▶ Data integration initiatives

Data integration allows data structures to be exchanged to ensure the management context is available across different Tivoli products. This data integration includes event transfer and management resource consolidation.

- ▶ Task integration initiatives

Task integration allows a Tivoli management application to use a facility that is provided by a separate Tivoli product. Hence, they do not need to provide an overlapping functionality.

- ▶ Reporting integration

Reporting integration provides centralized management reporting across various Tivoli products. This reporting integration is realized by using the Tivoli Common Reporting.

- Agent management

Agent management allows self-monitoring of various Tivoli products using IBM Tivoli Monitoring agents.

In this IBM Redbooks publication, we provide a guide for integrating Tivoli technologies based on these initiatives. We only discuss the current implementation of the products. Future product versions and releases will contain additional integration features to allow more seamless coordination between products. Part 2, “Implementation” on page 15 discusses the implementation.

We also explore the use of scenarios. Scenarios are collections of work items that typically depict how an operations group uses a set of products to achieve a solution. We derived the scenarios from common real-world examples. In this book, we use scenarios in the following manner:

1. Explain the scenario.
2. Demonstrate the background information.
3. List the products used.
4. Explain the adoption route.
5. Describe the implementation steps.
6. Demonstrate the result.

We use scenarios to explain the integration in more detail. We discuss scenarios in Part 3, “Scenario walk-through” on page 339.

1.2 Aspects of integration

There are several aspects of integration, and several approaches to achieve integration. We describe integration from the operator’s point of view. IBM provides multifaceted and deep integration among its products to provide a seamless experience for users and enable the automation of processes:

- Coordinating security
- Establishing single sign-on
- Synchronizing data
- Centralizing the management of resources
- Managing the management system
- Invoking tasks across products

Within each aspect, there are separate levels of integration. For example, one aspect is *navigation*, which is the ability to move seamlessly between views provided by multiple related products. One level of navigation integration is “*launch*”, where one product console can be launched from another. A deeper level of navigation integration is “*launch in context*”, where the launched console

comes up in the same context that the user had in the launching console. A user might be looking at an event about a problem with a computer system and launch in context to another product console. When it comes up, it displays further information about that computer system. A deeper level of navigation integration is *shared console*, such as Tivoli Integrated Portal. The same console has panels with information from multiple products. When the user changes contexts in one panel, the other panels switch to the same context.

1.3 Product coverage

We chose the products that address several of the common scenarios that we have identified. We performed scenarios that use products from the system and asset management categories:

- ▶ IBM Tivoli Change and Configuration Management Database
- ▶ IBM Tivoli Application Dependency Discovery Manager
- ▶ IBM Tivoli Provisioning Manager
- ▶ IBM Tivoli Business Service Manager
- ▶ IBM Tivoli Netcool/OMNIBus
- ▶ IBM Tivoli Netcool/Impact
- ▶ IBM Tivoli Monitoring
- ▶ IBM Tivoli Network Manager for IP
- ▶ IBM Tivoli Service Request Manager
- ▶ IBM Tivoli Workload Scheduler
- ▶ IBM Tivoli Usage and Accounting Manager
- ▶ IBM Tivoli Storage Productivity Center

1.4 Document organization

The book is organized into the implementation and the scenarios:

- ▶ Part 1, “Introduction” on page 1:
 - Chapter 1, “Integration overview” on page 3
 - Chapter 2, “Integration scenarios” on page 9
- ▶ Part 2, “Implementation” on page 15:
 - Chapter 3, “Product installation overview” on page 17
 - Chapter 4, “Security integration” on page 45
 - Chapter 5, “Data integration” on page 129
 - Chapter 6, “Navigation integration” on page 231

- Chapter 7, “Self monitoring and management” on page 321
- ▶ Part 3, “Scenario walk-through” on page 339:
 - Chapter 8, “Operational drill down” on page 341
 - Chapter 9, “Automated application management scenario” on page 365
 - Chapter 10, “Executive IBM Tivoli Business Service Manager scenario” on page 403
 - Chapter 11, “Change and release management scenario” on page 413

Integration scenarios

In this chapter, we introduce the scenarios of the integration that we implement in our environment.

The discussion includes these topics:

- ▶ 2.1, "Overview of the scenarios" on page 10
- ▶ 2.2, "Common elements" on page 12
- ▶ 2.3, "Lab configuration" on page 13

2.1 Overview of the scenarios

The integration scenarios are usage patterns of Tivoli products. Tivoli development uses these usage patterns to drive product development. Development identifies and classifies these scenarios so that each scenario can be documented and planned into the product life cycle.

There are several scenarios that have been identified by the integration team as commonly implemented in client environments. These identified scenarios are useful for product development and direction, because they provide guidance about how the products are used.

The scenarios involve multiple products and encompass various management domains. From availability to provisioning, from security to monitoring, development defines and analyzes scenarios to ensure that integration can be performed and documented to be reused for future implementations. Development performs gap analysis to identify missing functions and to develop feedback for product development to complete the necessary functions.

Table 2-1 lists several of the identified scenarios that were available as of mid-2009.

Table 2-1 Identified scenarios

Scenario	Descriptions
Application availability	Monitoring application availability and ensuring its operation
Application deployment for management	Automated deploying and monitoring of application
Application monitoring with resource correlation	Integration of application and resource monitoring
Application problem	Application monitoring automated problem record creation
Asset and configuration management	Integration between asset management and configuration management
Asset management	Asset management with its interfaces to accounting and application configuration
Autonomic process flow	Automation on provisioning and monitoring servers
Autonomic serviceability and health	Monitoring of server status and potentially automatic resolution
Automated application management	Application management with its monitoring capabilities

Scenario	Descriptions
Automated compliance deployment	Ensuring security compliance with automated policy enforcement
Automated resource management	Automated monitoring of resources after being provisioned
Automated server provisioning	Automating server provisioning based on monitoring result
Automated storage provisioning	Storage provisioning when the disk reached a utilization threshold
Availability Center for SAP®	Monitoring the availability of SAP servers
Business service management (BSM) executive	Executive view for business services
Capacity management	Integrated capacity management from monitoring and accounting information
Change and release management	Integrated change and release management
Change and release management production deployment	Integrated change and release management in a production environment
Compliance management and remediation	Compliance checking and correction
Data center life cycle automation	Automating data center life cycle with provisioning and deprovisioning servers
Hardware asset management	Integrated discovery and asset management
Integrated log management	Collecting log data from various components
Integrated compliance reporting	Reporting tools to integrate compliance and provisioning information
IT business service management	Looking at business services provided by IT
IT and business event integration	Looking at event management and applying events to business services
Link failure at edge of network	Network management for edge of network, integrated failure with service desk
Link failure core network	Network management for core network, integrated failure with service desk
NextGen - Security fault performance	Next generation security fault and performance management

Scenario	Descriptions
NextGen - Layered performance isolation	Net generation performance problem isolation
Monitoring-Analysis-Planning-Execution (MAPE) loop advanced automation	Advanced automation with MAPE loop
Manage Our Stuff With Our Stuff (MOSWOS) transaction management for Tivoli Access Manager	Self monitoring for Tivoli Access Manager
Problem determination and impact analysis	Comprehensive problem determination and impact analysis
Release process manager	Integrated release management
Resource desired state configuration management	Resource state comparison with a baseline to determine resource health
Security configuration	Automated security configuration
Service management essentials	Basic service management, provisioning, and monitoring
Solution health	Managing the health of an integrated management system Managing the management system
SME level 2 analysis	Problem determination tools for level 2
Storage event	Event data integration from storage subsystem
Storage Process Manager	Working with Storage Process Manager
STG platform management	Integration of storage management to overall management infrastructure
Tivoli Enterprise Portal - How things are running	How are things running based on Tivoli Enterprise Portal
Tivoli Integration Portal - How things are running	How are things running based on Tivoli Integration Portal
User compliance management	Monitoring and enforcing user compliance

2.2 Common elements

From the scenarios that are identified in Table 2-1 on page 10, there are common elements of integration that are typical for many scenarios. These common

elements appear in several scenarios; therefore, we decided to demonstrate the implementation of these elements.

We identified and implemented the following common elements:

- ▶ Single sign-on (SSO) requirement on various products. The SSO requirement is extremely common for providing a usable and seamless experience for various products, either through a Java interface or a Web interface.
- ▶ Resource data in providing the context information for launch in context customization. The common resource data flows include feeding IBM Tivoli Application Dependency Discovery Manager using Discovery Library and then extracting the data to the necessary applications in a common data format.
- ▶ Event data integration that allows events to be forwarded to existing systems for further application. The additional facilities for events include the business system view and problem ticket creation.
- ▶ Development of a specialized outbound link to quickly jump across various products, while aware of the context of the original invocation.

2.3 Lab configuration

Figure 2-1 on page 14 depicts our lab configuration.

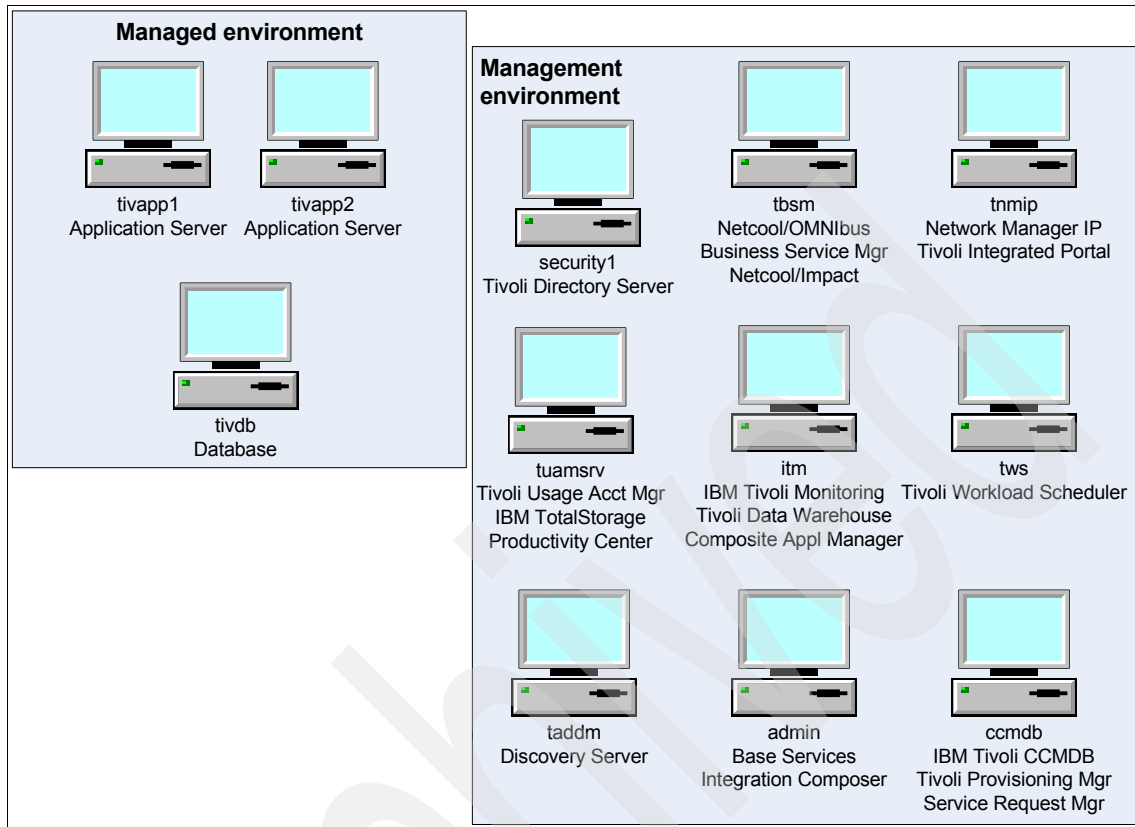


Figure 2-1 Environment summary

Figure 2-1 includes the following environments:

- ▶ **Managed environment**, which is a set of managed servers. The managed environment implements the Trader application using a WebSphere cluster, a database, and a security server.
- ▶ **Management environment**, which uses various Tivoli products that allow the managed environment to be automated, analyzed, and provisioned.



Part 2

Implementation

In this part, we discuss implementation for the integration of various Tivoli products. We divide the implementation into the integration types for ease of referencing.

Product installation overview

In this chapter, we discuss the fundamental installation and the component implementation for the Tivoli products that we use in our environment. We intend for this chapter to serve as a reference to the specific products that we install on certain machines. We also provide a list of installation parameters and results, such as directory paths, port numbers, user IDs, and other information.

We divide this discussion into the following areas:

- ▶ 3.1, “Installation overview” on page 18
- ▶ 3.2, “IBM Service Management products” on page 19
- ▶ 3.3, “IBM Tivoli Monitoring family” on page 28
- ▶ 3.4, “IBM Tivoli Netcool installation overview” on page 34
- ▶ 3.5, “IBM Tivoli Workload Scheduler” on page 39
- ▶ 3.6, “IBM Tivoli Usage and Accounting Manager” on page 41
- ▶ 3.7, “IBM Tivoli Storage Productivity Center for Data” on page 44

3.1 Installation overview

In this chapter, we do not discuss individual product installation. We describe the components and map them to the machines in our environment. We explain the fundamental steps and the common default parameters. Later, we explain the changes from this standard installation that are necessary to perform the product integration. We describe the installations of the following products:

- ▶ 3.2, “IBM Service Management products” on page 19:
 - IBM Tivoli Change and Configuration Management Database
 - IBM Tivoli Maximo Asset Manager for IT
 - IBM Tivoli Service Request Manager
 - IBM Tivoli Provisioning Manager
 - IBM Tivoli Application Dependency Discovery Manager
 - IBM Tivoli Integration Composer
 - Process Management products
- ▶ 3.3, “IBM Tivoli Monitoring family” on page 28:
 - IBM Tivoli Monitoring
 - IBM Tivoli Monitoring for Databases
 - IBM Tivoli Composite Application Manager for Web Resources
 - IBM Tivoli Composite Application Manager for Transaction
- ▶ 3.4, “IBM Tivoli Netcool installation overview” on page 34:
 - IBM Tivoli Netcool/OMNibus
 - IBM Tivoli Netcool/Webtop
 - IBM Tivoli Netcool/Impact
 - IBM Tivoli Business Service Manager
 - Tivoli Integration Portal
- ▶ 3.5, “IBM Tivoli Workload Scheduler” on page 39
- ▶ 3.6, “IBM Tivoli Usage and Accounting Manager” on page 41
- ▶ 3.7, “IBM Tivoli Storage Productivity Center for Data” on page 44

Clients can start with one or two products and then add products incrementally and integrate as needed. Apart from these product installations, we also discuss agent deployment and high availability implementation.

3.1.1 Agent deployment

You must integrate the agent deployment for various management products into an overall strategy. Separate types of agents require separate installation procedures and have unique requirements. You install certain agents in a more

stable environment of servers whose configurations do not change frequently. You must install other agents on user machines, for monitoring and maintenance purposes.

The agents residing on servers do not change much. You can install them in a single attended process; however, you likely need to automate the installation for user-based machines. You typically automate installation by packaging the installation into a “silent” installation or an installation with minimal interaction in which most of the parameters are provided with default values.

Tivoli uses a Common Offering Installer (COI) facility that allows multiple installations to be deployed with a minimal dialog. The Middleware installer for the IBM Tivoli Change and Configuration Management Database and IBM Tivoli Application Dependency Discovery Manager uses this technology.

3.1.2 Resiliency and high availability

The management environment becomes more and more critical for business, because automation and performance monitoring help ensure the availability of business processes. Without monitoring and automation functions, disrupted business processes and applications can cause businesses major problems. The management environment can be then deployed into a redundant environment to allow failover and fault tolerance, hence sustaining high availability.

There are several high availability options for implementing various Tivoli products. Typical Tivoli products have inherent fault tolerance and failover capabilities:

- ▶ Usage of RAID disks for Tivoli-related data storage that allow significantly improved data availability
- ▶ Usage of IBM Tivoli System Automation to allow more comprehensive automated subsystem switching, including DB2® and WebSphere infrastructure for the Tivoli environment

3.2 IBM Service Management products

IBM Service Management is an approach that is designed to automate and simplify the management of business services. IBM Tivoli Change and Configuration Management Database (CCMDB) is the foundation for the IBM Service Management solution. It is the foundation for the implementation of core Information Technology Infrastructure Library (ITIL) processes with the IBM Tivoli solution.

In our environment, we implement IBM Service Management products on several machines. Figure 3-1 on page 20 shows this implementation.

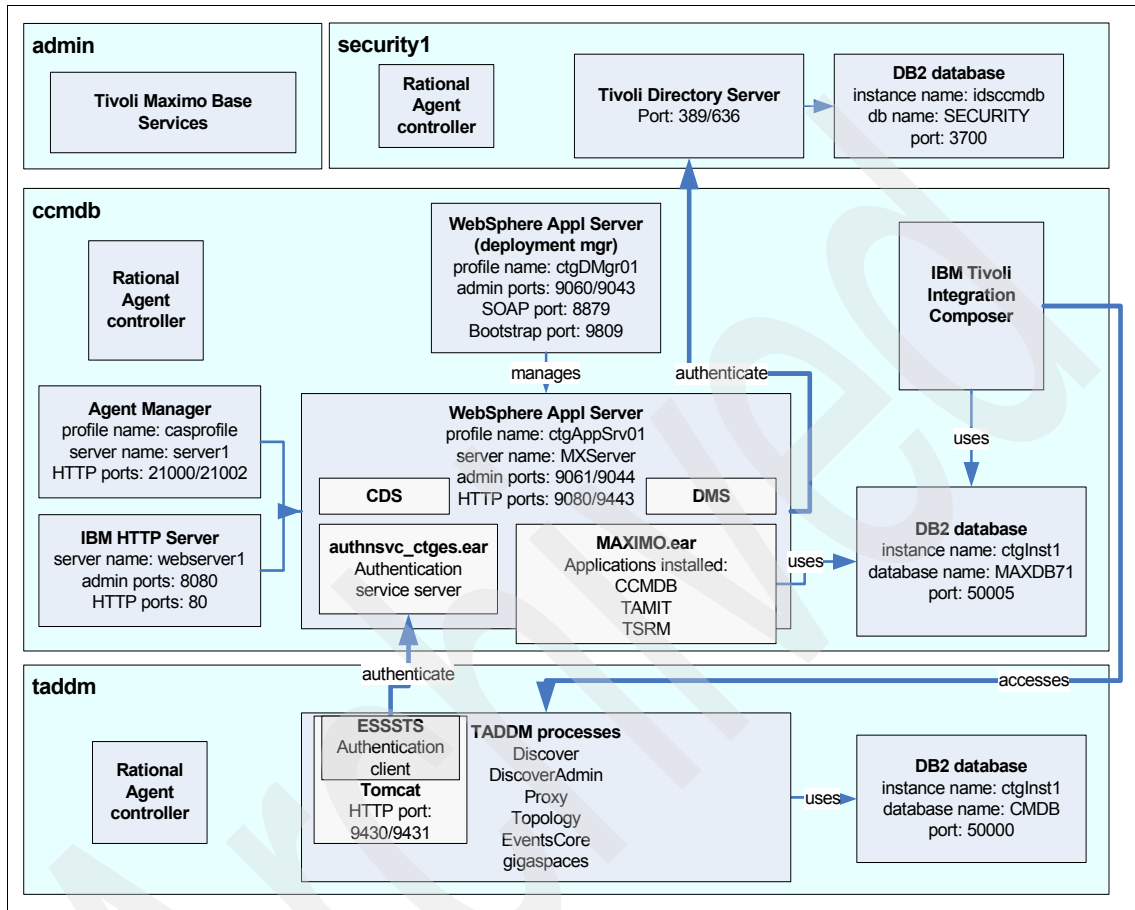


Figure 3-1 IBM Service Management solution configuration

In this section, we discuss these installations:

- ▶ 3.2.1, “IBM Service Management” on page 20
- ▶ 3.2.2, “IBM Tivoli Application Dependency Discovery Manager” on page 22
- ▶ 3.2.3, “IBM Tivoli Provisioning Manager” on page 24

3.2.1 IBM Service Management

Table 3-1 on page 21 shows the software versions of the products that we use for our IBM Service Management environment.

Table 3-1 IBM Service Management product versions

Product	Version	Part number
WebSphere Application Server	6.1	CZ0QEML and CZ0QFML
DB2 Enterprise Server	9.1	
(Middleware installer - Windows®)		
IBM Tivoli Directory Server (Middleware installer - Linux®)	6.1	CZ0QIML and CZ0QJML
IBM Tivoli Change and Configuration Management Database	7.1.1	CZ0QBML, CZ2JTML, and CZ0QDML
Tivoli Asset Management for IT	7.1	CZ2JZML and CZ2K0ML
IBM Tivoli Service Request Manager	7.1	C1C3EML and CZ33QML
Tivoli Application Dependency Discovery Manager	7.1.2	C1B3CML, C1B3DML, C1B3EML, and C1NE1ML
IBM Tivoli Unified Process Composer	7.1.0	C19ZNML
Tivoli Common Reporting	1.2.0.1	C1Y4IML

The implementation process consists of these steps:

1. Run the Middleware installer to install IBM Tivoli Directory Server on the security1 machine. Refer to this Web site:
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.ccmb.doc_7.1.1/install/c_ccmb_ccmbcmiddlewareoverview.html
2. Use the same Middleware installer to install DB2 and WebSphere Application Server on the ccmb Windows machine. We separated the IBM Tivoli Directory Server, because we wanted to have a shared directory server for all of our product environment.
3. Install IBM Tivoli Change and Configuration Management Database with Tivoli Base Services on the ccmb machine. Refer to this Web site:
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.ccmb.doc_7.1.1/install/t_ccmb_installfoundation.html
4. Install Rational® Agent Controller for collecting log and trace information. Refer to this Web site:
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.ccmb.doc_7.1.1/install/t_lta_acinstall.html

5. Install IBM Tivoli Integration Composer on the ccmdb machine, following the steps from this Web site:
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.ccmdb.doc_7.1.1/install/c_ccmdb_icoverview.html
6. Install IBM Maximo Asset Management for IT on top of IBM Tivoli Change and Configuration Management Database using Solution Installer. Refer to this Web site:
http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tamit.doc_7.1/pdf/tamit71_install_was.pdf
7. Install the Release Process Manager product as described at this Web site:
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.rpm.doc_7.1.1/rpm/t_rpm_install_gui.html
8. Install IBM Tivoli Service Request Manager:
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.srm.doc_7.1/installing/src/t_installing_srm.html

3.2.2 IBM Tivoli Application Dependency Discovery Manager

We have only one IBM Tivoli Application Dependency Discovery Manager domain server installed in our environment. This domain server is installed on one Linux machine. The domain server contains the following components:

- ▶ DB2 Universal Database Enterprise Server Edition V9.5 with Fix Pack 1 (part of the middleware packages CZ0QIML and CZ0QJML)
- ▶ IBM Tivoli Application Dependency Discovery Manager server V7.1.2:
 - IBM Tivoli Application Dependency Discovery Manager 1 of 4 V7.1.2, Linux (x86), Multilingual (C1B3CML)
 - IBM Tivoli Application Dependency Discovery Manager 2 of 4 V7.1.2, Linux (x86), Multilingual (C1B3DML)
 - IBM Tivoli Application Dependency Discovery Manager 3 of 4 V7.1.2, Linux (x86), Multilingual (C1B3EML)
 - IBM Tivoli Application Dependency Discovery Manager 4 of 4 V7.1.2, Linux (x86), Multilingual (C1NE1ML)

The implementation of IBM Tivoli Application Dependency Discovery Manager server consists of these steps:

1. Perform the prerequisite tasks:
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/InstallGuide/t_cmdb_install_prereq.html

2. Install IBM Tivoli Application Dependency Discovery Manager and DB2 using the simple installation:

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/InstallGuide/t_cmdb_install_simple_db2.html

3. Perform the post-installation tasks:

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/InstallGuide/c_cmdb_install_troubleshooting.html

Table 3-2 shows the fundamental settings for our IBM Tivoli Application Dependency Discovery Manager server.

Table 3-2 Installation settings

Setting	Value
Install Directory of IBM Tivoli Application Dependency Discovery Manager	/opt/IBM/cmdb
Non-root user	cmdbadmin
DB2 instance user ID	db2inst1
DB2 server port	50000
Archive DB2 user ID	archuser
Database name	cmdb
Remote Method Invocation (RMI) host name	default

Table 3-3 on page 24 shows the port usage for the Tivoli Application Discovery and Dependency Manager.

Table 3-3 Tivoli Application Discovery and Dependency Manager server ports

Setting	Port Number
Web server port	9430
Secure Sockets Layer (ssl) Web server port	9431
GUI server port	9435
GUI system SSL port	9434
Java Naming and Directory Interface (JNDI) port	9432
RMI port	9433
Topology Manager port	5636
Topology Builder port	5637
RMI daemon (RMID) port	1098

Table 3-4 contains information about ports that are used by the PingSensor and PortSensor.

Table 3-4 Ports used by the PingSensor and PortSensor to make connections

Port name	Port number
Domain Name System (DNS)	53
Lightweight Directory Access Protocol (LDAP)	389
Secure Shell (SSH)	22
Windows Management Instrumentation (WMI)	135
CiscoWorks	1741

3.2.3 IBM Tivoli Provisioning Manager

Tivoli Provisioning Manager is an automated resource management solution that allows you to manipulate the IT environment in real time according to defined business policies. Tivoli Provisioning Manager also helps you to manage the application life cycle of your managed systems.

When using IBM Tivoli Provisioning Manager V7.1, you must install IBM Tivoli Provisioning Manager on the same machine with other Tivoli Process Automation Engine-based products, such as IBM Tivoli Change and Configuration Management Database or other IBM Service Management

products, to achieve task integration. The APIs to invoke provisioning workflow remotely using SOAP will be available in IBM Tivoli Provisioning Manager V7.1.1. Therefore, we install IBM Tivoli Provisioning Manager in the same machine where IBM Service Management products are installed.

IBM Tivoli Provisioning Manager consists of many components. The following components are the major components in IBM Tivoli Provisioning Manager server:

- ▶ **Agent Manager**

Agent Manager is the server component of Common Agent Services (CAS) architecture. It provides services that allow Tivoli Common Agent to get information about agents and resource managers.

- ▶ **Device Management Service (DMS)**

DMS is responsible for job management operations. It initiates jobs, tracks the progress of jobs, and maintains the history of past jobs.

- ▶ **Dynamic Content Delivery Service (CDS)**

CDS is a grid-like distributed service that distributes large files around the network. It has a scalable design that allows second-tier machines or even agents to be a distribution point. Additional features include adaptive bandwidth control, file encryption, and a download activity report.

- ▶ **Base Services**

Base Services is the foundation layer of the IBM Service Management process layer, which provides, among other things, a common security model, a work management platform, and an integration service.

IBM Tivoli Provisioning Manager 7.1 runs on the WebSphere Application Server environment. There are two cells involved:

- ▶ The main cell hosts Tivoli Process Automation Engine with Content Delivery Service and Device Management Service.
- ▶ A stand-alone application server hosts Agent Manager.

All these WebSphere Application Server applications use DB2 Database Server through the ctginst1 instance and are hosted in the same machine. The external directory is hosted in the security1.itso.ral.ibm.com machine, which is accessed by WebSphere Application Server applications for authentication.

For more information about IBM Tivoli Provisioning Manager components and functions, go to this Web site:

http://publib.boulder.ibm.com/infocenter/tivihelp/v11r1/topic/com.ibm.tivoli.tpm.admin.doc/book/part_intro.html

IBM Tivoli Provisioning Manager is installed in Windows 2003 Standard Edition Service Pack (SP) 2. Use the following images to install IBM Tivoli Provisioning Manager 7.1:

- ▶ Use IBM Tivoli Provisioning Manager V7.1.0 Installation Multiplatform - TPM_V710_Install (C1Q8CML).
- ▶ Use IBM Tivoli Provisioning Manager V7.1.0 Core Components for Windows 32 - TPM_V710_CoreComp_Win32 (C1Q8DML).
- ▶ Do not use IBM Tivoli Provisioning Manager V7.1.0 Middleware for Windows 32 - TPM_V710_Midlwr_Win32 (C1Q8HML), because it is installed with IBM Service Management components.

We take these steps to install IBM Tivoli Provisioning Manager:

1. Install Cygwin manually in the IBM Tivoli Change and Configuration Management Database machine

The IBM Tivoli Provisioning Manager installation process can install Cygwin, but it assumes that the machine has a connection to the Internet. Because our machines do not have access to the Internet, we have to install the full copy of Cygwin.

Download the complete copy of Cygwin from this Web site:

<http://www.cygwin.com>

The following document explains which packages to install:

http://publib.boulder.ibm.com/infocenter/tivihelp/v11r1/topic/com.ibm.tivoli.tpm.ins.doc/install/tins_cygwin.html

2. Install IBM Tivoli Provisioning Manager core components

Before you begin the installation, make sure that you can ping the IBM Tivoli Directory Server machine using the fully qualified name. We use the following document, which discusses installing IBM Tivoli Provisioning Manager with IBM Tivoli Service Request Manager, at the following Web site:

http://publib.boulder.ibm.com/infocenter/tivihelp/v11r1/topic/com.ibm.tivoli.tpm.ins.doc/install/tins_srmwithtpm71.html

3. Install IBM Tivoli Provisioning Manager Web components (refer to the following Web site):

You must install the Web components from the same Admin machine where IBM Tivoli Change and Configuration Management Database and IBM Tivoli Service Request Manager are installed. Copy and extract the IBM Tivoli Provisioning Manager V7.1.0 Installation package to the Admin machine and follow the steps as described in the following document:

http://publib.boulder.ibm.com/infocenter/tivihelp/v11r1/topic/com.ibm.tivoli.tpm.ins.doc/install/tins_insttpmweb.html

You must make these initial important configuration changes after the installation:

- ▶ Installation paths:
 - DB2 installation - C:\Program Files\IBM\SQLLIB
 - WebSphere Application Server - C:\Program Files\IBM\WebSphere\AppServer
 - Cygwin - C:\cygwin
 - IBM Tivoli Provisioning Manager - C:\Program Files\IBM\tivoli\tpm
 - Agent Manager - C:\Program Files\IBM\AgentManager
 - Content Deliver Service (CDS) - C:\Program Files\IBM\tivoli\CDS
 - Device Manager Service (DMS) - C:\Program Files\IBM\DeviceManager
 - HTTP server - C:\Program Files\IBM\HTTPServer
 - IBM Tivoli Monitoring agent - C:\ibm\itm
- ▶ User IDs:
 - db2admin: DB2 administrators and Windows service user ID
 - wasadmin: WebSphere admin user name
 - cyg_server: Cygwin user name
 - tioadmin: IBM Tivoli Provisioning Manager user
 - maxadmin: IBM Tivoli Provisioning Manager admin user
- ▶ Port usage:
 - 80: HTTP port
 - 443: IBM Tivoli Provisioning Manager for OS Deployment HTTPS Port
 - 21000: Agent Manager starting port
 - 8080: IBM Tivoli Provisioning Manager for OS Deployment HTTP Port
 - 9043: Software Distribution Infrastructure (SDI) server SSL Port
 - 9046: Software Distribution Infrastructure (SDI) client SSL Port
 - 9080: Software Distribution Infrastructure (SDI) non-SSL Port
 - 9511: Agent Manager registration port
 - 9512: Agent Manager secure port
 - 9513: Agent Manager public port
 - 8008: HTTP administrative server port
 - 50005: DB2 ctginst1 instance port

Maximo is the key user interface of IBM Tivoli Provisioning Manager. In our environment, we access Maximo through the following Web site:

<http://ccmdb.itso.ral.ibm.com/maximo>

There is another interface to access Dynamic Content Deliver service. In our environment, we access the Dynamic Content Deliver service through this Web site:

<https://ccmdb.itso.ral.ibm.com:9443/admin>

3.3 IBM Tivoli Monitoring family

IBM Tivoli Monitoring products monitor the performance and availability of distributed operating systems and applications. We install IBM Tivoli Monitoring on our Red Hat Enterprise Linux 4 system with the installation packages that are shown in Table 3-5.

Table 3-5 Installation packages

Part number	Product
C1X0UEN	IBM DB2 for Linux, UNIX® and Windows V9.5 - Limited Use for Linux on 32-bit AMD™ and Intel® systems (x86) Multilingual
C1MP8EN	IBM Tivoli Monitoring V6.2.1 Base, Linux, English
C1MQ0EN	IBM Tivoli Monitoring V6.2.1 Agent, Multiplatform, English
C1R3JIE	IBM Tivoli Monitoring for Databases V6.2 Fix Pack 1 Base, Multiplatform, English
CZ0WREN	IBM Tivoli Composite Application Manager for WebSphere V6.1.0.4: Data Collector Linux, English
C1MQ5EN	IBM Tivoli Composite Application Manager for Web Resources V6.2.0.4: Tivoli Enterprise Management Agent for WebSphere Linux
CZ1VLEN CZ1VGEN CZ1WPEN CZ1VREN	IBM Tivoli Composite Application Manager for Transactions V7.1.0.2: Application Management Console, Web Response Time agent, Client Response Time agent, and Transaction Tracking components

We perform the installation according to the installation instructions in the IBM Tivoli Monitoring and DB2 Universal Database™ publications. We perform these specific installation processes:

1. Plan the deployment:

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp?topic=/com.ibm.itm.doc_6.2.1/itm_install33.htm

2. Install DB2 server:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.qb.server.doc/doc/t0008921.html>

3. Install IBM Tivoli Monitoring:

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp?topic=/com.ibm.itm.doc_6.2.1/itm_install122.htm

4. Additional configurations might be necessary, as discussed in this documentation:

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.1/itm_install191.htm

5. Install the application support files on Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server and Web client, and the Tivoli Enterprise Portal desktop:

- a. Although the installation processes differ, additional configuration is necessary after the installation process.

- b. We update the Tivoli Enterprise Monitoring Server using the command **itmcmd support -t <tems_name> <agent_codes>**:

tems_name	The name of Tivoli Enterprise Monitoring Server. The default is called TEMS.

agent_code	The two character agent code for each agent depending on the products installed.

- c. We update the Tivoli Enterprise Portal Server and Web client by configuring the cq agent or through the Manage Tivoli Enterprise Monitoring Services application.
- d. We update the Tivoli Enterprise Portal desktop by configuring the cj agent or through the Manage Tivoli Enterprise Monitoring Services application.

6. The agent installations differ by agent:

- The warehouse proxy, summarization, and pruning agent and operating system agents are installed with IBM Tivoli Monitoring:

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp?topic=/com.ibm.itm.doc_6.2.1/itm_install161.htm

- IBM Tivoli Composite Application Manager for Applications agents:
 - DB2 agent
 - WebSphere monitoring agent (and WebSphere data collector)
 - Web Server agent
- IBM Tivoli Composite Application Manager for Transaction agents:
 - BM Tivoli Composite Application Manager Console agent
 - Web Response Time agent

Because of the limitation of our test environment, we implement IBM Tivoli Monitoring server on a single machine. Figure 3-2 shows the resulting configuration. You can read about the IBM Tivoli Monitoring implementation configuration in *IBM Tivoli Monitoring: Implementation and Performance Optimization for Large Scale Environments*, SG24-7443.

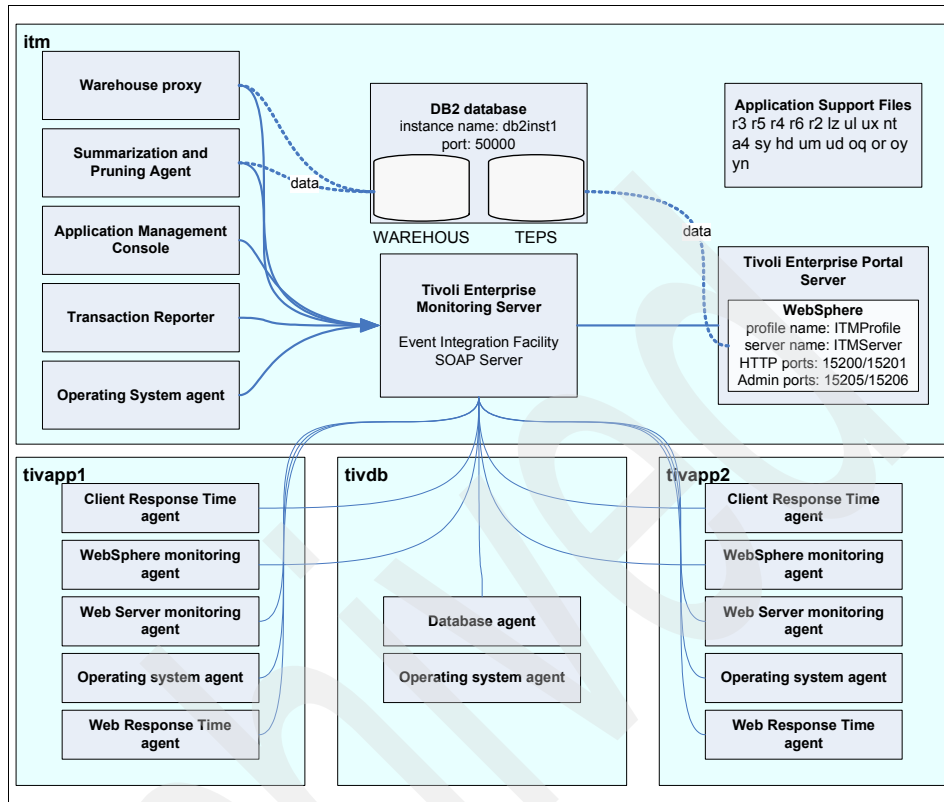


Figure 3-2 IBM Tivoli Monitoring configuration

Figure 3-2 on page 31 shows the installed components:

- Hub Tivoli Enterprise Monitoring Server is a key application component, which contains the collection and control points for performance and availability data and alerts that are received from the monitoring agents. The monitoring server manages the connection status of the agents and can be integrated to event management tools through Event Integration Facility. The standard Hub Tivoli Enterprise Monitoring Server implements a SOAP server with a Web Services interface, which is useful for administration and integration.
- Tivoli Enterprise Portal Server is a core interface and presentation component, which retrieves, manipulates, analyzes, and pre-formats data from Hub Tivoli Enterprise Monitoring Server in response to user actions at the portal client. Tivoli Enterprise Portal Server sends the data back to the portal client for presentation and to render the user interface views. It requires a relational database for storing information that is related to presentation and authentication. You can have many Tivoli Enterprise Portal servers

communicating with the same Hub Tivoli Enterprise Monitoring Server. Tivoli Enterprise Portal Server communicates to Tivoli Enterprise Portal clients. The client can be Java desktop-based or Web browser-based.

- ▶ Tivoli Data Warehouse is a database component that is used to store historical data collected from the monitoring agents in the environment. Tivoli Data Warehouse allows you to analyze historical trends.
- ▶ Warehouse Proxy Agent is a component that is used to transfer the data collected by the agents (Tivoli Enterprise Monitoring Agents) directly to Tivoli Data Warehouse (the WAREHOUS database) in a controlled way. It is possible to have several WarehouseProxy agents communicating with the same Tivoli Data Warehouse.
- ▶ Summarization and Pruning Agent is a component responsible for performing aggregation (summarization) and pruning (removal of data) functions on the Tivoli Data Warehouse data. These functions allow you to create flexible historical reports from the Tivoli Data Warehouse data. Implement the Summarization and Pruning Agent in the same machine as the Tivoli Data Warehouse for better performance.
- ▶ Operating System Agent is one of the Tivoli Enterprise Management Agents. It collects availability and performance data from the operating system and distributes it to the Tivoli Enterprise Monitoring Server.

It is important to use these installation paths as part of your setup:

- ▶ DB2 installation: /opt/ibm/db2/V9.5
- ▶ IBM Tivoli Monitoring: /opt/IBM/ITM

It is important to use these user IDs as part of your setup:

db2inst1	This ID is the database admin ID and instance owner for Tivoli Enterprise Portal Server DB and WAREHOUS DB. It is created in the db2iadm1 group.
itmuser	This ID is the login name of the database user that the portal server will use to access the database. Tivoli Enterprise Portal Server, Warehouse Proxy, and Summarization and Pruning also use this name as the Warehouse user ID to access the database. It is created in the db2iadm1 group.
wasadmin	This ID is the login name for WebSphere administration at http://localhost:15205/ibm/console .

It is important to use these ports as part of your setup. We use the default port mappings that are listed in Table 3-6.

Table 3-6 Port usage

Port	Component	Protocol
1918	Tivoli Enterprise Monitoring Server	IP.PIPE
3660	Tivoli Enterprise Monitoring Server SOAP Server	IP.SPIPE
1920	TEPS http requests from Tivoli Enterprise Portal clients	IP.PIPE
3661	TEPS https requests from Tivoli Enterprise Portal clients	IP.SPIPE
15001	TEPS default interface definition	TCP
15200	TEPS access using Java Web Start	TCP
6014	Warehouse Proxy	TCP
50000	DB2 Workgroup Server	TCP
389	LDAP Server (IBM Tivoli Directory Server)	TCP
9999	Eclipse Help Server	TCP

It is important to enter the additional configuration items that are listed in Table 3-7 as part of your setup.

Table 3-7 Configuration parameters

Description	Value
Hub Tivoli Enterprise Monitoring Server Name	HUB_itm
Tivoli Enterprise Portal Server Web site	http://itm.itso.ral.ibm.com:1920///cnp/client
Encryption key	IBMTivoliMonitoringEncryptionKey
Portal Server DB name	TEPS
Warehouse DB name	WAREHOUS
Installed Application Support	r3 r5 r4 r6 r2 lz ul ux nt a4 sy hd um ud oq or oy yn
Agent depot path	/opt/IBM/ITMDepot
Agents added to the deployment depot	lz ul um ud
Local Tivoli Enterprise Portal Client instance name	N/A

We perform the following additional customization:

- ▶ We change the path of the agent depot: We changed this path by setting the variable `DEPOTHOME` in the `<itm_installdir>/tables/Hub_ITM/KBBENV` path and in the `<itm_installdir>/config/kbbenv.ini` path:

```
DEPOTHOME=/opt/IBM/ITMDepot
```

- ▶ We used the commands in Example 3-1 to give appropriate authorizations to the portal server's DB2 user ID. An alternative is to assign the `itmuser` to the DB2 Administrator group. For this implementation, the group name is `db2iadm1`.

Example 3-1 Giving authorization for itmuser

```
db2 connect to TEPS
db2 GRANT
DBADM,CREATETAB,BINDADD,CONNECT,CREATE_NOT_FENCED_ROUTINE,IMPLICIT_S
HEMA,LOAD,CREATE_EXTERNAL_ROUTINE,QUIESCE_CONNECT ON DATABASE to
user itmuser
db2 disconnect
db2 connect to WAREHOUS
db2 GRANT
DBADM,CREATETAB,BINDADD,CONNECT,CREATE_NOT_FENCED_ROUTINE,IMPLICIT_S
HEMA,LOAD,CREATE_EXTERNAL_ROUTINE,QUIESCE_CONNECT ON DATABASE to
user itmuser
db2 disconnect
```

- ▶ WebSphere agent data collector installation paths:
 - `/var/ibm/tivoli/common`
 - `/opt/IBM/itcam/WebSphere/DC`
- ▶ Additional port usage:
 - Port 63335 used by Data Collector to connect to the WebSphere Agent
 - Port 8880 as the SOAP Connector

3.4 IBM Tivoli Netcool installation overview

In this chapter, we do not discuss the individual product installation; however, we map the product components and put them with the machines in our environment. We explain the steps and the common default variables. Later, we explain how to change this standard installation.

3.4.1 Netcool product versions

Table 3-8 shows the versions of the products that make up our IBM Tivoli Netcool environment.

Table 3-8 IBM Tivoli Netcool product versions

Product	Version
IBM Tivoli Netcool/OMNIBus	7.2.1 Fix Pack 3
IBM Tivoli Business Service Manager	V4.2.0.0 Build ID: 200809161731
IBM Tivoli Integrated Portal	Build ID: cf170821.07, 20080915_1203
IBM DB2	V9.5.0.1 Fix Pack 1
IBM Tivoli Network Manager for IP	V3.8 Build ID 27
IBM Tivoli Netcool/Impact	V5.1 Build ID: 20081024v51b30
IBM Tivoli Netcool/Webtop	V2.2 (included in IBM Tivoli Business Service Manager and IBM Tivoli Network Manager)

3.4.2 Configuration

We install the software listed in Table 3-8 on page 35 on two Linux systems, as shown in Figure 3-3. Figure 3-3 also shows the underlying components for the installed products.

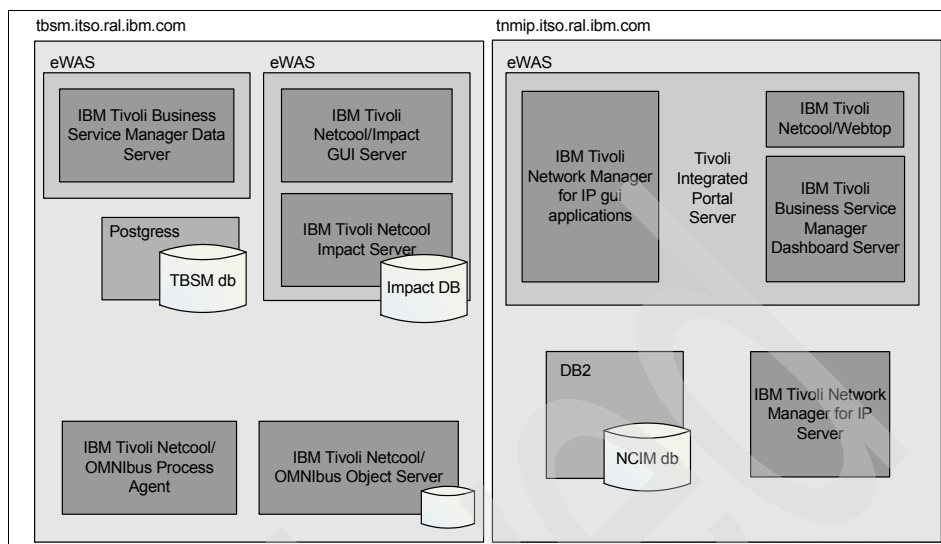


Figure 3-3 Component detail

During the installation, we use the following options:

- ▶ Use Object Server as the default authentication method
- ▶ Install as a non-root user
- ▶ Configure the Process Agent to manage the Object Server

We install our environment using the following steps:

1. Perform the default advanced IBM Tivoli Business Service Manager Data Server and IBM Tivoli Netcool/OMNibus installation with the IBM Tivoli Business Service Manager installer on the tbsm.itso.ral.ibm.com.
2. Perform the default advanced IBM Tivoli Business Service Manager Dashboard Server installation with the IBM Tivoli Business Service Manager installer on the tnmip.itso.ral.ibm.com.
3. Perform the default IBM DB2 installation on tnmip.itso.ral.ibm.com, and create the IBM Tivoli Netcool/Impact (NCIM) database to be used by IBM Tivoli Network Manager for IP.
4. Perform the advanced IBM Tivoli Network Manager for IP installation on tnmip.itso.ral.ibm.com (connecting to the existing Object Server running on tbsm.itso.ral.ibm.com) to the existing Tivoli Integrated Portal running on tnmip.itso.ral.ibm.com and to the existing NCIM IBM DB2 database on tnmip.itso.ral.ibm.com.

5. Perform the advanced IBM Tivoli Netcool/Impact installation on `tbsm.itso.ral.ibm.com`, changing the database port number to 5445 to avoid conflict with the Postgres that is database used by IBM Tivoli Business Service Manager.

3.4.3 Installation results

Table 3-9 shows all of the users that are created by the installation.

Table 3-9 Netcool environment default users

Role	System	User
Non-root installation system user	tbsm tnmip	netcool
IBM Tivoli Netcool/OMNIBus Administrator	tbsm	root
Tivoli Integrated Portal administrator	tnmip	tipadmin
IBM Tivoli Business Service Manager postgres user	tbsm	postgres
Tivoli Integrated Portal IBM Tivoli Business Service Manager administrator	tnmip	tbsmadmin
IBM DB2 instance owner	tnmip	db2inst1
IBM DB2 administrator	tnmip	dasusr1
IBM DB2 fenced user	tnmip	db2fenc1
IBM DB2 NCIM database owner	tnmip	ncim
Tivoli Integrated Portal IBM Tivoli Network Manager for IP administrator	tnmip	itnmadmin
Tivoli Integrated Portal IBM Tivoli Network Manager for IP user	tnmip	itnmuser
IBM Tivoli Netcool/Impact administrator	tbsm	admin
IBM Tivoli Netcool/Impact WebSphere Application Server administrator	tbsm	wasadmin

Table 3-10 on page 37 lists the port numbers that are used by the software.

Table 3-10 IBM Tivoli Netcool product communication ports

Utilization	System	Port number
Tivoli Integrated Portal starting ports	tnmip	16310, 16316

Utilization	System	Port number
IBM DB2 communication port	tnmip	50000
IBM Tivoli Netcool/OMNIBus Object server port	tbsm	4100
IBM Tivoli Netcool/OMNIBus Process Agent port	tbsm	4200
IBM Tivoli Business Service Manager Postgres communication port	tbsm	5435
IBM Tivoli Business Service Manager Data Server communication port	tbsm	17542
IBM Tivoli Business Service Manager Data Server starting port	tbsm	17310
IBM Tivoli Business Service Manager Dashboard Server communication port	tnmip	17543
IBM Tivoli Netcool/Impact name server port and http port	tbsm	9080
IBM Tivoli Netcool/Impact admin port	tbsm	9060
IBM Tivoli Netcool/Impact netcool database port	tbsm	5445
IBM Tivoli Netcool/Impact command line port	tbsm	2000
IBM Tivoli Network Manager for IP rendezvous ports	tnmip	7979 - daemon 7600 - rva

Table 3-11 on page 38 shows the product installation paths on the two systems.

Table 3-11 IBM Tivoli product installation paths

System	Product name	Path
tbsm	IBM Tivoli Business Service Manager data server	/opt/IBM/tivoli/tbsm
tbsm	IBM Tivoli Netcool/Impact	/opt/IBM/netcool/impact
tbsm	IBM Tivoli Netcool/OMNIBus	/opt/IBM/tivoli/netcool/omnibus
tnmip	Tivoli Integrated Portal	/opt/IBM/tivoli/tip
tnmip	IBM Tivoli Network Manager for IP	/opt/IBM/tivoli/netcool/precision
tnmip	IBM DB2	/home/db2inst1/
tnmip	IBM Tivoli Business Service Manager dashboard server	/opt/IBM/tivoli/tbsm

As in UNIX-like systems, each process has its own set of environmental variables. IBM Tivoli Netcool products have their own environmental variable to set. Example 3-2 shows an example of the `.bash_profile` file.

Example 3-2 The `.bash_profile` file

```
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi
# User specific environment and startup programs
PATH=$PATH:$HOME/bin
LANG=C
JAVA_HOME=/usr/bin/java
NCHOME=/opt/IBM/tivoli/netcool
TIPHOME=/opt/IBM/tivoli/tip
OMNIHOME=$NCHOME/omnibus
PRECISION_HOME=$NCHOME/precision
PERLLIB=$PRECISION_HOME/perl/lib/5.8.8:$PRECISION_HOME/perl/lib/site_perl:$PRECISION_HOME/perl/lib/site_perl/5.8.8
PATH=$PATH:$PRECISION_HOME/perl/bin:$NCHOME/bin:$PRECISION_HOME/bin:$NCHOME/license/bin:$OMNIHOME/bin
NC_RULES_HOME=$NCHOME/etc/rules
export NCHOME TIPHOME OMNIHOME LANG JAVA_HOME
export PRECISION_HOME PERLLIB PATH NC_RULES_HOME
unset USERNAME
```

3.5 IBM Tivoli Workload Scheduler

We install Tivoli Workload Scheduler V8.5 and Tivoli Dynamic Workload Console on our Red Hat Enterprise Linux 4 operating system with the following installation packages:

C1V0BML	Integration tools
C1V0NML	Tivoli Workload Scheduler V8.5
C1V0PML	Tivoli Dynamic Workload Console
C1V0QML	launchpad
C1V6VML	DB2 Universal Database V9.5

We perform the installation according to the installation instructions in the Tivoli Workload Scheduler V8.5 publications that can be found at this Web site:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.tivoli.itws.doc/toc.xml>

We perform the following specific installation processes:

1. Prepare the prerequisites:

<http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27012175>

2. Install DB2 database for Tivoli Workload Scheduler.

3. Install the Master Domain Manager:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tivoli.itws.doc/awspimst61.htm#freshinstall>

4. Install the Tivoli Dynamic Workload Console:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tivoli.itws.doc/awspimst176.htm#webui_wizard

Because of the limitation of our test environment, we implement Tivoli Workload Scheduler server on a single machine. Figure 3-4 on page 40 shows the resulting configuration. You can read about the Tivoli Workload Scheduler implementation configuration in *Getting Started with IBM Tivoli Workload Scheduler V8.3*, SG24-7237.

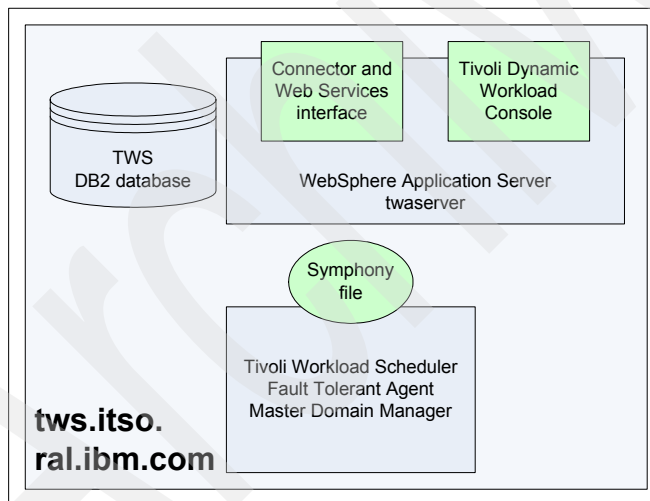


Figure 3-4 Tivoli Workload Scheduler configuration

Figure 3-4 shows the installed components:

- Tivoli Workload Scheduler agent is the agent that runs the batch workload. The batch workload definition is stored in the Symphony file. Because we run a stand-alone environment, the agent also acts as the Master Domain Manager, on which all job dependencies are resolved.

- ▶ The DB2 database that is called TWS and TWS_DB is used to store the workload definitions. The definition is read and loaded into the Symphony file to control the current day's work plan.
- ▶ The J2EE™ Enterprise Applications provide the user interface support running on WebSphere Application Server. The standard Tivoli Workload Scheduler installation implements the Web Services interface and connector for the Job Scheduling Console. We have the Web-based Tivoli Dynamic Workload Console as the additional interface.

Note the following important setup information:

- ▶ Installation paths:
 - DB2 installation: /opt/IBM/db2/V9.5
 - Tivoli Workload Applications: /opt/IBM/TWA
 - IBM Tivoli Monitoring Database monitoring agent: /opt/IBM/ITM
 - Tivoli System Automation agent: /opt/IBM/tsamp
- ▶ User IDs:
 - twsinst: Tivoli Workload Scheduler instance owner
 - db2inst1: DB2 database instance owner and administrator
 - db2fenc1: DB2 database stored procedure executor
- ▶ Port usage:
 - 31111: Fault Tolerant Agent netman's listening port
 - 31123/31124: HTTP listener ports
 - 31125: SOAP port and so on

The Integrated Solution Console is the base of the user interface tools for Tivoli Workload Schedule. We use this URL for our environment:

<https://tws.itso.ral.ibm.com:31124/ibm/console>

Our basic installation of the embedded WebSphere Application Server uses a local operating system user as the authentication method. In order to perform the security and navigation integration, we must switch this authentication method to authenticate through IBM Tivoli Directory Server's LDAP.

3.6 IBM Tivoli Usage and Accounting Manager

IBM Tivoli Usage and Accounting Manager V7.1.2 is installed on our Red Hat Enterprise Linux 4 machine, based on the following installation packages:

C92A0ML	DB2 Enterprise Server Edition, the DB2 Client, and the DB2 Runtime Client V9.1 for Linux on 32-bit AMD and Intel systems (x86) Multilingual

CZ3DLML	IBM Tivoli Usage and Accounting Manager V7.1.2 Enterprise Collector Pack for Linux IA32, Multilingual
CZ3DDML	IBM Tivoli Usage and Accounting Manager V7.1.2 Enterprise Edition for Linux IA32, Multilingual

We perform the installation according to the installation instructions in the Tivoli Usage and Accounting Manager V7.1 publications, which are at this Web site:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=/com.ibm.ituam.doc/welcome.htm>

We perform the following specific installation processes:

1. Prepare the prerequisites:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.ituam.doc/install/c_supported_hardware_and_software.html

2. Install DB2 database for Tivoli Usage and Accounting Manager.

3. Install the Usage and Accounting Manager Enterprise Edition:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.ituam.doc/install/t_installing_ee_ve_on_unix.html

or

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.ituam.doc/install/t_installing_ee_ve_on_win.html

4. Install the Usage and Accounting Manager Enterprise Collector Pack:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.ituam.doc/install/t_installing_ecp_on_unix.html

or

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.ituam.doc/install/t_installing_ecp_on_win.html

5. Perform the initial customization for database initialization and job processing:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.ituam.doc/config/c_configuring_after_installation.html

Because of the limitation of our test environment, we implement the Tivoli Usage and Accounting Manager server on a single machine. Figure 3-5 on page 43 shows the resulting configuration. You can read about Tivoli Usage and Accounting Manager implementation configurations in *IBM Tivoli Usage Accounting Manager V7.1 Handbook*, SG24-7404.

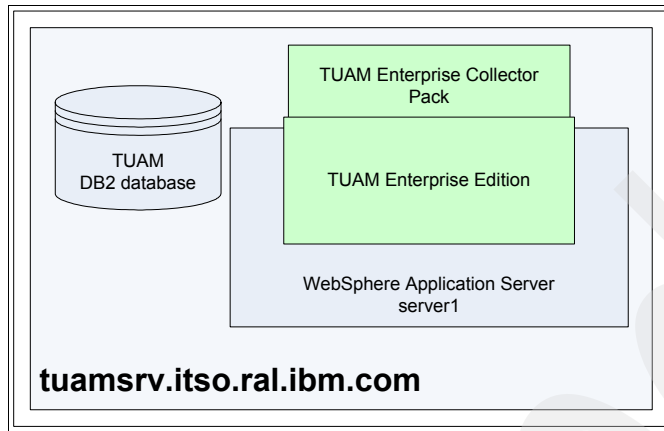


Figure 3-5 Tivoli Usage and Accounting manager

Tivoli Usage and Accounting Manager is primarily an application that stores usage and billing data in a database. Its main component is the database that stores its data. WebSphere Application Server provides a Web-based interface for administering and operating the product.

Note the following important setup information:

- ▶ Installation paths:
 - /opt/IBM/db2/V9.1
 - /opt/IBM/tuam
- ▶ User IDs: Tivoli Usage and Accounting Manager does not require any specific user ID settings:
 - db2inst1: DB2 database instance owner and administrator
 - db2fenc1: DB2 database stored procedure executor
- ▶ Port usage:

Tivoli Usage and Accounting Manager Web administration uses port 11052 for HTTP access and 11053 for HTTPS access. You can access the user interface from this Web site:

<http://tuamsrv.itso.ral.ibm.com:11052/ibm/console>

The fundamental installation of the Tivoli Usage and Accounting Manager Web interface does not use security.

3.7 IBM Tivoli Storage Productivity Center for Data

In our environment, IBM Tivoli Storage Productivity Center for Data is installed also in a Red Hat Enterprise Linux environment. The installation is performed with the following packages:

CZ2HZML	IBM DB2 9.1, Fix Pack 5 for Linux IA32
CZ2ILML	Agent Manager for Linux
CZ2HPML	Agent for Linux IX86
CZ2GYML	IBM Tivoli Storage Productivity Center for Data V4.1, Linux, Part 1
CZ2GZML	IBM Tivoli Storage Productivity Center for Data V4.1, Linux, Part 2

Follow these steps for the installation process:

1. Evaluate the planning information for the implementation as discussed in this document:

http://publib.boulder.ibm.com/infocenter/tivihelp/v4r1/topic/com.ibm.tpc_v41.doc/fqz0_r_planning.html

2. We must install the database. We choose to install the DB2 database locally on the IBM Tivoli Storage Productivity Center (TPC) server. The instruction is provided in this document:

http://publib.boulder.ibm.com/infocenter/tivihelp/v4r1/topic/com.ibm.tpc_v41.doc/fqz0_t_installing_db2_on_linux_or_unix_gui.html

3. We install the Agent Manager. We choose an installation that utilizes a local DB2 32-bit installation, as described in this document:

http://publib.boulder.ibm.com/infocenter/tivihelp/v4r1/topic/com.ibm.tpc_v41.doc/fqz0_t_installing_the_agent_manager_32bit_DB2_gui.html

4. The IBM Tivoli Storage Productivity Center server components are installed with an existing Tivoli Integrated Portal. We perform the installation according to this document:

http://publib.boulder.ibm.com/infocenter/tivihelp/v4r1/topic/com.ibm.tpc_v41.doc/fqz0_t_installing_tpc_custom.html

IBM Tivoli Storage Productivity Center for Data is then installed with the Tivoli Integrated Portal as the GUI interface. It integrates seamlessly with our Tivoli Integrated Portal that we install in 3.4, “IBM Tivoli Netcool installation overview” on page 34.

Security integration

In this chapter, we discuss the implementation of centralized user management and authentication and the configuration of single sign-on (SSO) for security integration. We achieve centralized user management by using WebSphere Federated Repositories technology that enables Tivoli products to share a common Lightweight Directory Access Protocol (LDAP)-based user repository.

We discuss the following topics in this chapter:

- ▶ 4.1, “Background security concepts” on page 46
- ▶ 4.2, “Security setup overview” on page 53
- ▶ 4.3, “Integrated Solution Console setup” on page 56
- ▶ 4.4, “Tivoli Process Automation Engine security setup” on page 73
- ▶ 4.6, “IBM Tivoli Netcool products LDAP configuration” on page 95
- ▶ 4.7, “IBM Tivoli Monitoring” on page 120

4.1 Background security concepts

In this section, we discuss background security concepts that are important for you to understand before you begin to configure security for Tivoli products. We discuss these concepts:

- ▶ 4.1.1, “Lightweight Directory Access Protocol” on page 46
- ▶ 4.1.2, “WebSphere federated repositories” on page 48
- ▶ 4.1.3, “External authentication” on page 50
- ▶ 4.1.4, “Single sign-on” on page 51

4.1.1 Lightweight Directory Access Protocol

Lightweight Directory Access Protocol (LDAP) is an open industry standard that defines a common method for accessing and updating information in a directory. The integration of various Tivoli products starts with sharing a common user repository. Having various components with separate technologies requires an integration point that is external to the products. We use the LDAP directory for coordinating security authentication issues.

A *directory* is a listing of information about objects arranged in a certain order that gives details about each object. Directories allow users or applications to find resources that have the characteristics needed for a particular task.

A directory contains a collection of objects organized in a tree structure. The LDAP naming model defines how entries are identified and organized. Entries are organized in a tree-like structure called the *Directory Information Tree* (DIT). Entries are arranged within the directory tree based on their *distinguished name* (DN). A DN is a unique name that unambiguously identifies a single entry. DNs are made up of a sequence of *relative distinguished names* (RDNs). Each RDN in a DN corresponds to a branch in the directory tree leading from the root of the directory tree to the directory entry. A DN is composed of a sequence of RDNs separated by commas, such as `cn=vbudi,ou=users,ou=SWG,o=IBM,c=US`.

You can define your directory tree based on your organizational needs as shown in Figure 4-1 on page 47. Each RDN uses a qualifier that is used to signify a type of entity. This list shows several common qualifiers:

c	country
o	organization
ou	organizational unit
dn	distinguished name
cn	common name
sn	surName

The leaf nodes in the LDAP tree can have a set of attributes that further qualifies and defines the entity.

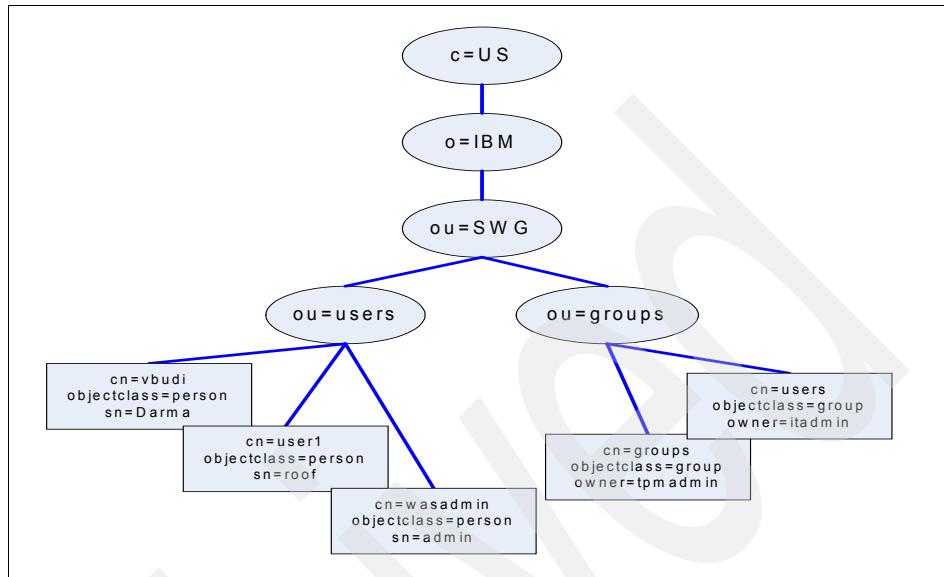


Figure 4-1 LDAP tree

In Figure 4-1, the tree starts with the node `c=US`. The main storage branch resides under `ou=SWG,o=IBM,c=US`. The main storage branch on which all processing (inserts, queries, and removals) is performed is typically called the *suffix*.

In IBM Tivoli Directory Server, users and groups are typically defined under `ou=users` and `ou=groups` nodes as shown in Figure 4-1. The leaf user node has the following important attributes:

uid	User identifier
userPassword	Binary field to hold the password
objectclass	Separate supported objectclass for this user (typical object classes are <code>inetOrgPerson</code> , <code>person</code> , <code>ePerson</code> , and so on)

The leaf group node has the following important attributes:

objectclass	Separate supported objectclass for this group entity
members	User members of the group

LDAP is an TCP/IP-based application. It listens to the port 389 for plain communication and port 636 for Secure Sockets Layer (SSL) communication. Use SSL for LDAP processing, because the LDAP traffic contains sensitive information. In our environment, we use non-SSL communication. However, in typical client environments, use SSL communication.

IBM Tivoli Directory Server implements the Internet Engineering Task Force (IETF) LDAP V3 specifications. It also includes enhancements that have been added by IBM in functional and performance areas. This version uses IBM DB2 Universal Database as the data storage to provide individual LDAP operational transaction integrity, high performance operation, and online backup and restore capability. IBM Tivoli Directory Server interoperates with the IETF LDAP V3-based clients.

IBM Tivoli Directory Server has three base components:

- ▶ IBM DB2 Universal Database is the data storage to provide individual LDAP operational transaction integrity, high performance operation, and online backup and restore capability.
- ▶ The server executable is named `ibmslapd`.
- ▶ Tools to administer and configure the directory. These tools rely on the directory administration daemon (`ibmdiradm`), which runs on each server machine and also enables remote management.

Using IBM Tivoli Directory Server, the initial authentication for the LDAP connection uses a user ID in the `bindDN` field. The `bindDN` that is typically used is `cn=root`. For more information about LDAP, refer to *Understanding LDAP - Design and Implementation*, SG24-4986.

4.1.2 WebSphere federated repositories

WebSphere federated repositories is the latest addition to the authentication mechanism in WebSphere V6.1. Federated repositories is also known by several other names, such as WebSphere Identity Manager and Virtual Member Manager.

Prior to WebSphere V6.1, user authentication is supported through one of the following repositories:

- ▶ Local operating system
- ▶ An LDAP directory
- ▶ Custom user registry

Federated repositories allows users to be authenticated through one or more repositories. It not only allows read-only access, but it also allows the creation of users and groups to one of the defined repositories. The supported repositories for federated repositories include the following repositories:

- ▶ File-based repository (this is the default)
- ▶ Local operating system
- ▶ An LDAP directory

Note: Even though you cannot configure a custom user registry in federated repositories using the administration console or the `wsadmin` command-line tool, certain Tivoli products introduce a custom user registry that participates in federated repositories.

Federated repositories provides the ability to map entries from multiple individual user repositories into a single virtual repository. A federated repository consists of a single named realm that consists of a set of independent user repositories. Each repository can be an entire external repository or, in the case of LDAP, a subtree within that repository. The root of each repository is mapped to a *base entry* within the federated repository, which is basically a starting point within the hierarchical namespace of the virtual realm.

Note the following considerations for federated repositories:

- ▶ You can only configure one user repository to be the target for creating users/groups from the administration console. By default, this one user repository is the file repository, but you can change the repository.
- ▶ The username (for example, LDAP uid) must be unique across the various repositories. For example, users cannot have the same uid in separate LDAP directories, even under separate organizational structures.
- ▶ If any repositories in the federation are down, you cannot authenticate (even as an admin), regardless of which repository your particular ID is stored in. The federated repositories component always checks all repositories before letting an authentication succeed.
- ▶ Although federated repositories has the capability to support multiple realms, WebSphere Application Server only supports a single realm at this time. This single realm support is defined at the cell level and is shared by all applications.

The configuration file for federated repositories is stored in the WebSphere configuration. Federated repositories is activated when the current authentication method in the `security.xml` file under the `profiles/<profile name>/config/cells/<cellname>` directory refers to the Wim repository. The settings of the federated repositories are stored in the `wimconfig.xml` file in the `profiles/<profilename>/config/cells/<cellname>/wim/config` directory.

Figure 4-2 describes the authentication mechanism for WebSphere to verify access for users.

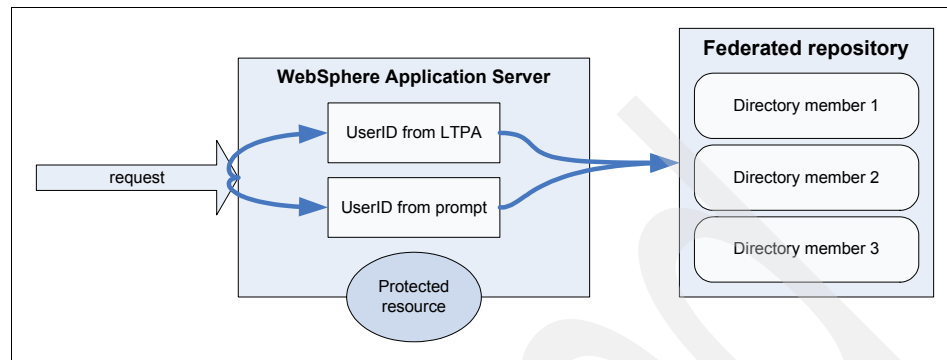


Figure 4-2 Federated repositories authentication

Figure 4-2 shows the authentication process in WebSphere:

1. When a protected resource is accessed, WebSphere asks for a credential (user ID and password).
2. When there is no credential, WebSphere prompts for the user ID and password. The user ID is validated against the currently selected authentication mechanism. Typically, the mechanism includes a verification to a user registry. When the user ID and password combination is valid, the credential is verified.
3. If there is an existing credential (such as a prior login or through an Lightweight Third Party Authentication (LTPA) token), the credential is used to verify access.
4. The user ID and the group to which it belongs are checked against the protection role of the resource. If the user or any group to which it belongs has access to the role, the access is granted; otherwise, the access is denied.

Setting up LDAP as an authentication provider requires that the LDAP is used as the repository. We explain how to set up LDAP for inclusion in the federated repositories. Although the federated repositories allows the users to be defined on several repositories, the same user ID must not appear on separate repositories.

4.1.3 External authentication

The external authentication mechanism uses the authentication client that connects to an authentication service inside a WebSphere cell. The components of this scheme are also called Embedded Security Services and Security Token

Services. Figure 4-3 demonstrates the concept of the external authentication service.

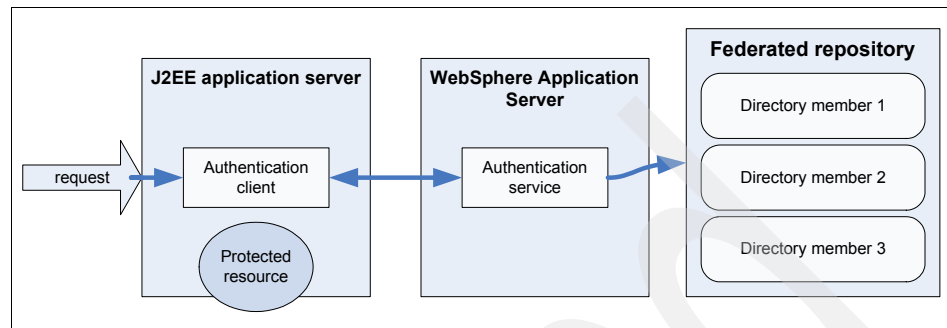


Figure 4-3 External authentication

Figure 4-3 shows these concepts:

- ▶ The external authentication provides a secure communication between the authentication service and the authentication client.
- ▶ The authentication service resides in a WebSphere Application Server as an Enterprise Application. For IBM Service Management products, this application is called `authnsvc_ctges.ear`. This authentication service interacts with Virtual Member Manager to authenticate, accept, or reject a user.
- ▶ The authentication client accesses the authentication server using a Web Services SOAP call. The authentication server then provides an authentication token. The authentication client acts as an extension of the local security mechanism for the client environment (such as Apache Tomcat for IBM Tivoli Application Dependency Discovery Manager).

Note: Although external authentication can generate the LTPA token, IBM Tivoli Application Dependency Discovery Manager currently does not utilize this feature. Thus, you cannot launch with single sign-on from IBM Tivoli Application Dependency Discovery Manager.

4.1.4 Single sign-on

Single sign-on (SSO) is a mechanism that allows applications that reside on separate servers to cross-authenticate the user. Authenticated users on one application do not need to re-authenticate when accessing the other application.

Because Tivoli products use Web-based interfaces, SSO becomes a critical usability challenge as operators traverse multiple application servers to use

separate products. Signing on multiple times using the same user ID and password pair is cumbersome and error-prone.

The most common mechanism to provide a single sign-on is to use the LTPA token. The LTPA token is a Web browser session cookie that contains an encrypted user ID and authentication information. Application servers that share the encryption information and use the same authentication can decrypt the information and use the existing authentication information. Figure 4-4 illustrates the LTPA token mechanism.

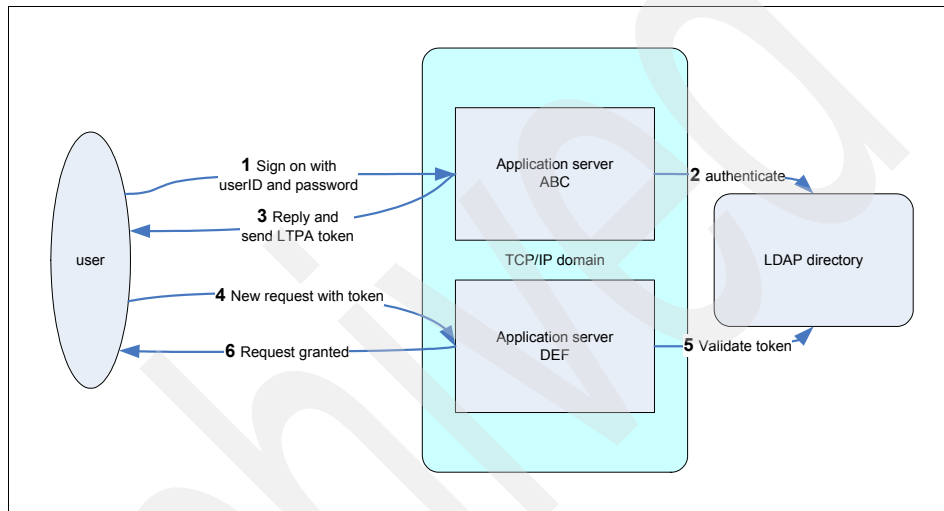


Figure 4-4 Single sign-on with LTPA

Figure 4-4 shows these processes:

1. The user authenticates on server1 with the user's user ID and password by using a Web browser.
2. Application server server1 generates an LTPA token as a session cookie to the Web browser. This token is in an encrypted message.
3. All further requests to server1 are authenticated based on the LTPA token.
4. When the same user tries to access server2, while server2 shares the same TCP/IP domain as server1, the browser retains the LTPA token.
5. Upon receiving the LTPA token, server2 tries to decrypt the token based on its LTPA key pair.
6. When the token can be decrypted successfully and the user is authorized to access the resource in server2, the user obtains access to the resource in server2 without needing to log in.

The following requirements apply for two application servers to allow SSO using an LTPA token:

- ▶ They must use the same LDAP server for authentication information. The LTPA token can only be verified through the LDAP server.
- ▶ They must reside in the same TCP/IP domain; otherwise, the LTPA token cookie is not sent to the server. In our sample environment, all servers have the same domain of `itso.ral.ibm.com`.
- ▶ They must have synchronized time, because the LTPA token contains an expiration time stamp; this synchronization can be achieved using Network Time Protocol (NTP). We do not discuss NTP implementation in this IBM book.
- ▶ They must share the LTPA encryption key to be able to decrypt the token.
- ▶ You must enable LTPA authentication on both servers.

Regarding single sign-on, the LTPA token allows automatic login by preserving authentication information in a cookie. However, this capability does not provide the facility to perform an integrated sign-off. As an illustration, log in to Tivoli Integrated Portal to work with IBM Tivoli Netcool/OMNIBus, and then, open Tivoli Enterprise Portal to see the monitoring situation. Your LTPA token allows you to automatically log in to Tivoli Enterprise Portal. You then have a session with both Tivoli Integrated Portal and Tivoli Enterprise Portal. Assume you log out of the Tivoli Enterprise Portal session, but then, you decide that you must log back in. When you log back in, you get a separate LTPA token. At this state, when you open Tivoli Integrated Portal, you present the new LTPA token; however, Tivoli Integrated Portal already has your user ID in session (with the previous token). Therefore, Tivoli Integrated Portal does not allow you to log in because of the other session. You can mitigate the sign-off problem by using a session timeout for inactive sessions, or you can sign off explicitly from the application using a new browser window before re-invoking single sign-on.

4.2 Security setup overview

In this section, we discuss the common configuration needs for preparing and implementing security integration for Tivoli products. The discussion is divided into these topics:

- ▶ 4.2.1, “IBM Tivoli Directory Server implementation” on page 54
- ▶ 4.2.2, “Security setup considerations” on page 54
- ▶ 4.2.3, “Setting up LDAP authentication for federated repositories” on page 55
- ▶ 4.2.4, “Setting up single sign-on on multiple WebSphere cells” on page 55

4.2.1 IBM Tivoli Directory Server implementation

In our environment, we install IBM Tivoli Directory Server using the Middleware Installer from Tivoli Process Automation Engine. We install IBM Tivoli Directory Server on a separate server than the IBM Service Management product to simulate a common need from enterprises to install a stand-alone corporate directory server for all products.

You can obtain the documentation for IBM Tivoli Directory Server at this Web site:

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?toc=/com.ibm.IBMDS.doc/toc.xml>

You can obtain the middleware installer documentation at this Web site:

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.cmdb.doc_7.1.1/install/c_ccmdb_ccmdbcmiddlewareoverview.html

Note: Apart from IBM Tivoli Directory Server, you can use another implementation of LDAP, such as Microsoft® Active Directory®, or you can use the z/OS® Security Server. Your choice depends on the overall security strategy of your enterprise.

4.2.2 Security setup considerations

In this section, we discuss the procedure to enable LDAP authentication for WebSphere Application Server with federated repositories. This procedure is similar for any WebSphere Application Server V6.1-based product, even the embedded WebSphere server. However, certain products keep their own custom authentication to participate in the overall authentication, such as:

- ▶ IBM Tivoli Netcool/OMNIBus authentication for Netcool products
- ▶ Tivoli Enterprise Monitoring Server-based authentication for IBM Tivoli Monitoring

When you further evaluate how the products activate security, you see subtle differences in how the products configure security:

- ▶ Products have a preset realm name, such as ISMRealm or TIPrealm. Other products use the default realm name, such as DefaultWIMFileBasedRealm. In our environment, we use a custom realm called itsorealm to ensure that we modify and synchronize these realms. The realm information is critical for the SSO implementation.

- ▶ A product's security configuration might leave the default file-based authentication or remove the file authentication entry. This file-based authentication must not contain any user information, because a user entry must not reside in more than one repository.
- ▶ Products might use a specific basic entry mapping for the repository. This basic entry mapping allows a user to be custom-defined in the managing application. For example, IBM Tivoli Monitoring uses an entry mapping of `o=ITMSSOEntry` for the base suffix that we provide.

Non-WebSphere-based processes can also use federated repositories or LDAP for authentication by using individual authentication to LDAP or by using the external authentication services facility.

4.2.3 Setting up LDAP authentication for federated repositories

The following steps activate LDAP authentication for federated repositories. We discuss steps in detail for each individual product in other sections. These steps provide an overview:

1. Activate global security in WebSphere. You can activate or deactivate global security using the administration console or the **wsadmin** interface.

Note: In case you experience a problem after enabling the security and you cannot get into WebSphere, you can disable security by editing the `security.xml` file under the `profiles/config/cells/<cellname>` directory. Stop the Java process and restart WebSphere.

2. Modify the federated repositories setting to add an LDAP repository that points to the LDAP server with the appropriate credential.
3. Modify the suffix list for the federated repository to add the appropriate suffix and its mapping for the authentication.
4. Verify the object creation suffix for the creation interface so that new users and groups can be created using the security administrative interface.
5. Verify the object class and field identifier to use to identify users and groups.
6. Save the modification and restart WebSphere.

4.2.4 Setting up single sign-on on multiple WebSphere cells

Setting up single sign-on requires the following tasks:

1. Single sign-on requires setting LDAP authentication as discussed in 4.2.3, "Setting up LDAP authentication for federated repositories" on page 55.

2. You must set the realm for the federated repositories to the same realm for all participating servers. We decide in our environment to use the realm called itsorealm.
3. All servers participating in single sign-on must enable SSO and ensure that the domain name is the same. The LTPA cookie is passed as a session cookie at the Web browser. You can set the session cookie to be active only for a certain domain.
4. Servers participating in SSO must encrypt and decrypt the token using the same key. This key must be exported from one of the participants, and then, the exported key must be imported to all of the SSO participants.
5. WebSphere has a facility to automatically generate the SSO key. You must disable this facility. Otherwise, every time that the key is regenerated, it must be exported and then imported again.

4.3 Integrated Solution Console setup

Tivoli products that use the Integrated Solution Console as their management interface provide predefined menu options for you to set up security, manage users, and troubleshoot. These products include Tivoli Dynamic Workload Console, IBM Tivoli System Automation for Multiplatform, IBM Tivoli Usage and Accounting Manager, and others.

In this section, we discuss how to set up security from the Integrated Solution Console, including systems with embedded WebSphere or full function WebSphere Application Servers with administrative consoles. We divide this discussion into these topics:

- ▶ 4.3.1, “LDAP authentication setup” on page 56
- ▶ 4.3.2, “Single sign-on setup” on page 67

4.3.1 LDAP authentication setup

Integrated Solution Console provides an interactive means for configuring WebSphere Application Server:

- ▶ When the WebSphere Application Server is installed as a full product, either as a stand-alone application server or as a Network Deployment configuration, you install the Integrated Solution Console (isclite.ear) as the Administration server.
- ▶ When the IBM Tivoli product, such as Tivoli Dynamic Workload Console, Tivoli Usage and Accounting Manager, and Tivoli Integrated Portal, uses the

Integrated Solution Console for its user interface, you can interactively configure WebSphere Application Server.

For other embedded WebSphere Application Servers, you must perform configuration by using a command-line interface or a scripting interface through the **wsadmin** command. Tivoli products usually provide a script to configure security by invoking the **wsadmin** interface.

Several configuration options are available:

- ▶ When the application server is not secured, perform the security configuration wizard as discussed in step 2 of the following steps.
- ▶ When security is active but not using federated repositories with LDAP, start with configuring the federated repository in step 3 on page 61 of the following steps.
- ▶ When the federated repositories setting must be changed, you can reconfigure the system.

You must perform several steps to enable authentication using the Integrated Solution Console:

1. Set up and start the LDAP repository, which you can verify by using an LDAP browser.
2. Enable security and specify the main administrative user. We use the Security Configuration Wizard that is available from the Integrated Solution Console:
 - a. Click **Security** → **Secure administration, applications, and infrastructure** on the initial Welcome dialog, as shown in Figure 4-5 on page 58.

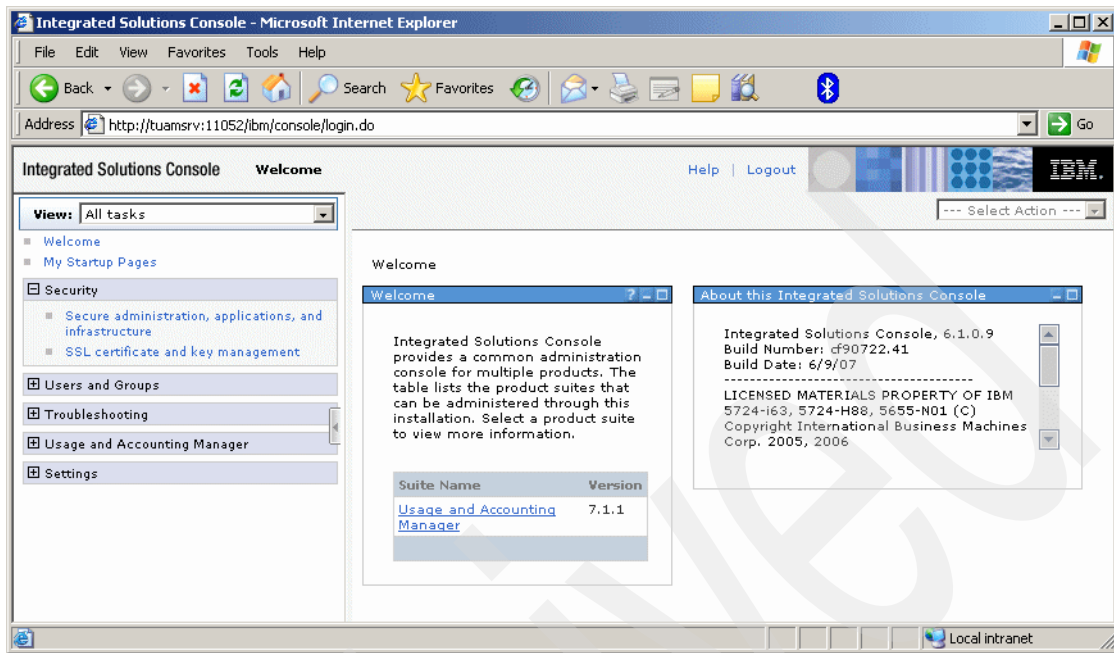


Figure 4-5 Integrated Solutions Console Welcome window

- b. From the security administration window in Figure 4-6 on page 59, click **Security Configuration Wizard**.

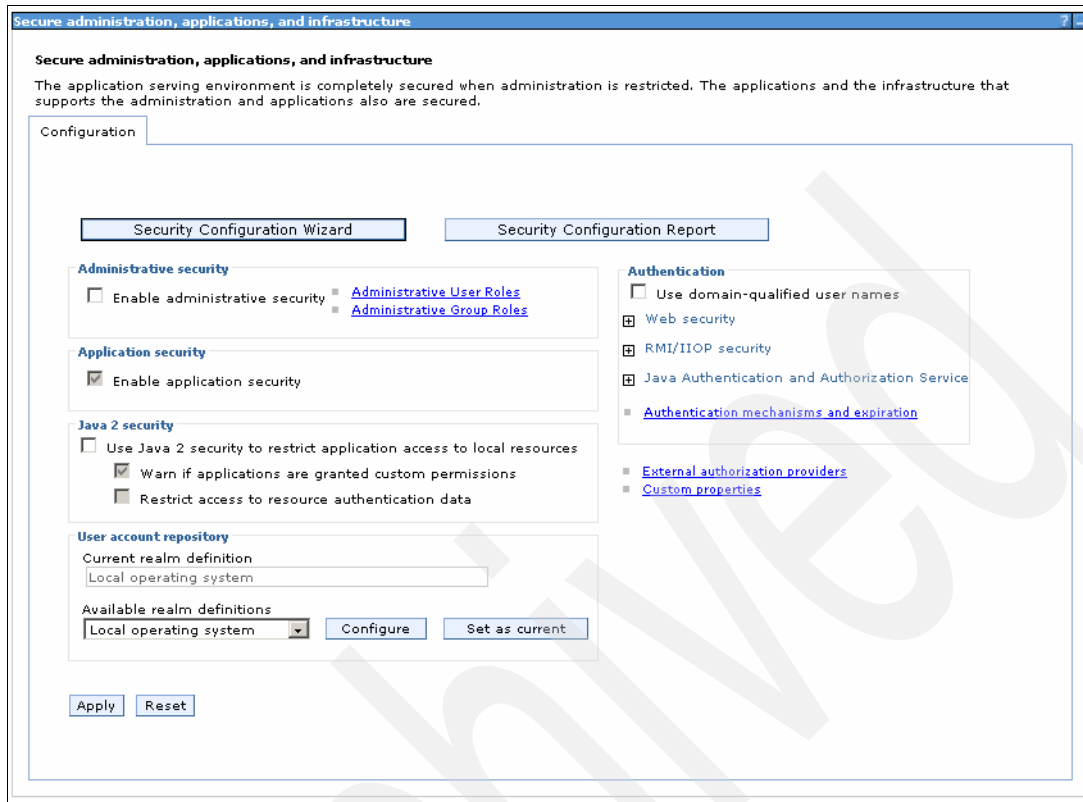


Figure 4-6 Secure administration, applications, and infrastructure window

- c. Figure 4-7 on page 60 shows the wizard. First, select **Enable application security**. Click **Next**.

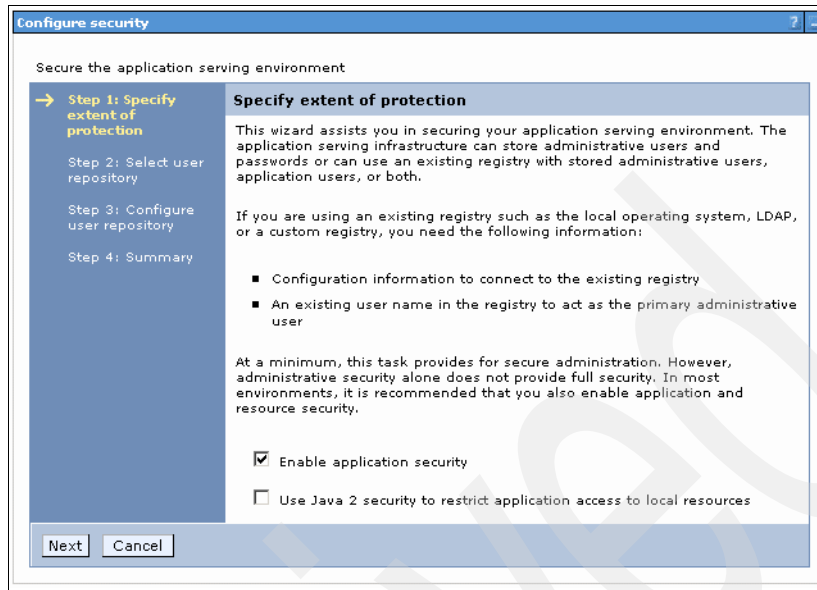


Figure 4-7 Configure security wizard: Step 1: Specify extent of protection

- d. In the second step in Figure 4-8, select **Federated repositories**, and click **Next**.

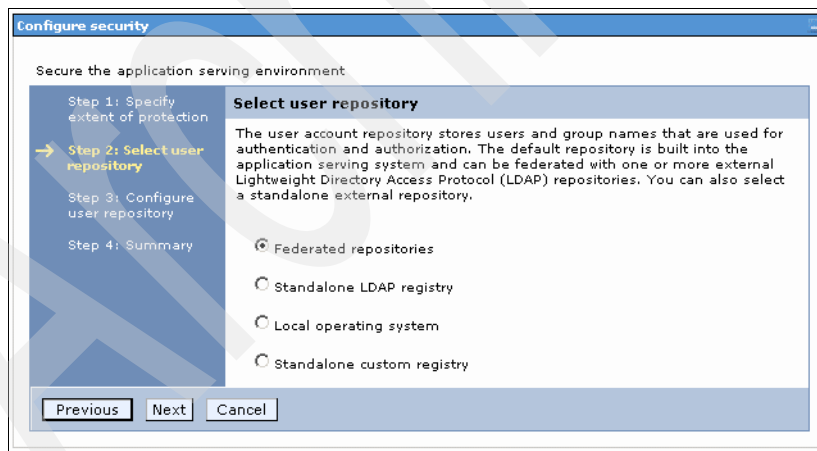


Figure 4-8 Configure security wizard: Step 2: Select user repository

- e. Figure 4-9 on page 61 allows you to assign the primary administrative user. Define this user ID in the default file-based repository. For our LDAP scenario, we must define this user separately in the LDAP repository. Click **Next**.

Configure security

Secure the application serving environment

Step 1: Specify extent of protection
Step 2: Select user repository
→ Step 3: Configure user repository
Step 4: Summary

Configure user repository

A secure, file-based user repository is built into the system for storing administrative users or environments with a small number of users. The file-based user repository can be federated with one or more external LDAP repositories. If this is the first time security has been enabled using this repository, provide a new user name and password to act as an administrator. If security was previously enabled using this repository, provide the name of a user with administrator privileges that is in the built-in repository.

Note: Use this panel to configure a federated repository with a built-in, file-based repository in the realm. To configure a federated repository with a non file-based repository in the realm, you must use the User accounts repository section on the Secure administration, applications, and infrastructure panel.

* Primary administrative user name
wasadmin

Password

Confirm password

Previous Next Cancel

Figure 4-9 Configure security wizard: Step 3: Configure user repository

- f. Figure 4-10 shows the summary operation and performs the necessary changes. Click **Finish**.

Configure security

Secure the application serving environment

Step 1: Specify extent of protection
Step 2: Select user repository
Step 3: Configure user repository
→ Step 4: Summary

Summary

Displays the list of values that are selected during the wizard and are used to enable security.

Options	Values
Enable administrative security	true
Enable application security	true
Use Java 2 security to restrict application access to local resources	false
User repository	Federated repositories
Primary administrative user name	wasadmin

Previous Finish Cancel

Figure 4-10 Configure security wizard: Step 4: Summary

3. Configure the Federated Repositories to include your repository:
 - a. Back in the secure administration dialog (Figure 4-6 on page 59), select **Federated repositories** from the Available realm definition list box, and

click **Set as current**. Make sure that the current realm definition is changed to Federated repositories.

- b. Click **Configure** to start configuring the repository. Figure 4-11 shows the configuration dialog.

Secure administration, applications, and infrastructure > Federated repositories

By federating repositories, identities stored in multiple repositories can be managed in a single, virtual realm. The realm can consist of identities in the file-based repository that is built into the system, in one or more external repositories, or in both the built-in repository and one or more external repositories.

Configuration

General Properties

* Realm name
defaultWIMFileBasedRealm

* Primary administrative user name
wasadmin

Server user identity

☒ Automatically generated server identity

☐ Server identity that is stored in the repository

Server user ID or administrative user on a Version 6.0.x node
Password

☒ Ignore case for authorization

Repositories in the realm:

Select	Base entry	Repository identifier	Repository type
☐	o=defaultWIMFileBasedRealm	InternalFileRepository	File

Additional Properties

- Property extension repository
- Entry mapping repository
- Supported entity types

Related Items

- Manage repositories

Apply OK Reset Cancel

Figure 4-11 Federated repositories configuration

- c. Under the Related Items heading, select **Manage repositories**. Figure 4-12 on page 63 shows the Manage repositories dialog. Click **Add** to add our LDAP repository.

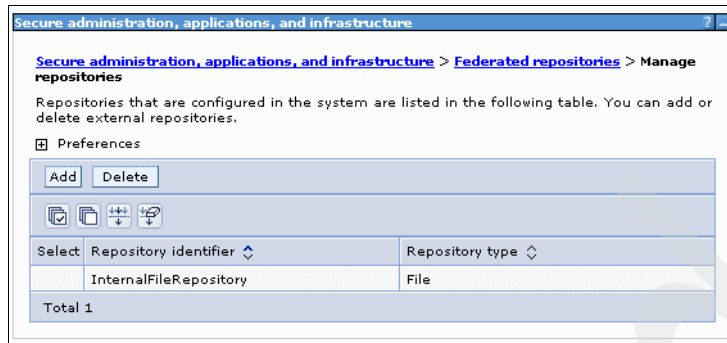


Figure 4-12 Manage repositories

- d. Figure 4-13 on page 64 shows the definition of an IBM Tivoli Directory Server V6 repository. Note these important parameters:
- Primary host name: We use our server host name.
 - Port: We use the default non-SSL port 389.
 - Bind distinguished name: Typically, this bind distinguished name is cn=root for IBM Tivoli Directory Server.
 - Bind password: We use the password to connect to the server.

Secure administration, applications, and infrastructure

[Secure administration, applications, and infrastructure](#) > [Federated repositories](#) > [Manage repositories](#) > New

Specifies the configuration for secure access to a Lightweight Directory Access Protocol (LDAP) repository with optional failover servers.

Configuration

General Properties

* Repository identifier
IDSITDS

LDAP server

* Directory type
IBM Tivoli Directory Server Version 6

* Primary host name
security1

Port
389

Failover server used when primary is not available:

Delete

Select	Failover host name	Port
	None	

Add

Support referrals to other LDAP servers
ignore

Security

Bind distinguished name
cn=root

Bind password

Login properties
uid

Certificate mapping
EXACT_DN

Certificate filter

☐ Require SSL communications

☒ Centrally managed

■ [Manage endpoint security configurations](#)

☒ Use specific SSL alias

NodeDefaultSSLSettings [SSL configurations](#)

The additional properties will not be available until the general properties for this item are applied or saved.

Additional Properties

- Performance
- LDAP entity types
- Group attribute definition

Apply OK Reset Cancel

Figure 4-13 Defining the repository

- e. To support single sign-on, all the user definitions must come from LDAP. The users in the file-based repository cannot participate in single sign-on and need to be migrated to LDAP. Deleting the default file-based repository from the federated list also helps to ensure that there will be no user duplication from LDAP and the file.
- f. From Figure 4-11 on page 62, click **Add Base entry to realm**. Figure 4-14 on page 65 lets you define a new realm definition. Define the appropriate suffix as defined in IBM Tivoli Directory Server. Click **OK**.

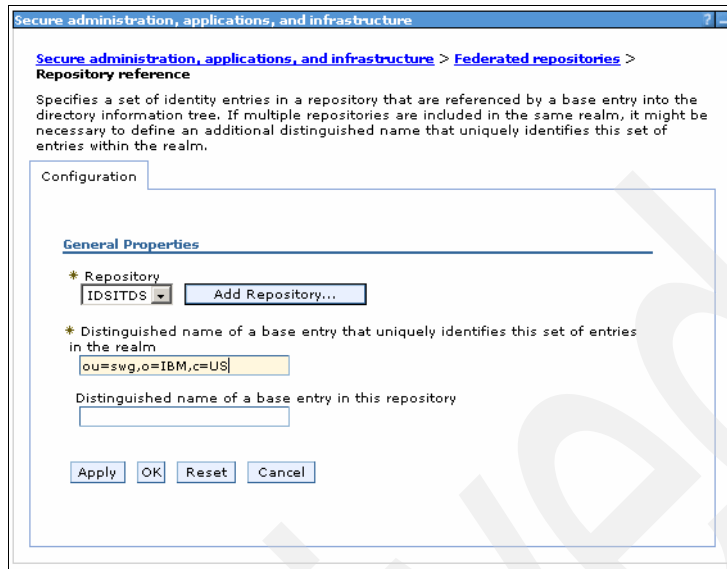


Figure 4-14 Adding suffix base entry

- g. Figure 4-15 on page 66 shows the final configuration of the federated repository. Click **OK**.

Secure administration, applications, and infrastructure

[Secure administration, applications, and infrastructure](#) > **Federated repositories**

By federating repositories, identities stored in multiple repositories can be managed in a single, virtual realm. The realm can consist of identities in the file-based repository that is built into the system, in one or more external repositories, or in both the built-in repository and one or more external repositories.

Configuration

General Properties

* Realm name
defaultWIMFileBasedRealm

* Primary administrative user name
wasadmin

Server user identity

☒ Automatically generated server identity

☐ Server identity that is stored in the repository

Server user ID or administrative user on a Version 6.0.x node

Password

☒ Ignore case for authorization

Repositories in the realm:

Select	Base entry	Repository identifier	Repository type
☐	ou=swg,o=IBM,c=US	IDSITDS	LDAP:IDS6

Additional Properties

- [Property extension repository](#)
- [Entry mapping repository](#)
- [Supported entity types](#)

Related Items

- [Manage repositories](#)

Figure 4-15 Federated repository setting summary

- h. Save the WebSphere configuration.
4. After you define the LDAP repository, you can verify user roles. You must ensure that there is an LDAP user that has the authority to log on to the Web application. Follow these steps:
 - a. From the leftmost menu in Figure 4-5 on page 58, select **Users and Groups** → **Manage users**. Click **Search** to make sure that you can see the users that are defined in the LDAP server.

- b. Select **Users and Groups** → **Administrative user role** to make sure that the LDAP user has the appropriate roles.
 - c. Save the changes to the WebSphere configuration.
5. Restart WebSphere Application Server.

After the application server authenticates through IBM Tivoli Directory Server, we can proceed to the next step, which is to define single sign-on (SSO).

4.3.2 Single sign-on setup

In this section, we illustrate how to enable various application servers to participate in a single sign-on scenario. The setup consists of exporting and importing Lightweight Third Party Authentication (LTPA) keys and enabling LTPA authentication. Setting up SSO with LTPA includes the following steps:

1. In WebSphere Application Server, enabling the security with the Security Configuration Wizard generates the initial LTPA key. This key then automatically regenerates.
2. To prevent the LTPA key from becoming invalid due to automatic regeneration, we must disable automatic key generation:
 - a. Go to **Security** → **SSL certificate and key management**. See Figure 4-16 on page 68.

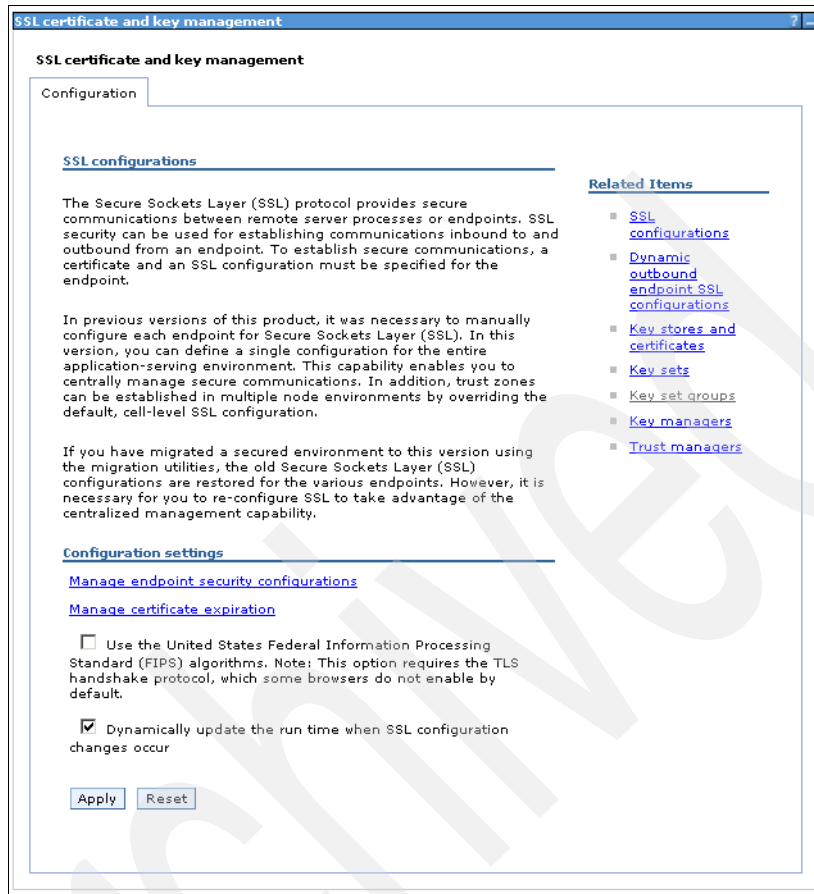


Figure 4-16 SSL and certificate and key management

- b. On Figure 4-16, under the Related Items heading, click **Key set groups**. Refer to Figure 4-17 on page 69.

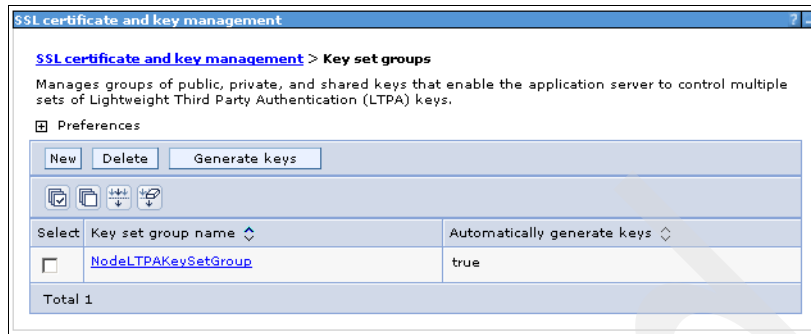


Figure 4-17 Key set groups

- c. Click **NodeLTPAKeySetGroup**.
- d. On Figure 4-18 on page 70, under the Key generation heading, clear the Automatically generate keys check box, and click **OK**.

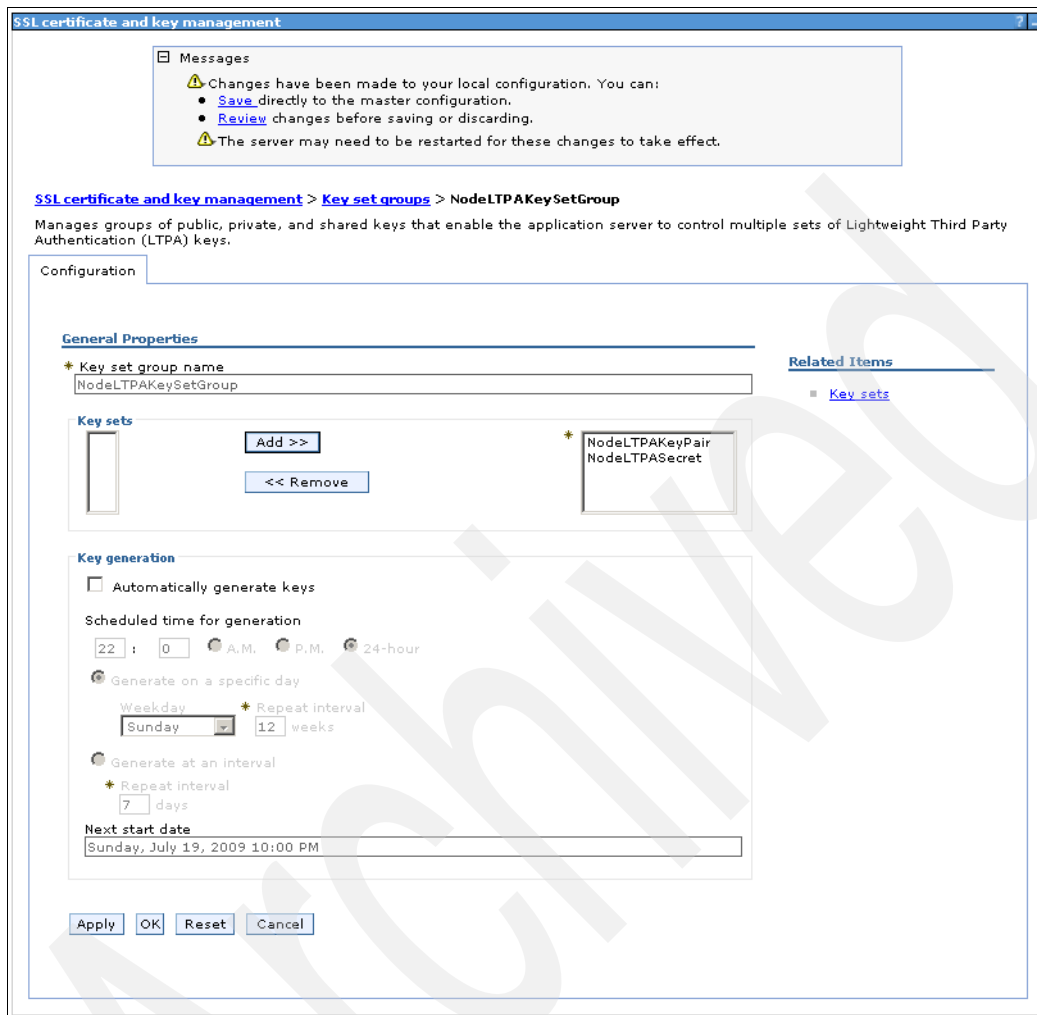


Figure 4-18 Disabling automatic key generation

3. To set up the SSO options, on Figure 4-5 on page 58, in the leftmost list of options, click **Security** → **Secure administration, applications, and infrastructure** → **Web security** → **single sign-on (SSO)**.
4. In Figure 4-19 on page 71, select **Enabled** to select SSO, and type the domain name that you want to SSO to cover. All servers that participate in SSO must be addressed with its fully qualified host name and have the domain name as the suffix.

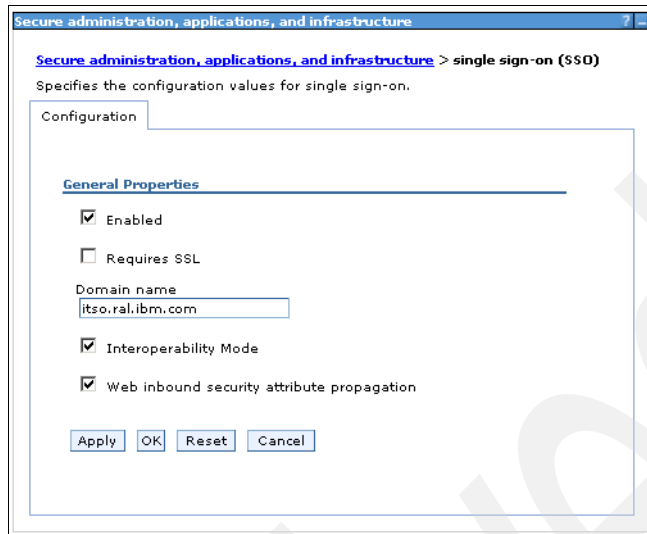


Figure 4-19 SSO settings

5. Check your realm definition. We use the Federated Repositories for user authentication. The default realm name is DefaultWIMFileBasedRealm. You can find the realm definition and change it by clicking **Security** → **Secure administration, applications, and infrastructure** and clicking **Configure** beside the selected authentication method. Figure 4-20 shows the realm name.

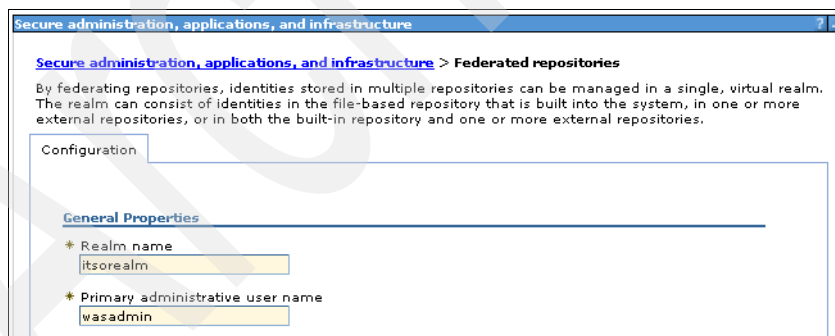


Figure 4-20 Checking realm name

6. When SSO is enabled, the LTPA keys must be extracted from a source and imported to all participating application servers. The following steps export the LTPA key into a file. Perform this task *only once* in your environment:
 - a. Go to **Security** → **Secure administration, applications, and infrastructure** → **Authentication mechanisms and expiration**. See Figure 4-21.

The screenshot displays a web-based configuration interface for 'Secure administration, applications, and infrastructure'. The main heading is 'Authentication mechanisms and expiration'. Below this, there is a 'Configuration' tab. The interface is divided into three main sections:

- Key generation:** This section explains that authentication data is encrypted and decrypted using keys from one or more key stores. It features a 'Key set group' dropdown menu currently set to 'NodeLTPAKeySetGroup', a 'Generate keys' button, and a link to 'Key set groups'.
- Authentication expiration:** This section states that authentication information persists for a limited time. It includes two input fields: 'Authentication cache timeout' (set to 10 minutes and 0 seconds) and 'Timeout value for forwarded credentials between servers' (set to 720 minutes).
- Cross-cell single sign-on:** This section describes how single sign-on can be provided by sharing keys and passwords. It contains two password fields labeled 'Password' and 'Confirm password', both masked with dots. Below these is a 'Fully qualified key file name' input field with the value '/ltpa-keys'. There are 'Import keys' and 'Export keys' buttons. At the bottom of this section is a checkbox labeled 'Use SWAM-no authenticated communication between servers' which is currently unchecked.

At the very bottom of the configuration area, there are four buttons: 'Apply', 'OK', 'Reset', and 'Cancel'.

Figure 4-21 LTPA key export

- b. Export the key by entering the password and target or fully qualified key file name. Click **Export keys** to generate the key. The result is similar to Example 4-1 on page 73. Make sure that you have the correct realm name for `com.ibm.websphere.ltpa.Realm`.

Example 4-1 Sample exported LTPA key

```
#IBM WebSphere Application Server key file
#Tue Apr 28 00:48:00 EDT 2009
com.ibm.websphere.CreationDate=Tue Apr 28 00\:48\:00 EDT 2009
com.ibm.websphere.ltpa.version=1.0
com.ibm.websphere.ltpa.3DESKey=NXJLFp4TYR19T5ebefc0cG0/DpSS7iyDRd
VD6++93pY\=
com.ibm.websphere.CreationHost=tnmip.itso.ral.ibm.com
com.ibm.websphere.ltpa.PrivateKey=XFNooaCxUqBuF4BFtTY1d9spfs/vQbk
JAA1NhQJ2pR92KPSm8CDaApzFztvmPza2wxHLNng9s0ygyWGKx439aI7btYjwf5GJR
n2J5ATlAdaBjepKgnu0xhwG08k3YikW6/HIUJr9VU89KFKBzJMIkcMKsux0KDFfJ6
UZ8kIlvbu1ufQZKmbA7S0ZsqumZf8dM+vu64KZ8VyjbqTVPPrKCQcb0BliEXAWOD
di6U2UDNusRcGeit/Ppv5Bfoc9AaV1x2Rz+Mot44skueCf0Kp3Mt5th9YaiginC43
RRusuN21YrpR+w+2069YorXN01+k+5gFYegLSycXzvhZx2SQ/CKH0ggX4ZaBJVcCF
gNjKPxU\=
com.ibm.websphere.ltpa.Realm=itsorealm
com.ibm.websphere.ltpa.PublicKey=AOn11d3UbrZCvy8hL9drnebPs6z3wf1Y
YvERlmMJtLDqmVQz5orWYf409CaygZS/XTBmrBI fY7J1PDr/3XyZEq30eBVqeUamN
H0gwCkORsKrT7quTnFOHKRJEGb6i6UkFOYHhDo/B+r7+ULN4+5B4pIGGb3XnhOepu
Cvg03a6I03AQAB
```

7. All other application servers that participate in the SSO environment must get the exported file and import the key pairs into their LTPA key. You use the same page (Figure 4-21 on page 72) to import the key. The key is actually stored in the `ltpa.jceks` file. This file is a Java Cryptography Extension (JCE) keystore that can be managed by the `keytool` utility to ensure that importing the key is successful.
8. After the import is successful, restart WebSphere Application Server and start testing SSO.

4.4 Tivoli Process Automation Engine security setup

The two major IBM Service Management components with respect to security are the process environment hosted inside the Java 2 Platform, Enterprise Edition (J2EE) WebSphere Application Server runtime environment and the Discovery Server, which is referred to as IBM Tivoli Application Dependency Discovery Manager.

IBM Service Management products are hosted inside a J2EE WebSphere environment and rely on the facilities that the WebSphere Application Server provides.

There are two major components that are relevant for IBM Service Management security: federated repositories and external authentication services. These WebSphere-based security components provide authentication and authorization services to WebSphere-based applications.

In this section, we discuss the necessary setup in the Tivoli Process Automation Engine using WebSphere Virtual Member Manager. The discussion includes the following steps:

- ▶ 4.4.1, “LDAP configuration in WebSphere Application Server” on page 74
- ▶ 4.4.2, “Single sign-on configuration in WebSphere Application Server” on page 84
- ▶ 4.4.3, “VMMSYNC configuration” on page 87

4.4.1 LDAP configuration in WebSphere Application Server

The first step for applications to participate in single sign-on is to configure them to be authenticated through a single Lightweight Directory Access Protocol (LDAP) server. In this section, we discuss how to configure WebSphere server to be authenticated through an LDAP server.

We use the federated repository type of registries. Federated repository allows us to use multiple repositories, such as a file-based repository. WebSphere federated repositories as a common model can securely access a file-based repository for authentication.

For more information about WebSphere Application Server security, refer to *IBM WebSphere Application Server V6.1 Security Handbook*, SG24-6316.

Follow these steps to configure LDAP authentication in WebSphere Application Server:

1. Log in to the WebSphere Application Server administrative console, and then, navigate to **Security** → **Secure Administration, applications, and infrastructure**.
2. Locate the User account repository section, choose **Federated repositories** from the Available Realm definitions list, and click **Configure**.
3. Click **Manage repositories** under the Related Items section.
4. Click **Add**. Figure 4-22 on page 75 shows the panel for configuring the LDAP repository.

Secure administration, applications, and infrastructure Close page

Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure > Federated repositories > Manage repositories > ISMITDS

Specifies the configuration for secure access to a Lightweight Directory Access Protocol (LDAP) repository with optional failover servers.

Configuration

General Properties

Repository identifier

ISMITDS

LDAP server

Directory type

IBM Tivoli Directory Server Version 6

Primary host name

security1

Port

389

Failover server used when primary is not available:

Delete

Select

Failover host name

Port

None

Add

Support referrals to other LDAP servers

ignore

Security

Bind distinguished name

cn=root

Bind password

Login properties

Certificate mapping

EXACT_DN

Certificate filter

☐ Require SSL communications

☒ Centrally managed

Manage endpoint security configurations

☒ Use specific SSL alias

CellDefaultSSLSettings

SSL configurations

Additional Properties

Performance

LDAP entity types

Group attribute definition

Apply

OK

Reset

Cancel

Figure 4-22 WebSphere Application Server repository properties

Table 4-1 on page 76 describes the options.

Chapter 4. Security integration 75

Table 4-1 WebSphere Application Server repository configuration

Setting	Value	Remark
Repository Identifier	ISMITDS	Enter any unique repository identifier.
Directory type	IBM Tivoli Directory Server Version 6	The software that we use for LDAP server
Primary hostname	security1.itso.ral.ibm.com	Enter the fully qualified name of the machine where IBM Tivoli Directory Server installed.
Port	389	Default port of LDAP server
Support referrals to other LDAP servers	Ignore	N/A
Bind distinguished name	cn=root	Enter the authoritative distinguished name used to authenticate LDAP connection.
Bind password	itso4you	Enter the password of the authoritative distinguished name.
Login properties	N/A	Leave this field blank.
Certificate mapping	EXACT_DN	This field will map X.509 certificates into an LDAP directory by exact distinguished name.

5. Click **Apply**, and then, click **Save** at the top.
6. Go back to the Federated repositories panel (Figure 4-15 on page 66), and click **Add Base entry to Realm** in Repositories in the realm: table.
7. Choose the repository identifier that you have just created, and enter the following values as stated in Table 4-2.

Table 4-2 Base entry panel

Setting	Value	Remark
Repository	ISMITDS	Enter the repository identifier that you have just created.
Distinguished name of a base entry	ou=SWG,o=IBM,c=US	Enter the base entry of the directory information tree identified for this specific realm.
Distinguished name of a base entry in this repository	ou=SWG,o=IBM,c=US	Enter the base entry of the directory information tree as defined in your LDAP server.

Figure 4-23 shows the base entry panel for the specific repository identifier.

Secure administration, applications, and infrastructure [Close page](#)

Secure administration, applications, and infrastructure

[Secure administration, applications, and infrastructure](#) > [Federated repositories](#) > ou=SWG,o=IBM,c=US

Specifies a set of identity entries in a repository that are referenced by a base entry into the directory information tree. If multiple repositories are included in the same realm, it might be necessary to define an additional distinguished name that uniquely identifies this set of entries within the realm.

Configuration

General Properties

* Repository
ISMITDS [Add Repository...](#)

* Distinguished name of a base entry that uniquely identifies this set of entries in the realm
ou=SWG,o=IBM,c=US

Distinguished name of a base entry in this repository
ou=SWG,o=IBM,c=US

[Apply](#) [OK](#) [Reset](#) [Cancel](#)

Figure 4-23 Repository identifier configuration

8. Click **Apply**, and then, click **Save** at the top.
9. Go back to the Federated repositories panel (Figure 4-15 on page 66) and enter the rest of the information as stated in Table 4-3.

Table 4-3 Federated repositories configuration

Setting	Value	Remark
Realm name	itsorealm	For SSO purposes, this value must be the same as the configurations of other applications that participate in SSO.
Primary administrative user name	wasadmin	This value must be a valid user in the LDAP repository.

10. In the Server user identity section, choose **Automatically Generated server identity**.
11. Select **Ignore case for authorization**.
12. Figure 4-24 on page 78 shows the final configuration of the federated repositories.

Secure administration, applications, and infrastructure Close page

Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure > Federated repositories

By federating repositories, identities stored in multiple repositories can be managed in a single, virtual realm. The realm can consist of identities in the file-based repository that is built into the system, in one or more external repositories, or in both the built-in repository and one or more external repositories.

Configuration

General Properties

* Realm name
itsorealm

* Primary administrative user name
wasadmin

Server user identity

☒ Automatically generated server identity

☐ Server identity that is stored in the repository

Server user ID or administrative user on a Version 6.0.x node

Password

☒ Ignore case for authorization

Repositories in the realm:

Select	Base entry	Repository identifier	Repository type
☐	ou=SWG,o=IBM,c=US	ISMITDS	LDAP:IDS6

Additional Properties **Related Items**

☐ [Property extension repository](#)
☐ [Manage repositories](#)

☐ [Entry mapping repository](#)

☐ [Supported entity types](#)

Figure 4-24 Federated repositories panel

13. Click **Apply**, and then, click the **Save** link.

Also, we must configure the repository entity types that are supported by federated repositories. It is important to configure the repository entity types correctly, because the repository entity types map the WebSphere Application Server repository to the LDAP repository. We need the repository entity types for configuring users and groups in the administrative console.

Table 4-4 on page 79 shows the entity types that are supported in WebSphere Application Server.

Table 4-4 Supported entity types of WebSphere Virtual Member Manager

Entity type	Description
PersonAccount	Data object in federated repositories to support user entries in repositories that combine person and account information
Group	Data object in federated repositories to support a collection of entities. It can be a group of groups, persons, accounts, and so forth.
OrgContainer	Data object in federated repositories to provide a container for organizational unit

Follow these steps:

1. In Figure 4-24 on page 78, in the Additional properties section, click **Supported entity types**.
2. Click **PersonAccount**, and enter the information as stated in Table 4-5.

Table 4-5 PersonAccount entity type configuration

Setting	Value	Remark
Base entry for the default parent	ou=users,ou=SWG,o=IBM,c=US	Parent base entry of user directory as configured in your LDAP server
Relative Distinguished Name properties	uid	LDAP attribute that is used for identifying the user name

Figure 4-25 on page 80 shows the final configuration of the PersonAccount entity type.

Secure administration, applications, and infrastructure Close page

Secure administration, applications, and infrastructure > Federated repositories > ou=SWG,o=IBM,c=US > Supported entity types > PersonAccount

Use this page to configure entity types that are supported by the member repositories.

Configuration

General Properties

- * Entity type
PersonAccount
- * Base entry for the default parent
ou=users,ou=SWG,o=IBM,c=US
- * Relative Distinguished Name properties
uid

Apply OK Reset Cancel

Figure 4-25 PersonAccount entity type panel

3. In Figure 4-25, click **Apply**, and then, click the **Save** link.
4. On the Supported entity types panel, click the **Group** link, and enter information as stated in Table 4-6.

Table 4-6 Group entity type configuration

Setting	Value	Remark
Base entry for the default parent	ou=groups,ou=SWG,o=IBM,c=US	Parent base entry of group directory as configured in your LDAP server
Relative Distinguished Name properties	cn	LDAP attribute that is used for identifying the group name

Figure 4-26 on page 81 shows the configuration of the Group entity type.

Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure > Federated repositories > ou=SWG,o=IBM,c=US > Supported entity types > Group

Use this page to configure entity types that are supported by the member repositories.

Configuration

General Properties

* Entity type
Group

* Base entry for the default parent
ou=groups,ou=SWG,o=IBM,c=US

* Relative Distinguished Name properties
cn

Apply OK Reset Cancel

Figure 4-26 Group entity type panel

- Click **Apply**, and then, click the **Save** link at the top.
- On the Supported entity types panel, click the **OrgContainer** link, and enter the information as stated in Table 4-7.

Table 4-7 OrgContainer entity type configuration

Setting	Value	Remark
Base entry for the default parent	ou=SWG,o=IBM,c=US	Parent base entry of organization container directory as configured in your LDAP server
Relative Distinguished Name properties	o;ou;dc;cn	LDAP attributes that are used for identifying the organizational unit

Figure 4-27 on page 82 shows the final configuration of the OrgContainer entity type.

Figure 4-27 *OrgContainer* entity type panel

7. Click **Apply**, and then, click the **Save** link at the top.
8. Navigate back to the Secure administration, applications, and infrastructure panel and complete the following steps:
 - a. Make sure that Administrative security is enabled. If not, select **Enable administrative security**.
 - b. Make sure that Application security is enabled. If not, select **Enable application security**.
 - c. Clear **Use Java 2 security** to restrict application access to local resources.
 - d. From the Available realm definition, select **Federated repositories** and click **Set as current**.
9. Click **Apply**, and then, click the **Save** link at the top.
10. Restart WebSphere Application Server Deployment Manager, Node Agent, and Application Server.

After you successfully configure LDAP, WebSphere Application Server now has a view of all of the users and the groups that are defined in the LDAP server.

If you go back to the WebSphere Application Server administrative console, navigate to **Users and Groups** → **Manage Users**, and click **Search**, you can see the list of users as defined in the LDAP server. Figure 4-28 on page 83 shows a sample list of users.

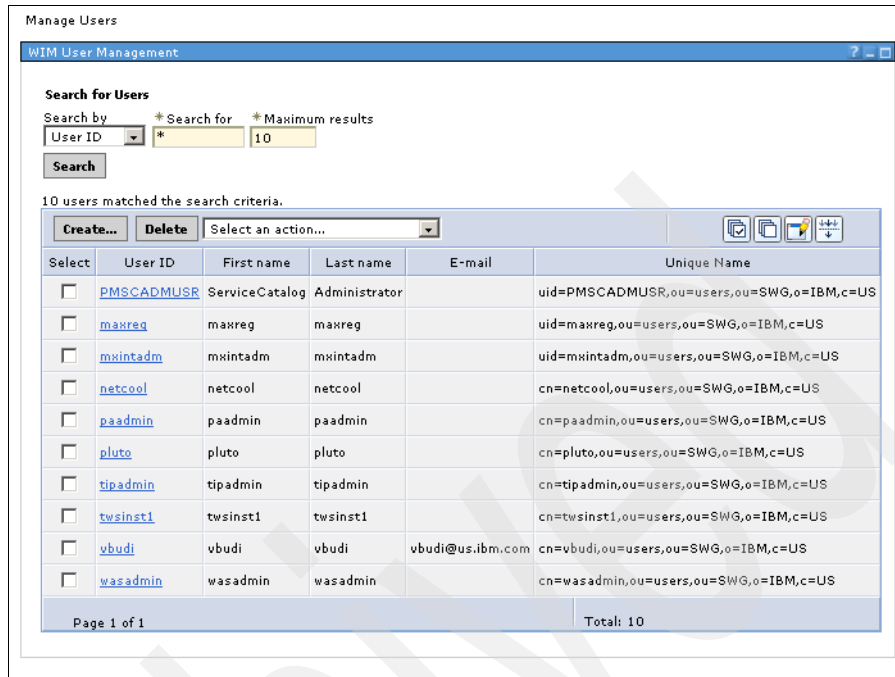


Figure 4-28 WebSphere Application Server Manage Users panel

Similarly for groups, you can see the list of groups as defined in the LDAP server by navigating to **Users and Groups** → **Manage Groups** and clicking **Search**. Figure 4-29 on page 84 shows a sample list of groups.

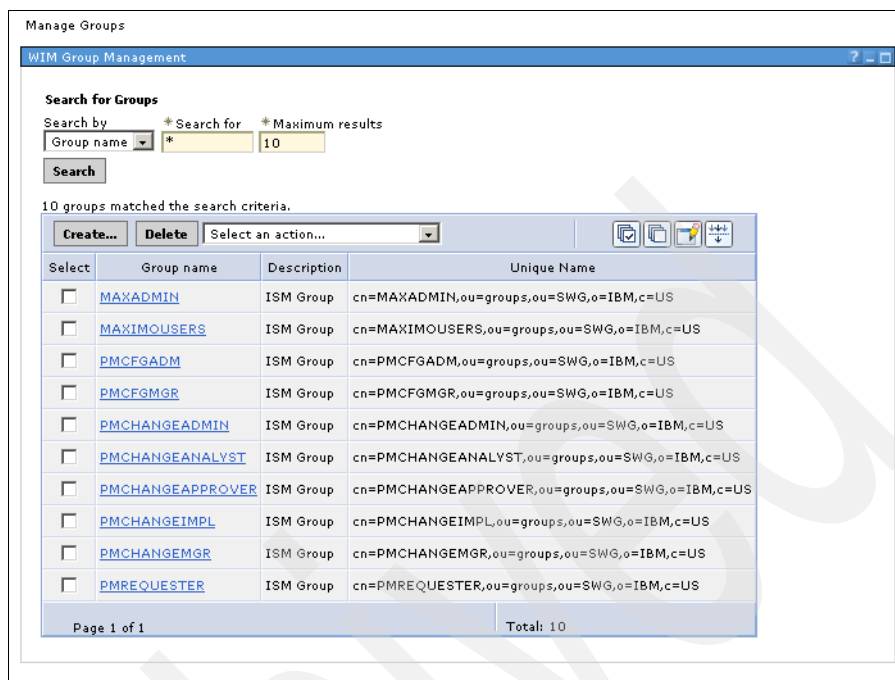


Figure 4-29 WebSphere Application Server groups panel

4.4.2 Single sign-on configuration in WebSphere Application Server

Single sign-on represents an authentication process where a user (or client) is validated one time and is subsequently identified to all resources and applications within the single sign-on domain.

In our implementation, we use a Lightweight Third Party Authentication (LTPA) token to achieve single sign-on. After the user is authenticated by WebSphere Application Server, the LTPA token, in the form of a cookie, is created and sent to the browser. The browser returns the cookie on subsequent requests so that other applications within the same TCP domain can recognize the user.

Several important considerations exist in the implementation of LTPA-based single sign-on in the WebSphere Application Server environment:

- TCP domain

The LTPA token is really a domain cookie. All applications that want to participate in single sign-on must reside in the same TCP domain. For example, in our case, we use `itso.ral.ibm.com` as the common domain.

► LTPA key

The LTPA key is used as a shared key to encrypt the LTPA token using the triple Data Encryption Standard (3DES) algorithm. The LTPA key must be imported into the configuration of all applications that participate in single sign-on, which makes sure that the LTPA token was created by a trusted source because the token can be decrypted by the same key. When the LTPA key is generated, it is protected by a password.

► Realm name

Because the federated repositories function is used, all Tivoli products that run on top of WebSphere have to be in the same virtual realm name to make sure that one application can recognize users from other applications. This rule is a WebSphere requirement rather than an LTPA requirement.

► Machine time

People often overlook the machine system time. The default implementation specifies that the LTPA token expires in 120 minutes. If the single sign-on environment consists of several machines running independently, you must ensure that the system times of all of the machines are synchronized.

Based on this information, we are now ready to configure the single sign-on function in WebSphere Application Server:

1. Log in to the WebSphere Application Server administrative console, and then, navigate to **Security** → **Secure administration, applications, and infrastructure**.
2. From the Authentication section, click **Authentication mechanism and expiration**.
3. In the Cross-cell single sign-on section, enter the security password, and enter the fully qualified key file name of the common LTPA key file. Figure 4-30 on page 86 shows the final configuration on the Authentication mechanisms and expiration panel.

In the Authentication expiration section, you will notice that the timeout value for forwarded credentials between servers is 120 minutes. This value defines how long the LTPA cookie that was created by WebSphere exists before it expires.

Note: The LTPA key file has to be copied into the local machines where WebSphere Application Server runs. In the previous setting, the file is copied into the C:\ directory.

Secure administration, applications, and infrastructure Close page

Secure administration, applications, and infrastructure > Authentication mechanisms and expiration

Encrypts authentication information so that the application server can send the data from one server to another in a secure manner. The encryption of authentication information that is exchanged between servers involves the LTPA mechanism.

Configuration

Key generation

Authentication data is encrypted and decrypted by using keys that are kept in one or more key stores.

Key set group

☐ [Key set groups](#)

Authentication expiration

Authentication information persists in the system for a limited amount of time before it expires and must be refreshed.

Authentication cache timeout
 minutes seconds

Timeout value for forwarded credentials between servers
 minutes

Cross-cell single sign-on

Single sign-on across cells can be provided by sharing keys and passwords. To share the keys and password, log on to one cell, specify a key file, and click Export keys. Then, log on to the other cell, specify the key file, and click Import keys.

* Password

* Confirm password

Fully qualified key file name

Figure 4-30 Authentication mechanisms and expiration panel

4. Click **Apply**, and then, click the **Save** link at the top.
5. Go back to the Secure administration, applications, and infrastructure panel. Click **Web Security** in the Authentication section.
6. Click **Single sign-on (SSO)**.
7. Make sure that the **Enabled** check box is chosen, clear the **Required SSL** check box, and in the Domain name section, enter your TCP domain name. In our case, it is `itso.ra1.ibm.com`. Figure 4-31 on page 87 shows the final configuration of the single sign-on function.

Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure > single sign-on (SSO)

Specifies the configuration values for single sign-on.

Configuration

General Properties

☒ Enabled

☐ Requires SSL

Domain name
itso.ral.ibm.com

☒ Interoperability Mode

☒ Web inbound security attribute propagation

Apply OK Reset Cancel

Figure 4-31 Single sign-on panel

8. Click **Apply**, and then, click the **Save** link at the top.
9. Restart WebSphere Application Server Deployment Manager, Node Agent, and Application Server.

4.4.3 VMMSYNC configuration

VMMSYNC is a cron task that is included in the Tivoli Process Automation Engine installation. It utilizes WebSphere federated repositories application programming interfaces (APIs) to populate database tables with user group and group membership records. Figure 4-32 on page 88 depicts the mechanism of the VMMSYNC cron task to populate the user tables. This mechanism ensures that the status of the system authentication and application authentication mechanism synchronizes the user and group information.

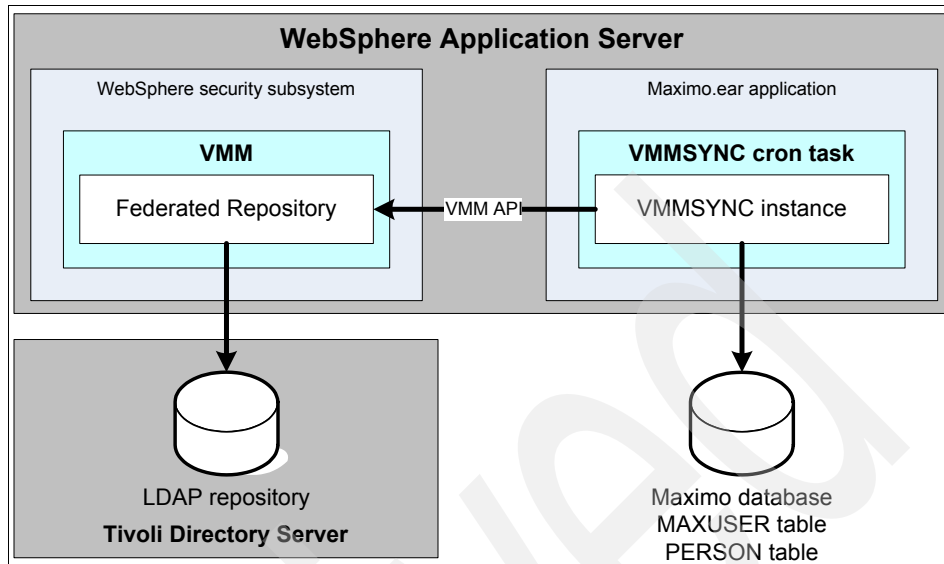


Figure 4-32 VMMSYNC cron task connectivity diagram

We configure WebSphere Application Server to authenticate through LDAP using the federated repositories federated repository. Tivoli Process Automation Engine use a module of the VMMSYNC cron task to synchronize the users and groups with the LDAP registry. Therefore, the VMMSYNC cron task plays an extremely important role in authentication for Tivoli Process Automation Engine-based products, such as IBM Tivoli Service Request Manager or IBM Tivoli Provisioning Manager. Make sure that the VMMSYNC cron task is always operational.

Follow these steps to configure the VMMSYNC cron task in IBM Tivoli Provisioning Manager Login to the Maximo console using **maxadmin**:

1. Navigate to **Go To** → **System Configuration** → **Platform Configuration** → **Cron Task Setup**:
2. Press Enter to list all of the available Cron Task instances.
3. Click **VMMSync** to open its configuration panel as shown in Figure 4-33 on page 89.

The screenshot shows the 'Cron Task Setup' window in the IBM Maximo application. The 'Cron Task' is 'VMMSYNC' with the description 'Invokes WebSphere VMM APIs to populate data'. The 'Class' is 'psdi.security.vmm.VMMSyncCronTask' and the 'Access Level' is 'FULL'. Below this, the 'Cron Task Instances' section shows a table with one instance, 'VMMSYNC01', with a schedule of '15m,*****', running as 'MAXADMIN', and with 'Active?' and 'Keep History?' checkboxes checked. The 'Max Number of History Records' is set to 5. Below the instances, the 'Parameters' section shows a table with five parameters: 'Credential' (value: '*****'), 'GroupMapping' (value: '<?xml version="1.0" encoding="UTF-8" ?><IDO'), 'GroupSearchAttribute' (value: 'cn'), 'Principal' (value: 'cn=wasadmin,ou=users,ou=SWVG,o=IBM,c=US'), and 'SynchAdapter' (value: 'psdi.security.vmm.DefaultVMMSyncAdapter').

Cron Task	Class	Access Level
VMMSYNC	psdi.security.vmm.VMMSyncCronTask	FULL

Cron Task Instance Name	Schedule	Run as User	Active?	Keep History?	Max Number of History Records
VMMSYNC01	15m,*****	MAXADMIN	☒	☒	5

Parameter	Value	Description
Credential	*****	VMM admin credentials.
GroupMapping	<?xml version="1.0" encoding="UTF-8" ?><IDO	The USER XML used by the VMM task.
GroupSearchAttribute	cn	VMM search attribute to query group records.
Principal	cn=wasadmin,ou=users,ou=SWVG,o=IBM,c=US	VMM admin principal.
SynchAdapter	psdi.security.vmm.DefaultVMMSyncAdapter	VMM synchronization adapter.

Figure 4-33 VMMSYNC configuration

Perform these tasks:

- In the Cron Task Instances section, select the **Active?** and **Keep History?** check boxes.
 - In the Cron Task Parameters section, select **Credential** for the parameter and enter the password of the WebSphere wasadmin ID as defined in the LDAP server.
4. Press Ctrl+the right arrow key or click the right arrow button to go to the next parameters page.
 5. Expand the UserMapping parameter by clicking the arrow on the left side. You use UserMapping to map the user entries in the LDAP repository into the MAXUSER and PERSON tables in the Maximo database. Any attribute entries that are missing in the mapping can cause VMMSYNC to fail. In our setup, we do not have PHONE and EMAIL detail in our LDAP directory; therefore, we change the XML value by removing the following sections:

```
<table allowdelete="true" name="PHONE">...</table>
<table allowdelete="true" name="PHONE">...</table>
<table allowdelete="true" name="EMAIL">...</table>
```

Example 4-2 on page 90 shows our UserMapping configuration.

Example 4-2 Sample of a working UserMapping configuration

```
<?xml version="1.0" encoding="UTF-8" ?><!DOCTYPE ldapsync SYSTEM
"ldapuser.dtd">
<ldapsync>
  <user>
    <basedn>ou=users,ou=SWG,o=IBM,c=US</basedn>
    <filter>PersonAccount</filter>
    <scope>subtree</scope>
    <attributes>
      <attribute>uid</attribute>
      <attribute>givenName</attribute>
      <attribute>sn</attribute>
      <attribute>displayName</attribute>
      <attribute>street</attribute>
      <attribute>telephoneNumber</attribute>
      <attribute>mail</attribute>
      <attribute>st</attribute>
      <attribute>postalCode</attribute>
      <attribute>c</attribute>
      <attribute>l</attribute>
    </attributes>
    <datamap>
      <table name="MAXUSER">
        <keycolumn name="USERID" type="UPPER">uid</keycolumn>
        <column name="LOGINID" type="ALN">uid</column>
        <column name="PERSONID" type="UPPER">uid</column>
        <column name="STATUS" type="UPPER">{ACTIVE}</column>
        <column name="TYPE" type="UPPER">{PRIMARY}</column>
        <column name="QUERYWITHSITE" type="YORN">{1}</column>
        <column name="FORCEEXPIRATION" type="YORN">{0}</column>
        <column name="FAILEDLOGINS" type="YORN">{0}</column>
        <column name="PASSWORD" type="CRYPTO">{0}</column>
        <column name="MAXUSERID"
type="INTEGER">{:uniqueid}</column>
        <column name="SYSUSER" type="YORN">{0}</column>
        <column name="INACTIVESITES"
type="YORN">{0}</column><column name="SCREENREADER"
type="YORN">{0}</column>
      </table>
      <table name="PERSON">
        <keycolumn name="PERSONID" type="UPPER">uid</keycolumn>
        <column name="FIRSTNAME" type="ALN">givenName</column>
        <column name="LASTNAME" type="ALN">sn</column>
        <column name="DISPLAYNAME" type="ALN">displayName</column>
```



```

        <column name="ADDRESSLINE1" type="ALN">street</column>
        <column name="STATEPROVINCE" type="ALN">st</column>
        <column name="CITY" type="ALN">l</column>
        <column name="POSTALCODE" type="ALN">postalCode</column>
        <column name="COUNTRY" type="ALN">c</column>
        <column name="STATUS" type="UPPER">{ACTIVE}</column>
        <column name="TRANSEMAILELECTION"
type="UPPER">{NEVER}</column>
        <column name="STATUSDATE" type="ALN">{:sysdate}</column>
        <column name="ACCEPTINGWFMAIL" type="YORN">{1}</column>
        <column name="LOCTOSERVREQ" type="YORN">{1}</column>
        <column name="PERSONUID"
type="INTEGER">{:uniqueid}</column>
        <column name="HASLD" type="YORN">{0}</column>
        <column name="LANGCODE" type="UPPER">{en}</column>
    </table>
</datamap>
</user>
</ldapsync>

```

6. Click the diskette icon on the menu bar to save the changes.
7. From the **Select Action** drop-down list, choose **Reload Request**, and confirm by selecting **OK** for the instance reload request.

Because VMMSYNC is configured to run every 5 minutes, wait for 5 minutes and check whether the users and groups from the LDAP repository have been populated into the Maximo tables. Perform these steps to check the MAXUSER table:

1. Navigate to **Go To → Security → Users**.
2. Press Enter to see all of the users that have been defined in the LDAP server. Figure 4-34 on page 92 shows an example.

The screenshot shows the 'Cron Task Setup' interface with a navigation bar at the top containing links like 'Web Replay', 'Bulletins: (1)', 'Go To', 'Reports', 'Start Center', 'Profile', 'Sign Out', and 'Help'. Below the navigation bar is a search and action area with a 'Find:' field, a 'Select Action' dropdown, and several icons. The main content area has tabs for 'List', 'User', 'Groups', and 'Security Profile'. Under the 'List' tab, there are options for 'Advanced Search', 'Save Query', and 'Bookmarks'. The 'Users' section shows a table with columns: 'User', 'Person', 'First Name', 'Last Name', 'Display Name', 'Status', and 'Type'. The table lists various users, including 'ADMIN', 'ADMINISTRATOR', 'GSTEIN', 'IMPACTUSER', 'ITNMADMIN', 'ITNMUSER', 'JENIE', 'MAXADMIN', 'MAXREG', 'MXINTADM', 'NETCOOL', 'PAADMIN', 'PLUTO', and several PMIN and PMPR users. The 'Status' column for all listed users is 'ACTIVE', and the 'Type' column for most is 'PRIMARY' or 'TYPE 1'. A 'Download' link is visible in the top right of the table area. At the bottom left of the table, there is a checkbox labeled 'Select Records'.

User	Person	First Name	Last Name	Display Name	Status	Type
ADMIN	ADMIN		admin		ACTIVE	PRIMARY
ADMINISTRATOR	ADMINISTRATOR		administrator		ACTIVE	PRIMARY
GSTEIN	GSTEIN		gstein		ACTIVE	PRIMARY
IMPACTUSER	IMPACTUSER		impactuser		ACTIVE	PRIMARY
ITNMADMIN	ITNMADMIN		itnmadmin		ACTIVE	PRIMARY
ITNMUSER	ITNMUSER		itnmuser		ACTIVE	PRIMARY
JENIE	JENIE	Andre	Jenie	Andre Jenie	ACTIVE	PRIMARY
MAXADMIN	MAXADMIN		maxadmin		ACTIVE	TYPE 1
MAXREG	MAXREG		maxreg		ACTIVE	TYPE 1
MXINTADM	MXINTADM		mxintadm		ACTIVE	TYPE 1
NETCOOL	NETCOOL		netcool		ACTIVE	PRIMARY
PAADMIN	PAADMIN		paadmin		ACTIVE	PRIMARY
PLUTO	PLUTO		pluto		ACTIVE	PRIMARY
PMINCADMUSR	PMINCADMUSR		PMINCADMUSR		ACTIVE	TYPE 1
PMINCANALUSR	PMINCANALUSR		PMINCANALUSR		ACTIVE	TYPE 1
PMINCMGRUSR	PMINCMGRUSR		PMINCMGRUSR		ACTIVE	TYPE 1
PMINCOVNNUSR	PMINCOVNNUSR		PMINCOVNNUSR		ACTIVE	TYPE 1
PMPRBADMUSR	PMPRBADMUSR		PMPRBADMUSR		ACTIVE	TYPE 1
PMPRBANALUSR	PMPRBANALUSR		PMPRBANALUSR		ACTIVE	TYPE 1
PMPRBMGRUSR	PMPRBMGRUSR		PMPRBMGRUSR		ACTIVE	TYPE 1

Figure 4-34 Example of a list of users in the MAXUSER table

Perform these steps to check the PERSON table:

1. Navigate to **Go To** → **Administrator** → **Resources** → **People**.
2. Press Enter to see all of the persons as defined in the LDAP server.

Perform these steps to check the MAXGROUP table:

1. Navigate to **Go To** → **Security** → **Security Groups**.
2. Press Enter to see all of the groups that have been defined in the LDAP server.

4.5 IBM Tivoli Application Dependency Discovery Manager security setup

In this section, we discuss the IBM Tivoli Application Dependency Discovery Manager security configuration. Figure 4-35 on page 93 shows the IBM Tivoli Application Dependency Discovery Manager security components.

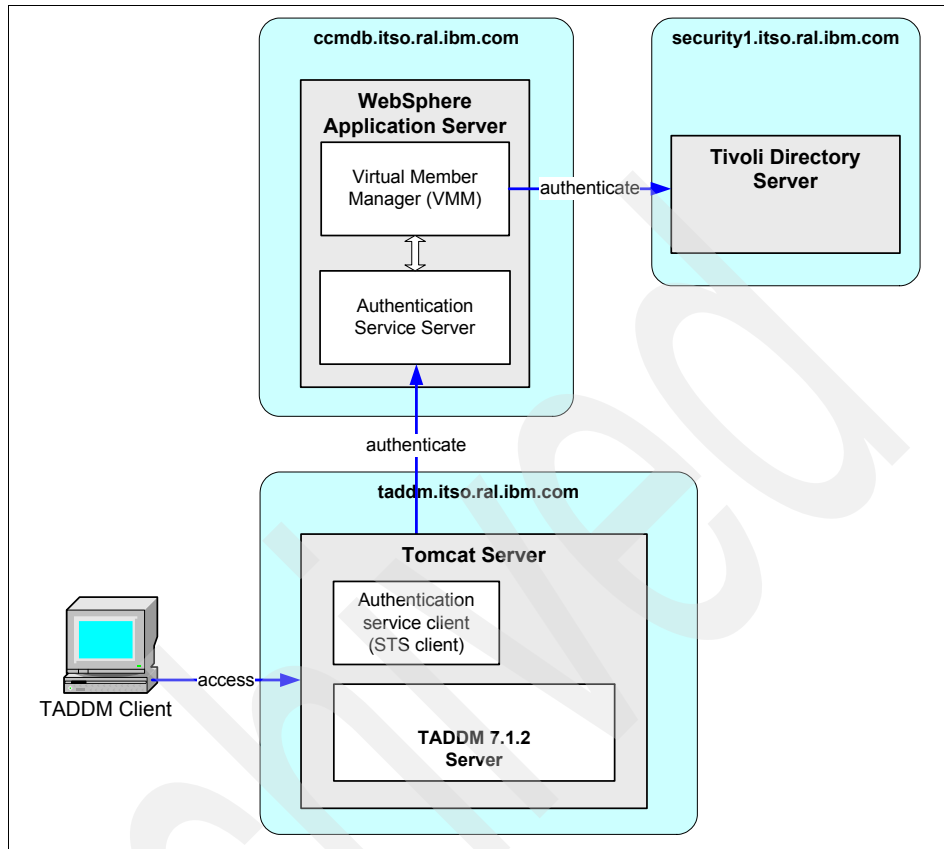


Figure 4-35 IBM Tivoli Application Dependency Discovery Manager security components

You perform most of the required configurations by configuring key-value pairs in the `collation.properties` file. It is located in the `$COLLATION_HOME/dist/etc` directory on the IBM Tivoli Application Dependency Discovery Manager server.

In the `collation.properties` file, you must set the user management module to `vmm` to define that IBM Tivoli Application Dependency Discovery Manager will use Virtual Member Manager to get access to the users and groups that are defined in LDAP:

```
com.collation.security.usermanagementmodule=vmm
```

In the Federated Repositories section of the `collation.properties` file, set the attributes as shown in Example 4-3 on page 94.

Example 4-3 Federated repositories settings

```
#=====
# Federated Repositories/ESS
# Authentication & SSO
#=====
# FQDN of the machine hosting WebSphere,
# Federated Repositories and ESS
com.collation.security.auth.websphereHost=ccmdb.itso.ral.ibm.com
# WebSphere system port (default = 2809)
com.collation.security.auth.webspherePort=9809
com.collation.security.auth.VMAdminUsername=wasadmin
com.collation.security.auth.VMAdminPassword=6JJNk5/aNG4SGoNc9Por9g==
com.collation.security.auth.VMMUserSearchBase=ou=users,ou=SWG,o=IBM,c=US
com.collation.security.auth.VMMGroupSearchBase=ou=groups,ou=SWG,o=IBM,c=US
com.collation.security.auth.ESSClientTrustStore=
com.collation.security.auth.ESSClientTrustPwd=
```

You must restart the IBM Tivoli Application Dependency Discovery Manager server for the changes to take effect. Restarting IBM Tivoli Application Dependency Discovery Manager also encrypts any password fields within the `collation.properties` file that were written in clear text.

The configuration is the communication between the authentication service client on the IBM Tivoli Application Dependency Discovery Manager server to the authentication service implementation on the WebSphere Application Server. On the IBM Tivoli Application Dependency Discovery Manager server, edit the `ibmessclientauthncfg.properties` file in the `$COLLATION_HOME/dist/etc` directory. Change the `authnServiceURL` parameter to point to the authentication server, which is the machine where the IBM Tivoli Change and Configuration Management Database is installed, as shown in Example 4-4.

Example 4-4 Authentication server in `ibmessclientauthncfg.properties` file

```
# This is the URL for the ESS Authentication Service
authnServiceURL=http://ccmdb.itso.ral.ibm.com:9080/TokenService/service
s/Trust
```

The authentication service client on IBM Tivoli Application Dependency Discovery Manager server uses this URL to call back to the Security Token Service on the WebSphere Application Server to authenticate an IBM Tivoli Application Dependency Discovery Manager user or to validate the LTPA token that IBM Tivoli Application Dependency Discovery Manager receives.

Configure the parameters in the `sas.client.props` file, which is located in the `$COLLATION_HOME/dist/etc` directory. You need to set the parameters as shown in Example 4-5 to validate your WebSphere session authentication.

Example 4-5 The `sas.client.props` file

```
com.ibm.CORBA.securityServerHost=ccmdb.itso.ral.ibm.com
com.ibm.CORBA.securityServerPort=9809
com.ibm.CORBA.loginTimeout=300
com.ibm.CORBA.loginSource=properties

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

4.6 IBM Tivoli Netcool products LDAP configuration

We discuss the following topics for the IBM Tivoli Netcool products:

- ▶ 4.6.1, “IBM Tivoli Netcool/OMNibus LDAP configuration” on page 95
- ▶ 4.6.2, “Configuring Tivoli Integrated Portal LDAP” on page 103
- ▶ 4.6.3, “IBM Tivoli Netcool/Impact LDAP Configuration” on page 114

4.6.1 IBM Tivoli Netcool/OMNibus LDAP configuration

You can configure IBM Tivoli Netcool/OMNibus to authenticate using LDAP by configuring the Process Agent and Object Server through the Pluggable Authentication Modules authentication on UNIX. Pluggable Authentication Modules (PAM) is a UNIX-provided authentication framework. The Process Agent manages the Object Server and other processes, automatically restarts the processes, and runs external procedures from the Object Server. The Process Agent by default uses system authentication, but you can configure it to authenticate using LDAP by configuring PAM authentication. The Object Server users authenticate to the IBM Tivoli Netcool/OMNibus Object Server Database by default. You can also configure the Object Server to authenticate using LDAP by configuring PAM authentication. On Windows, because PAM is not available, the Process Agent can only authenticate using system authentication and the Object Server can only authenticate to the Object Server database.

With PAM, we can configure the Object Server to authenticate using third-party PAM modules to multiple authentication sources. In this book, we look specifically at configuring Process Agent and Object Server PAM authentication

using the Red Hat Enterprise Linux 4-provided `pam_ldap.so` module to an IBM Tivoli Directory Server V6.1. The discussion includes these topics:

- ▶ “Configuring LDAP authentication with PAM” on page 96
- ▶ “Configuring Process Agent LDAP authentication” on page 97
- ▶ “Configuring Object Server with Process Agent LDAP username” on page 98
- ▶ “Configuring Object Server LDAP authentication” on page 99
- ▶ “Enabling PAM debugging” on page 102

Configuring LDAP authentication with PAM

The Red Hat Enterprise Linux 4-provided `pam_ldap.so` PAM module is configured through the `/etc/ldap.conf` and `/etc/ldap.secret` system files. For detailed information about configuring the system PAM LDAP module, see the operating system documentation. For this example, which shows configuration using the Red Hat Enterprise Linux 4-provided `pam_ldap.so` module to an IBM Tivoli Directory Server V6.1 without SSL, use the following steps to configure LDAP on the system:

1. Edit the `/etc/ldap.conf` file. Modify the host and base parameters:

host	A resolvable host name or IP address
base	The base distinguished name for the LDAP server

Example 4-6 shows the modified part of the `/etc/ldap.conf` file to address the LDAP server and the LDAP search base.

Example 4-6 Defining host and base parameters in /etc/ldap.conf file

```
# Your LDAP server. Must be resolvable without using LDAP.
# Multiple hosts may be specified, each separated by a
# space. How long nss_ldap takes to failover depends on
# whether your LDAP client library supports configurable
# network or connect timeouts (see bind_timelimit).
host security1.itso.ral.ibm.com

# The distinguished name of the search base.
base ou=SWG,o=IBM,c=US
```

2. If a user ID is not configured, `pam_ldap.so` will connect anonymously. If you need to change users' passwords in LDAP or if LDAP will not allow anonymous connections, configure a bind username and password. To configure an LDAP username, set `rootbinddn` in the `/etc/ldap.conf` file to the distinguished name that is used to bind to the LDAP server, as shown in the Example 4-7 on page 97.

Example 4-7 Adding rootbinddn in /etc/ldap.conf file

```
#The distinguished name to bind to the server with
# if the effective user ID is root. Password is
# stored in /etc/ldap.secret (mode 600)
#rootbinddn cn=manager,dc=example,dc=com
rootbinddn cn=root
```

3. If authenticating with a specific LDAP rootbinddn user, create the /etc/ldap.secret file with this user's password in plain text. It is not possible to encrypt the contents of the /etc/ldap.secret file, so permissions on the file must be read/write for root only. Example 4-8 shows how to store the password in the /etc/ldap.secret file and how to set permission only to the root user.

Example 4-8 Creating the /etc/ldap.secret file

```
echo password > /etc/ldap.secret
chown root /etc/ldap.secret
chmod 600 /etc/ldap.secret
```

Configuring Process Agent LDAP authentication

On UNIX, you can configure the Process Agent to authenticate via local UNIX authentication, PAM, Kerberos, and HP Trusted Computer Base. On Windows, the Process Agent can only use local Windows authentication. See the *IBM Tivoli Netcool/OMNIbus Administrator Guide* for more details about configuring PAM authentication:

http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?topic=/com.ibm.netcool_OMNIBus.doc/welcome.htm

Process Agent uses the PAM service netcool to authenticate. On Linux, PAM configuration files are in the /etc/pam.d directory and named for each service. The file is composed of space-separated tokens' type, control module-path, and optional module-arguments. On other UNIX operating systems, the PAM configuration file is the /etc/pam.conf file and is composed of space-separated tokens' service, type, control module-path, and optional module-arguments. In this example, we configure the Process Agent with PAM authentication using the pam_ldap.so PAM module on Linux:

1. If the /etc/pam.d/netcool file does not exist, copy the /etc/pam.d/passwd file to this file.
2. Edit the /etc/pam.d/netcool file. Three authentication types are required by the Process Agent: account, auth, and password. Issue the **man pam.conf** command for more information about the contents of this file. Edit the module-path to pam_ldap.so. The PAM module is located in the

/lib/security directory. Example 4-9 shows the three authentication types in the /etc/pam.d/netcool file.

Example 4-9 Contents of the /etc/pam.d/netcool file

```
#%PAM-1.0
auth      required      pam_ldap.so
account   required      pam_ldap.so
password  required      pam_ldap.so
```

3. If the Process Agent is running with the `-authenticate PAM` option, it queries the updated PAM configuration file on the next authentication attempt. There is no need to restart the Process Agent. If the Process Agent is not running with `-authenticate PAM`, it must be restarted with this option.
4. Verify authentication as an LDAP user with the `nco_pa_status` command, as shown in Example 4-10.

Example 4-10 Result of nco_pa_status command

```
[netcool@tbsm bin]# ./nco_pa_status -user paadmin
Login Password :
-----
Service Name Process Name      Hostname                User  Status  PID
-----
Core          MasterObjectServer tbsm.itso.ral.ibm.com netcoolRUNNING 26300
-----
```

Configuring Object Server with Process Agent LDAP username

Now that the Process Agent is configured to authenticate through LDAP using PAM authentication, you must reconfigure Object Server with an LDAP username and password for the Process Agent. Object Server connects and authenticates to the Process Agent when running external procedures. Object Server properties `PA.Username` and `PA.Password` are used to connect to the Process Agent. Follow these steps:

1. Edit the `$NCHOME/etc/<ObjectServer_Name>.props` file by making these changes:

PA.Name	Process Agent name
PA.Username	LDAP username
PA.Password	LDAP user's password encrypted with the <code>nco_pa_crypt</code> command (unless operating with Federal Information Processing Standard (FIPS) 140-2 mode)

Example 4-11 on page 99 shows sample entries.

Example 4-11 Sample changes to ObjectServer properties

```
PA.Name: 'NCO_PA'  
PA.Password: 'DNFCBIBCF0GBGGG'  
PA.Username: 'paadmin'
```

2. Restart Object Server using Process Control, as shown in Example 4-12.

Example 4-12 Restarting Object Server

```
[netcool@tbsm bin]$ ./nco_pa_stop -service Core -user paadmin  
Login Password:  
[netcool@tbsm bin]$ ./nco_pa_start -service Core -user paadmin  
Login Password:  
[netcool@tbsm bin]$ ./nco_pa_status -user paadmin  
Login Password:
```

Service Name	Process Name	Hostname	User	Status	PID
Core	MasterObjectServer	tbsm	netcool	RUNNING	27498

Configuring Object Server LDAP authentication

On UNIX, by default, the Object Server database is used for authentication. You can configure specific Object Server users to use PAM for authentication. In the following example, we configure Object Server PAM LDAP authentication and create a new user to use PAM authentication.

Object Server uses the PAM service `nco_objserv` to authenticate. On Linux, PAM configuration files are in the `/etc/pam.d` directory and are named for each service. The file is composed of space-separated tokens' type, control module-path, and optional module-arguments. On other UNIX operating systems, the PAM configuration file is the `/etc/pam.conf` file and is composed of space-separated tokens' service, type, control module-path, and optional module-arguments. In this example, we configure the Object Server with PAM authentication using the `pam_ldap.so` PAM module on Linux:

1. Copy the `/etc/pam.d/netcool` file that was created in "Configuring Process Agent LDAP authentication" on page 97 to the `/etc/pam.d/nco_objserv` file. Three authentication types are required by the Object Server: account, auth, and password. Example 4-13 on page 100 illustrates the file contents.

Example 4-13 The /etc/pam.d/nco_objserv file

```
#%PAM-1.0
auth      required    pam_ldap.so
account   required    pam_ldap.so
password  required    pam_ldap.so
```

2. Open the administrator console with the **nco_config** command.
3. Select the Object Server and log in as a system user, such as root.
4. Go to **User** → **Users**.
5. Right-click a user, and select **Add User**.
6. Enter the Username, Full Name, and select group assignment. Figure 4-36 on page 101 illustrates creating a user from the IBM Tivoli Netcool/OMNibus Administrator.

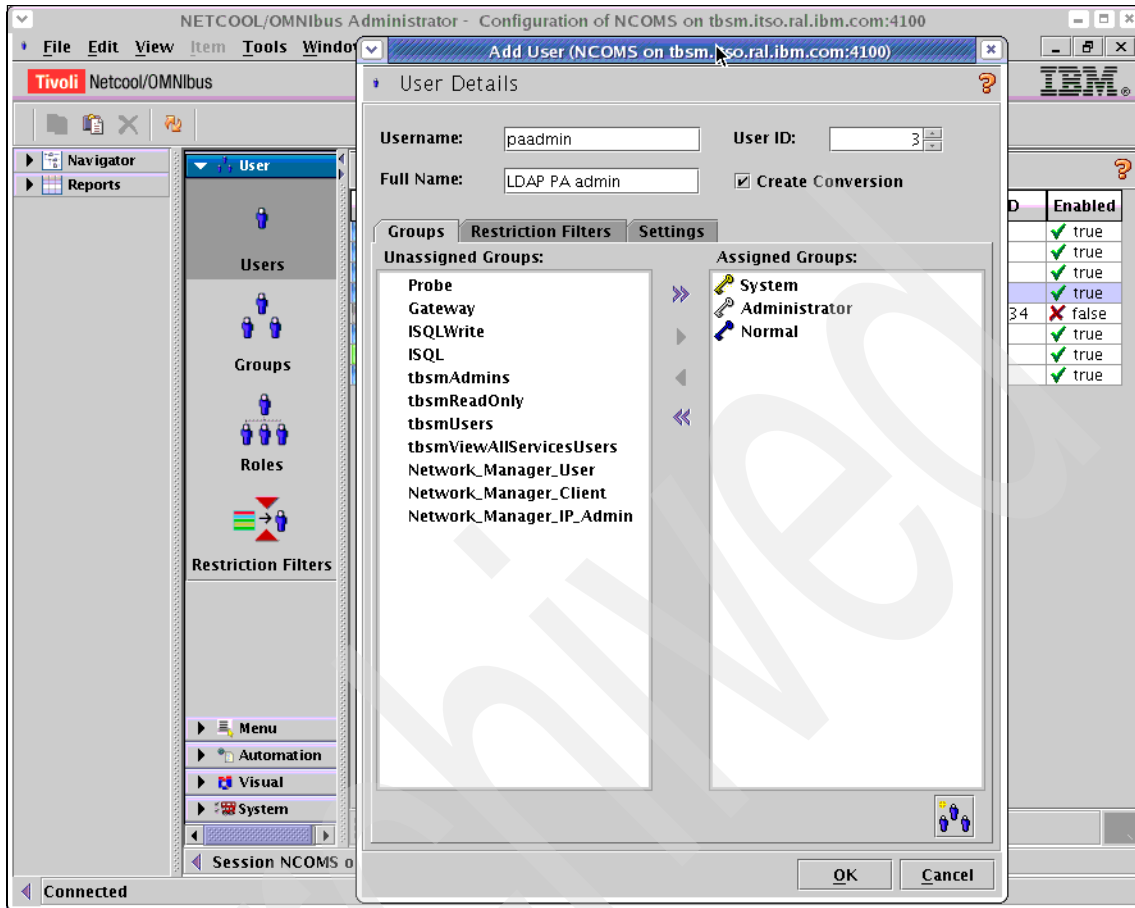


Figure 4-36 Create User

7. Select the **Settings** tab. Select the **Use PAM** check box, which will set the user to authenticate through PAM instead of the Object Server database. Figure 4-37 on page 102 illustrates configuring the user to use PAM for authentication. When selected, the password cannot be set from the Administrator.

Add User (NCOMS on tbsm.itso.ral.ibm.com:4100)

User Details

Username: User ID:

Full Name: ☒ Create Conversion

Groups Restriction Filters Settings

Passwords

Password: ☒ Use PAM

Verify:

User Type: Super User ☒ User Enabled

OK Cancel

Figure 4-37 Setting PAM authentication for the user

8. Verify that the LDAP user can log in using the Administrator, Event list, or the `nco_sql` command.

Enabling PAM debugging

If Process Agent or Object Server PAM LDAP authentication is unsuccessful, you can enable PAM debug to collect additional system information. The exact configuration for enabling PAM debug can vary by operating system. The following example is for configuring PAM debug on Red Hat Enterprise Linux 4:

1. In the `/etc/pam.d/netcool` file (for the Process Agent) or the `/etc/pam.d/nco_objserv` (for the Object Server) PAM configuration file, add debug to the end of each line that is used by Object Server authentication. Example 4-14 on page 103 shows the authentication lines for `nco_objserv` with debug enabled.

Example 4-14 Debugging directive

```
#%PAM-1.0
auth      required      pam_ldap.so debug
account   required      pam_ldap.so debug
password  required      pam_ldap.so debug
```

2. Make sure that syslog is configured to log debug statements to a file. In the `/etc/syslog.conf` file, there must be a line for debug followed by a filename. If this file is modified, restart syslogd:

```
*.debug                                /var/adm/ncolog
```
3. Create an empty `/etc/pam_debug` file using the command **touch** `/etc/pam_debug`.
4. Restart the Process Agent or Object Server with messagelevel debug.
5. Attempt authenticating as the LDAP user.
6. After the authentication fails, check the `$NCHOME/omnibus/log/<Object Server name>.log` Object Server log file, the `$NCHOME/omnibus/log/<process agent name>.log` Process Agent log file, and the `/var/adm/ncolog` syslog debug file for error messages.

4.6.2 Configuring Tivoli Integrated Portal LDAP

IBM Tivoli Business Service Manager and IBM Tivoli Network Manager for IP have the same console interface, Tivoli Integrated Portal. By default, the administration console is accessed through Tivoli Integrated Portal at `https://<hostname>:16316/ibm/console`. The default Tivoli Integrated Portal administrative user is `tipadmin`.

Note: For users to modify events in the Active Event List, the users must exist and have roles in IBM Tivoli Netcool/OMNIBus Object Server. Therefore, if the users need to acknowledge or modify events, you must configure Tivoli Integrated Portal to authenticate to IBM Tivoli Netcool/OMNIBus and you must configure IBM Tivoli Netcool/OMNIBus to authenticate to LDAP.

Before configuring LDAP, it is important to analyze the current Tivoli Integrated Portal, IBM Tivoli Business Service Manager, IBM Tivoli Network Manager for IP users, and the current LDAP users. As configured in the installation, the users authenticate through Object Server. If the same user is configured in IBM Tivoli Netcool/OMNIBus and in LDAP, the user will be unable to authenticate to the console. It is important that each user is uniquely defined in either IBM Tivoli Netcool/OMNIBus or LDAP. For example, by default, the `tipadmin` user is created in the IBM Tivoli Netcool/OMNIBus users database to access the embedded

WebSphere-based Administrative Console. If the tipadmin user exists in LDAP as well and no other unique administrative users are defined, a new unique administrative user must be created with the same roles prior to configuring LDAP so that at least one administrative user can log in and configure the remaining users.

This requirement is due to the definition of a federated repository that is used in WebSphere authentication. A federated repository is a single realm composed of several authentication sources. All authentication sources are combined into the single realm. For this reason, if the same user ID exists in multiple authentication sources, the user ID will not be added to the realm and an error will be thrown. Because all authentication sources are combined, if any one authentication source is unavailable, no user will be able to authenticate. For a description of federated repositories, see the article, “IBM WebSphere Developer Technical Journal: Expand your user registry options with a federated repository in WebSphere Application Server V6.1,” at this Web site:

http://www.ibm.com/developerworks/websphere/techjournal/0701_ilechko/0701_ilechko.html

We discuss the following topics in this section:

- ▶ “Creating the WebSphere administrative user” on page 104
- ▶ “Configuring LDAP on Tivoli Integrated Portal” on page 107
- ▶ “Setting the LDAP user roles” on page 113
- ▶ “Reconfiguring any duplicate users” on page 113

Creating the WebSphere administrative user

Due to the limitation of federated repositories requiring unique user IDs, if tipadmin exists in the LDAP repository and the IBM Tivoli Netcool/OMNIBus users database, the user will be unable to authenticate after LDAP is configured. If the tipadmin user is not configured in LDAP or another administrative user exists in the IBM Tivoli Netcool/OMNIBus users’ database but not in LDAP, creating a new administrative user is unnecessary.

To create a new embedded WebSphere administrative user in Tivoli Integrated Portal, perform the following steps:

1. Log in as tipadmin or an administrative user to the Tivoli Integrated Portal. The default URL is `https://hostname:16316/ibm/console`. Figure 4-38 on page 105 illustrates the Tivoli Integrated Portal login page.

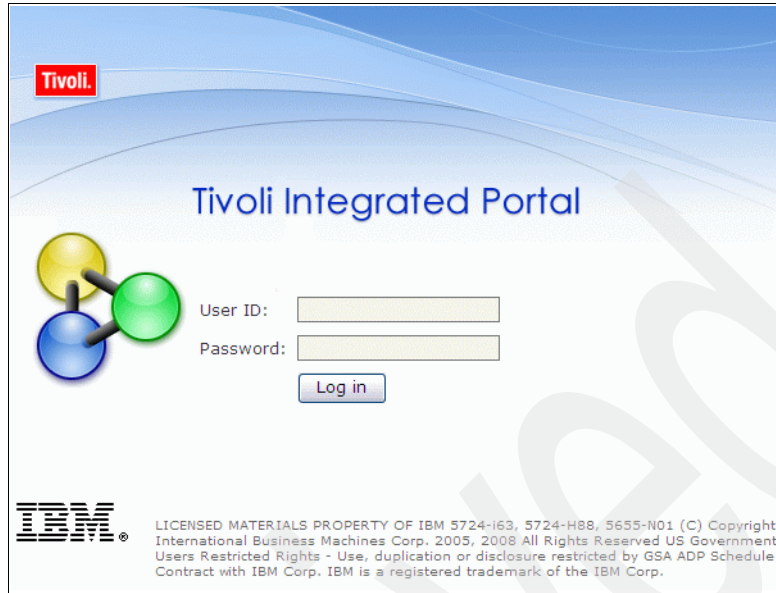


Figure 4-38 Tivoli Integrated Portal login

2. Select **Users and Groups** → **Manage Users** on the left navigation panel.
3. Select **Create** on the Manage Users window.
4. Enter a unique user id, first name, last name, password, and confirm password. Figure 4-39 on page 106 illustrates the options to create a user.

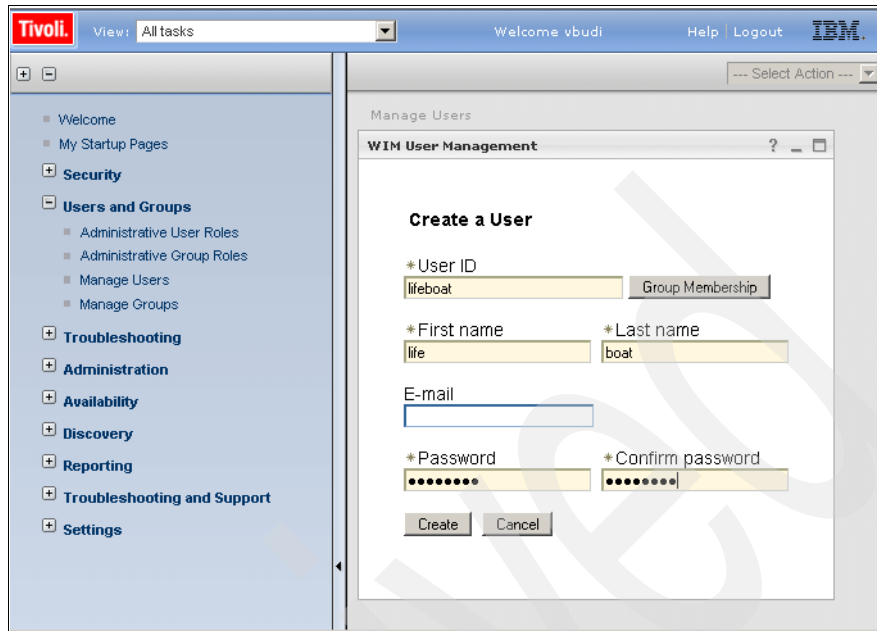


Figure 4-39 Creating a new user

5. Click **Create**.
6. The message “The user was created successfully” displays. Select **Close**.
7. Select **Administrative User Roles** from the left navigation panel.
8. Select **Add** under Administrative User Roles.
9. Enter the User name, which was just created, and select administrative roles. Select **Apply**.
10. You will be prompted whether to **Save** or **Review**. Select **Save**.
11. The user is now created and will be displayed under Administrative User Roles.

Figure 4-40 on page 107 illustrates the newly created user and the assigned roles.

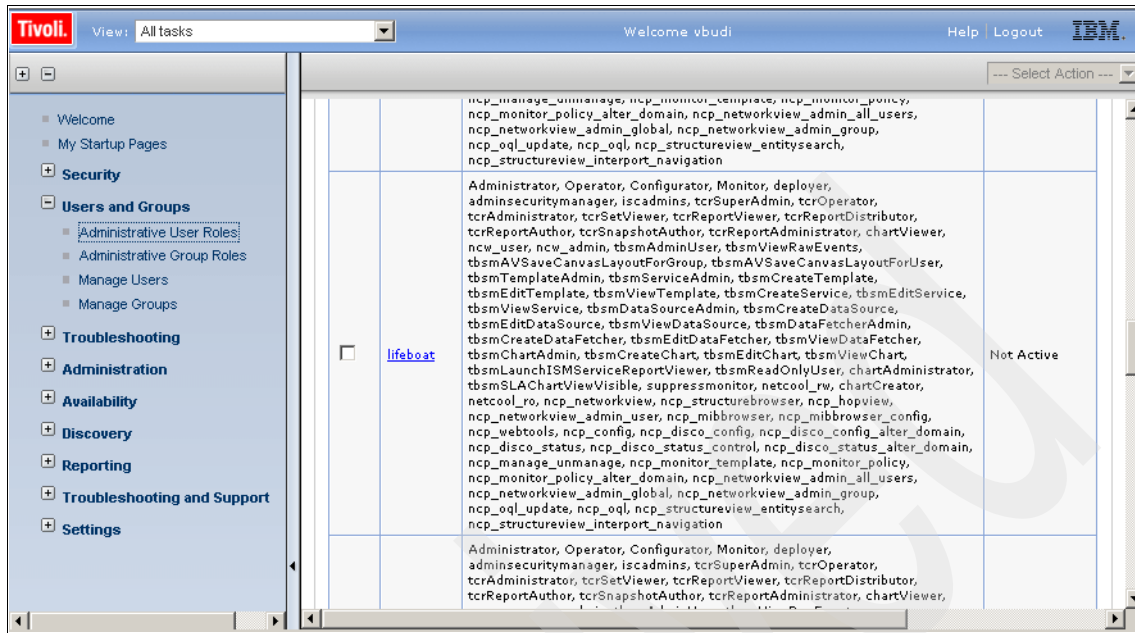


Figure 4-40 Adding administrative user roles

Configuring LDAP on Tivoli Integrated Portal

To configure authentication, perform the following steps:

1. Log in as tipadmin or an administrative user to the WebSphere Administration Console. The default URL is <https://hostname:16316/ibm/console>.
2. Select **Security** from the left navigation panel.
3. Select **Secure administration, applications, and infrastructure** from the left navigation panel. Figure 4-41 on page 108 illustrates the security setting options.

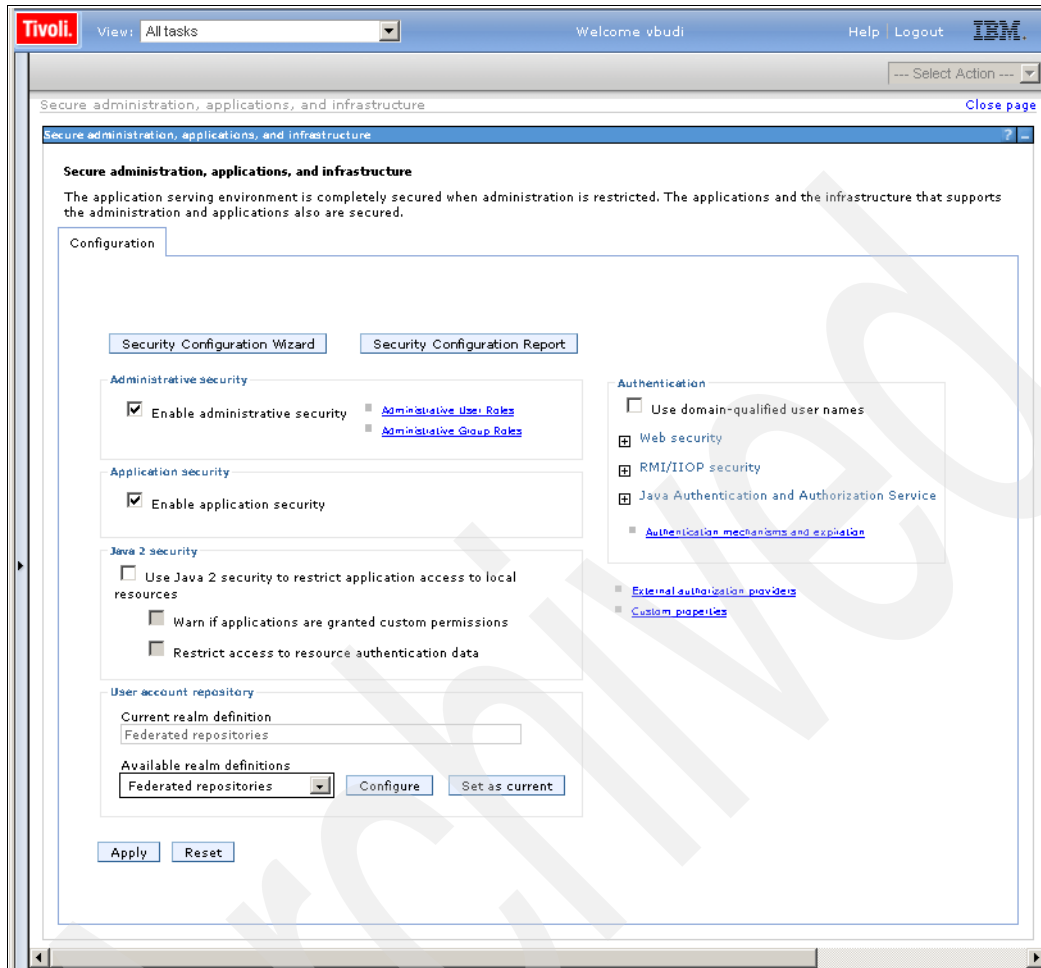


Figure 4-41 Security options from the Administration Console

4. Under User account repository, ensure that the Available realm definitions value is set to **Federated repositories**, and select **Configure**.
5. Under Related Items, select **Manage repositories**.

Figure 4-42 on page 109 illustrates the Manage repositories selection under Related Items.

General Properties

* Realm name
TIPRealm

* Primary administrative user name
tipadmin

Server user identity

☒ Automatically generated server identity

☐ Server identity that is stored in the repository

Server user ID or administrative user on a Version 6.0.x node

Password

☒ Ignore case for authorization

Repositories in the realm:

Select	Base entry	Repository identifier	Repository type
☐	o=defaultWIMFileBasedRealm	InternalFileRepository	File
☐	o=netcoolObjectServerRepository	NetcoolObjectServer	Custom

Additional Properties

- [Property extension repository](#)
- [Entry mapping repository](#)
- [Supported entity types](#)

Related Items

- [Manage repositories](#)

Figure 4-42 Manage repositories

- If the default IBM Tivoli Netcool/OMNIBus users database authentication is used, only InternalFileRepository and NetcoolObjectServer exist in the Managed repositories list. Select **Add** to add an LDAP repository.
- Under General Properties, enter a unique repository identifier to identify this repository. Select the directory type from the drop-down list. Enter the primary host name and the port of the LDAP server. Enter the bind distinguished name and bind password with the distinguished name and the password of the user that will connect to LDAP. Figure 4-43 on page 110 illustrates the LDAP repository settings.

Secure administration, applications, and infrastructure [Close page](#)

Secure administration, applications, and infrastructure > [Federated repositories](#) > **ITSOLDAP**

Specifies the configuration for secure access to a Lightweight Directory Access Protocol (LDAP) repository with optional failover servers.

Configuration

General Properties

* Repository identifier
ITSOLDAP

LDAP server

* Directory type
IBM Tivoli Directory Server Version 6

* Primary host name
security1 Port
389

Failover server used when primary is not available:

[Delete](#)

Select	Failover host name	Port
	None	

[Add](#)

Support referrals to other LDAP servers
ignore

Security

Bind distinguished name
cn=root

Bind password

Login properties
uid

Certificate mapping
EXACT_DN

Certificate filter

☐ Require SSL communications

☒ Centrally managed
[Manage endpoint security configurations](#)

☒ Use specific SSL alias
NodeDefaultSSLSettings [SSL configurations](#)

Additional Properties

- [Performance](#)
- [LDAP entity types](#)
- [Group attribute definition](#)

[Apply](#) [OK](#) [Reset](#) [Cancel](#)

Figure 4-43 LDAP repository settings

8. Select **Apply** and **Save**.
9. Select **Secure administration, applications, and infrastructure** from the left navigation panel. Under User account repository, ensure that Available realm definitions is set to **Federated repositories**, and select **Configure**.
10. Under Repositories in the realm, select **Add Base entry to Realm**.

11. Enter the distinguished name of the base entry that uniquely identifies this set of entries in the realm. Also, enter the distinguished name of a base entry in this repository:
 - a. Distinguished name of a base entry that uniquely identifies this set of entries in the realm: Label to uniquely identify the authentication source distinguished name in Tivoli Integrated Portal. This label can be any label as long as it is unique in the federated repository.
 - b. Distinguished name of a base entry in the repository: Root of the subtree in LDAP, which will become part of the federated repository. All users and groups to be defined in Tivoli Integrated Portal must be located within this subtree.

Figure 4-44 illustrates the configured distinguished name of the base entry in the repository.

Secure administration, applications, and infrastructure Close page

Secure administration, applications, and infrastructure > Federated repositories > ou=SWG,o=IBM,c=US

Specifies a set of identity entries in a repository that are referenced by a base entry into the directory information tree. If multiple repositories are included in the same realm, it might be necessary to define an additional distinguished name that uniquely identifies this set of entries within the realm.

Configuration

General Properties

* Repository
ITSOLDAP Add Repository...

* Distinguished name of a base entry that uniquely identifies this set of entries in the realm
ou=SWG,o=IBM,c=US

Distinguished name of a base entry in this repository
ou=SWG,o=IBM,c=US

Apply OK Reset Cancel

Figure 4-44 Distinguished name of base entry

12. Select **Apply** and **Save**.
13. Restart Tivoli Integrated Portal as the embedded WebSphere administrative user created in “Creating the WebSphere administrative user” on page 115 or with a unique administrative user. Example 4-15 on page 112 demonstrates restarting Tivoli Integrated Portal on UNIX.

Example 4-15 Restarting Tivoli Integrated Portal

```
[netcool@tnmip bin]$ ./stopServer.sh server1 -username lifeboat
-password itso4you
ADMU0116I: Tool information is being logged in file
/opt/IBM/tivoli/tip/profiles/TIPProfile/logs/server1/stopServer.log
ADMU0128I: Starting tool with the TIPProfile profile
ADMU3100I: Reading configuration for server: server1
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server server1 stop completed.

[netcool@tnmip bin]$ ./startServer.sh server1
ADMU0116I: Tool information is being logged in file
/opt/IBM/tivoli/tip/profiles/TIPProfile/logs/server1/startServer.log
ADMU0128I: Starting tool with the TIPProfile profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 10031
```

14. Verify that you can access Tivoli Integrated Portal as an LDAP user. If you are unable to log in as any administrative user, perform these operations to temporarily disable security:
 - a. Shut down Tivoli Integrated Portal or kill the Java process if you do not have an administrative user that can access Tivoli Integrated Portal.
 - b. Edit the
\$TIPHOME/profiles/TIPProfile/config/cells/TIPCell/security.xml
file.
 - c. Change the security tag enabled="true" (shown in Example 4-16) to enabled="false".

Example 4-16 Disable security in the security.xml file

```
<security:Security xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:orb.securityprotocol="http://www.ibm.com/websphere/appserver/sche
mas/5.0/orb.securityprotocol.xmi"
xmlns:security="http://www.ibm.com/websphere/appserver/schemas/5.0/secu
rity.xmi" xmi:id="Security_1" useLocalSecurityServer="true"
useDomainQualifiedUserNames="false" enabled="true" cacheTimeout="600"
issuePermissionWarning="false" activeProtocol="BOTH"
enforceJava2Security="false" enforceFineGrainedJCASecurity="false"
appEnabled="true" dynamicallyUpdateSSLConfig="true"
activeAuthMechanism="LTPA_1" activeUserRegistry="WIMUserRegistry_1"
defaultSSLSettings="SSLConfig_TIPNode_1">
```

- d. Start Tivoli Integrated Portal, as shown in Example 4-15.

- e. Open the Tivoli Integrated Portal default URL:
https://hostname:16316/ibm/console and click **login** without prompting for a user.
- f. Add administrative roles to a user or remove the user's repository base entry if it is wrong.
- g. Enable Tivoli Integrated Portal global security again by clicking **Security** → **Secure administration, applications, and infrastructure** or by changing the security tag of the security.xml file.
- h. Shut down and restart Tivoli Integrated Portal, as shown in the Example 4-15 on page 112.

Setting the LDAP user roles

Now that IBM Tivoli Business Service Manager and IBM Tivoli Network Manager for IP are configured to authenticate through both LDAP and the IBM Tivoli Netcool/OMNIbus users database, Tivoli Integrated Portal roles can be added for an LDAP authenticated user. Before the LDAP user can log in to the Tivoli Integrated Portal Server, roles must be assigned to the user. Refer to “Creating the WebSphere administrative user” on page 104.

Reconfiguring any duplicate users

Duplicate users in the source repositories cannot be authenticated. To repair any users that are defined in multiple repositories, one of the users must be deleted, and the user roles must be reconfigured. To view and remove any users that are configured in multiple source repositories, perform the following steps:

1. Log in as tipadmin or an administrative user to the Tivoli Integrated Portal. The default URL is https://hostname:16316/ibm/console.
2. Select **Users and Groups** → **Manage Users**.
3. Search for all users, and identify the users listed twice. The Unique Name listed will identify the source repository for the user. For example, uid=lifboat,o=netcoolObjectServerRepository indicates the lifboat user is in the IBM Tivoli Netcool/OMNIbus users database. And, cn=itnmuser,ou=users,ou=SWG,o=IBM,c=US indicates the itnmadmin user is in LDAP. Figure 4-45 shows duplicate itnmadmin users: one in the LDAP repository and one in the IBM Tivoli Netcool/OMNIbus users database.

☐	itnmadmin	itnmadmin	itnmadmin	cn=itnmadmin,ou=users,ou=SWG,o=IBM,c=US
☐	itnmadmin	itnmadmin	tivoli	uid=itnmadmin,o=netcoolObjectServerRepository
☐	itnmclient	itnmclient	tivoli	uid=itnmclient,o=netcoolObjectServerRepository
☐	itnmuser	itnmuser	itnmuser	cn=itnmuser,ou=users,ou=SWG,o=IBM,c=US
☐	lifboat	lifboat	lifboat	uid=lifboat,o=netcoolObjectServerRepository

Figure 4-45 Duplicate users

4. Select a user to remove. You can remove either the LDAP user or the IBM Tivoli Netcool/OMNIBus users database user from the embedded WebSphere administration interface.
5. Select **Delete**. Confirm to **Delete** the user.
6. Roles are assigned per user ID. If the roles do not exist for this user, assign new roles to this user. If the roles do exist for the user, modify the current roles so that they take effect.

4.6.3 IBM Tivoli Netcool/Impact LDAP Configuration

IBM Tivoli Netcool/Impact users can authenticate using LDAP for centralized user management. By default, all IBM Tivoli Netcool/Impact users authenticate using the local file-based repository in WebSphere Federated Repositories. LDAP authentication can be configured with Secure Sockets Layer (SSL) or non-SSL authentication. The following configuration example uses non-SSL authentication. To deploy with SSL authentication, you must import the LDAP signer certificate into the embedded WebSphere trust store prior to configuring LDAP. For more information, see the *IBM Tivoli Netcool/Impact 5.1 Administrator Guide*, SC23-8829.

IBM Tivoli Netcool/Impact has three console interfaces:

- ▶ The embedded WebSphere Administration console resides at `http://<hostname>:9060/ibm/console`. The default WebSphere Administration console user is `wasadmin` with the password `netcool`.
- ▶ GUI Server Console resides at `http://<hostname>:9080/nci`. The default GUI Server Console user is `admin` with the password `netcool`.
- ▶ Operator view resides at `http://<hostname>:9080/opview`.

As described in 4.6.2, “Configuring Tivoli Integrated Portal LDAP” on page 103, check that no users are defined in both the LDAP and in the WebSphere file-based user repository. Users duplicated in both repositories will not be able to log in.

Next, we discuss the following topics:

- ▶ “Creating the WebSphere administrative user” on page 115
- ▶ “Configuring LDAP for IBM Tivoli Netcool/Impact” on page 116
- ▶ “Setting the LDAP user roles” on page 117
- ▶ “Reconfiguring any duplicate users” on page 119

Creating the WebSphere administrative user

Because Federated Repositories requires unique user IDs, if wasadmin exists in the LDAP repository and in the WebSphere file-based user repository, the user will be unable to authenticate after LDAP is configured. If the wasadmin user is not configured in LDAP or another administrative user exists in the WebSphere file-based user repository but not in LDAP, creating a new embedded WebSphere Administrative user is not necessary.

To create a new embedded WebSphere administrative user, perform the following steps:

1. Log in as wasadmin or an administrative user to the WebSphere Administration Console. The default URL is `http://hostname:9060/ibm/console`.
2. Select **Users and Groups** → **Manage Users** on the left navigation panel.
3. Select **Create** on the Manage Users window.
4. Enter a unique user ID, first name, last name, password, and confirm password. Figure 4-46 shows the Create a User window options.

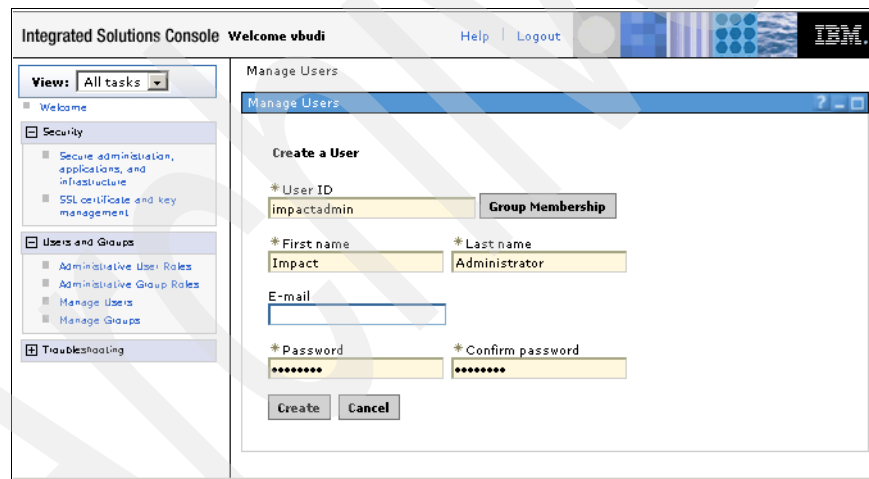
The screenshot shows the 'Integrated Solutions Console' interface. On the left is a navigation tree with 'Users and Groups' expanded, showing 'Manage Users' as the selected option. The main content area is titled 'Manage Users' and contains a 'Create a User' form. The form has the following fields: 'User ID' (containing 'impactadmin'), 'First name' (containing 'Impact'), 'Last name' (containing 'Administrator'), 'E-mail' (empty), 'Password' (masked with asterisks), and 'Confirm password' (masked with asterisks). There are 'Create' and 'Cancel' buttons at the bottom of the form. The top of the console shows a welcome message for 'vbudi' and links for 'Help' and 'Logout'.

Figure 4-46 Creating a new user

5. Select **Create**.
6. The message “The user was created successfully” will be displayed. Select **Close**.
7. Select **Users and Groups** → **Administrative User Roles** from the left navigation panel.
8. Select **Add** under Administrative User Roles.

9. Enter the User name, which was just created, and select all roles. Select **Apply**.
10. You will be prompted whether to Save or Review. Select **Save**.
11. The user is now created and will be displayed under Administrative User Roles. Figure 4-47 illustrates the resulting new user that was created and the roles assigned to that user.

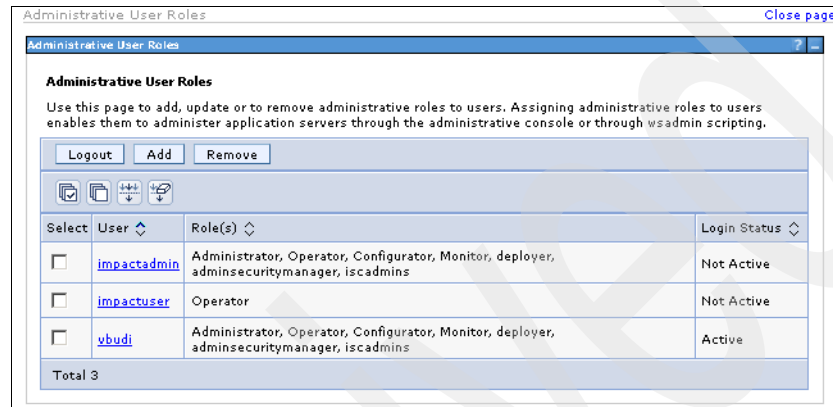


Figure 4-47 Administrative user roles

Configuring LDAP for IBM Tivoli Netcool/Impact

To configure IBM Tivoli Netcool/Impact authentication, perform similar steps as “Configuring LDAP on Tivoli Integrated Portal” on page 107 from the WebSphere Administration Console. The default URL is `http://hostname:9060/ibm/console`.

After you configure the security and federated repositories settings in WebSphere, restart IBM Tivoli Netcool/Impact as the embedded WebSphere administrative user created in “Creating the WebSphere administrative user” on page 115. Example 4-17 demonstrates restarting IBM Tivoli Netcool/Impact on Linux.

Example 4-17 Restarting IBM Tivoli Netcool/Impact

```
[netcool@tbsm bin]$ ./ewas.sh stop -username impactadmin -password
itso4you
ADMU0116I: Tool information is being logged in file
/opt/ibm/netcool/ewAS/profiles/ImpactProfile/logs/server1/stopServer.log
ADMU0128I: Starting tool with the ImpactProfile profile
ADMU3100I: Reading configuration for server: server1
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server server1 stop completed.
```

```
[netcool@tbsm bin]$ ./ewas.sh start -username impactadmin -password  
itso4you  
ADMU0116I: Tool information is being logged in file  
/opt/ibm/netcool/eWAS/profiles/ImpactProfile/logs/server1/startServer.l  
og  
ADMU0128I: Starting tool with the ImpactProfile profile  
ADMU3100I: Reading configuration for server: server1  
ADMU3200I: Server launched. Waiting for initialization status.  
ADMU3000I: Server server1 open for e-business; process id is 10171
```

Setting the LDAP user roles

Now that IBM Tivoli Netcool/Impact is configured to authenticate through both LDAP and in the WebSphere file-based user repository, the GUI Server administrative roles can be added for an LDAP authenticated user. Before the LDAP user can log in to the GUI Server, roles must be assigned to the user. Use the following steps to define an LDAP user as a GUI Server administration user and to configure the roles for the user:

1. Edit the `$NCHOME/etc/tivoli_vmm4ncos/guiserver.settings` file.
2. In the **ROLE SETTINGS** section of the file, add these roles `role.IMPACT_USER.user`, `role.NETCOOL_ADMIN.user`, and `role.OPVIEW_USER.user` to an LDAP-authenticated user. Alternately, an LDAP authenticated group can be configured. Either groups or users must be configured, but not a combination of both. Save the file.

Example 4-18 illustrates adding an LDAP-authenticated user `netcool` as a GUI Server administration user in addition to the default `admin` user.

Example 4-18 guiserver.settings

```
role.IMPACT_USER.user=admin  
role.IMPACT_USER.user=netcool  
  
role.NETCOOL_ADMIN.user=admin  
role.NETCOOL_ADMIN.user=netcool  
  
role.OPVIEW_USER.user=admin  
role.OPVIEW_USER.user=netcool
```

3. From the `$NCHOME/etc/tivoli_vmm4ncos/bin` directory, run the **update-impact-roles.sh** script for UNIX or **update-impact-roles.bat** for Windows. This command creates the necessary roles for the user in IBM Tivoli Netcool/Impact. When prompted for a username and password, enter the embedded WebSphere administrative user that was created in “Creating the WebSphere administrative user” on page 115 or a unique administrative

username. Figure 4-19 illustrates running the **update-impact-roles.sh** script on Linux.

Example 4-19 Running the update-impact-roles.sh script

```
[netcool@tbsm bin]$ ./update-impact-roles.sh
09:56:50 Configuring roles...
Realm/Cell Name: <default>
Username: impactadmin
Password:
WASX7209I: Connected to process "server1" on node ImpactNode using SOAP
connector; The type of process is: UnManagedProcess
WASX7303I: The following options are passed to the scripting
environment and are available as arguments that are stored in the argv
variable: "[guiserver]"
09:58:19 Successfully configured roles.
```

4. Log in to the GUI Server with the LDAP username and password to verify authentication. The default URL is <http://hostname:9080/nci>. Figure 4-48 shows the GUI Server login page.

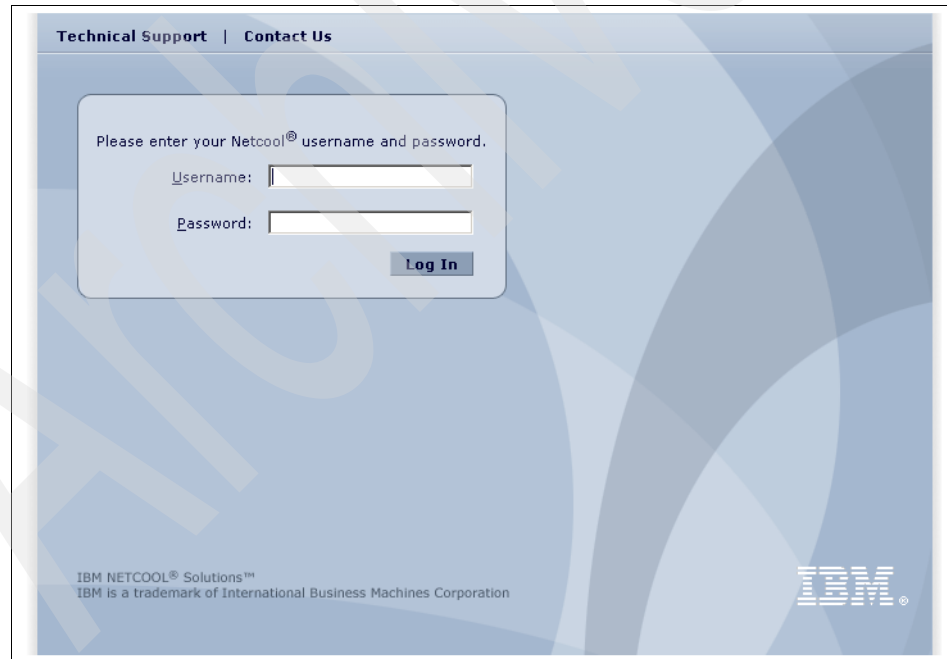


Figure 4-48 GUI Server login page

Reconfiguring any duplicate users

Duplicate users in the source repositories cannot be authenticated. To repair any users that are defined in multiple repositories, one of the users must be deleted, and the user roles must be reconfigured. To view and remove any users that are configured in multiple source repositories, perform the following steps:

1. Log in as an administrative user to the WebSphere Administration Console. The default URL is `http://hostname:9060/ibm/console`.
2. Select **Users and Groups** → **Manage Users**. You can also check the group authentication duplication in **Users and Groups** → **Manage Groups**.
3. Search for all users, and identify the users that are listed twice. The unique name listed will identify the source repository for the user. For example, `uid=wasadmin,o=defaultWIMFileBasedRealm` indicates the wasadmin user in the WebSphere file-based user repository.

`cn=wasadmin,ou=users,ou=SWG,o=IBM,c=US` indicates the wasadmin user in LDAP. Figure 4-49 shows duplicate wasadmin users: one user in the LDAP repository and one user in WebSphere file-based user repository.

Manage Users

Search for Users

Search by: User ID, * Search for: *, * Maximum results: 100

Search

20 users matched the search criteria.

Select	User ID	First name	Last name	E-mail	Unique Name
☐	tipadmin	tipadmin	tipadmin		cn=tipadmin,ou=users,ou=SWG,o=IBM,c=US
☐	twinstl	twinstl	twinstl		cn=twinstl,ou=users,ou=SWG,o=IBM,c=US
☐	vbudi	vbudi	vbudi		cn=vbudi,ou=users,ou=SWG,o=IBM,c=US
☐	wasadmin	wasadmin	wasadmin		cn=wasadmin,ou=users,ou=SWG,o=IBM,c=US
☐	wasadmin	wasadmin	wasadmin		uid=wasadmin,o=defaultWIMFileBasedRealm

Page 2 of 2, 2, Go, Total: 20

Figure 4-49 Duplicate users

4. Select a user to remove. You can remove either the LDAP user or the local IBM Tivoli Netcool/ImpactNetcool Database user from the embedded WebSphere Administration Console.
5. Select **Delete**. Confirm to **Delete** the user.

The default user wasadmin is automatically assigned the necessary roles, so the wasadmin user is now able to log in to the WebSphere Administration Console. Any non-default users need to have roles reassigned before they log in. To configure roles for an embedded WebSphere Administration Console user, see “Creating the WebSphere administrative user” on page 115. To configure roles for the GUI Server, see “Setting the LDAP user roles” on page 117.

4.7 IBM Tivoli Monitoring

IBM Tivoli Monitoring supports external authentication of the Hub Tivoli Enterprise Monitoring Server or Tivoli Enterprise Portal Server users with Lightweight Directory Access Protocol (LDAP) to shared registries. This support permits you to share user authentication information among IBM Tivoli Monitoring and other products.

There is also support for a single sign-on (SSO) capability between IBM Tivoli Monitoring and other Tivoli applications. This support allows you to launch other Web-based Tivoli applications from the Tivoli Enterprise Portal and to launch Tivoli Enterprise Portal from other applications without reentering the login credentials.

The single sign-on solution requires you to configure LDAP authentication from Tivoli Enterprise Portal Server. For our environment, we enable the single sign-on feature using LDAP authentication from Tivoli Enterprise Portal Server.

The single sign-on solution requires Lightweight Third Party Authentication (LTPA) tokens or keys to be exported and imported between the applications. Participating SSO applications pass LTPA tokens using browser cookies. These tokens are encrypted and signed so that they cannot be decoded without the correct keys.

4.7.1 Configuring Tivoli Enterprise Portal Server to authenticate to an LDAP repository

Table 4-8 on page 121 shows the LDAP parameters that are used for our environment.

Note: When you configure Tivoli Enterprise Portal Server to authenticate to an external LDAP repository, the base distinguished name (DN) defaults to o=ITMSS0Entry. If you want to use a federated user registry with another base distinguished name, you must specify an LDAP type of Other when you follow the procedures that are described in the section “Configuring the portal server to authenticate to an external LDAP repository” in the *IBM Tivoli Monitoring Administrator’s Guide*, SC32-9408. Then, use the TEPS/e administration console to configure an external LDAP server as described in the *IBM Tivoli Monitoring Administrator’s Guide* and specify the base distinguished name that you plan to use.

If you do not specify an LDAP type of “Other” when you change the base distinguished name, any subsequent reconfigurations of the portal server might result in unexpected LDAP configuration changes.

Table 4-8 LDAP parameters

Parameter	Value	Comment
LDAP Type	Other	This value is the LDAP Type for IBM Tivoli Directory Server 6.1.
LDAP base	ou=SWG,o=IBM,c=US	This value is the LDAP base node for finding users.
LDAP bind ID	cn=root	This value is the user ID used to search users in LDAP.
LDAP port number	389	This value is the port used by the LDAP server.
LDAP host name	security1	This value is the LDAP server host name.
Realm name	itsorealm	This value is the parameter shared across applications for SSO. The applications need to use the same realm name for SSO to work.
Domain name	itso.ral.ibm.com	This value is the network domain to be used by applications for SSO. Applications must reside in the same domain for SSO to work.
LDAP User ID (for testing)	itmadmin	This value is the user created in Tivoli Enterprise Portal Server and in the LDAP server to test LDAP and SSO.

Parameter	Value	Comment
LTPA Token file	/tnmip-tip-ltpa	This value is the key file used by SSO for authentication between the applications.

Before configuring Tivoli Enterprise Portal Server for federated repositories and LDAP authentication, note the following considerations:

- ▶ Tabulate all required LDAP parameters similar to Table 4-8 on page 121. The wrong parameters can cause the embedded WebSphere Application Server to become inaccessible. You can recover from this situation as discussed in 4.7.2, “Work-around for security problem” on page 126.
- ▶ The IBM Tivoli Monitoring sysadmin administrative account user must not be added to LDAP to be able to handle any unpredictable access issues due to LDAP. It can be used to create administrative accounts that are known only to LDAP.
- ▶ User IDs that need to make SOAP server requests (including user IDs that issue command-line interface (CLI) commands that invoke SOAP server methods) can be authenticated only through the hub Tivoli Enterprise Monitoring Server.
- ▶ The wasadmin user cannot be renamed or removed. IBM Tivoli Monitoring needs it to correctly synchronize Tivoli Enterprise Portal Server with federated repositories in the embedded WebSphere Application Server.
- ▶ User IDs cannot be a duplicate across separate repositories. Federated repositories for Tivoli Enterprise Portal Server with LDAP has the following members:
 - The default file repository for authenticating wasadmin
 - Tivoli Enterprise Monitoring Server authentication for sysadmin user
 - The LDAP repository for all other users

This function requires that wasadmin and sysadmin are not in LDAP.

Note: Integrated Solution Console in the embedded WebSphere Application Server is not enabled at startup to conserve resources. It has to be enabled by using the command `<itm_home>/<arch>/iw/scripts/enableISCLite.sh [true/false]` or from Manage Tivoli Enterprise Monitoring Services.

Follow these detailed steps to configure LDAP and SSO through Tivoli Enterprise Portal Server, assuming that the Hub Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server are already started:

1. Use an LDAP client to create IBM Tivoli Monitoring user IDs in the LDAP server. For this test, we use the JXplorer LDAP client to create the `itmadmin` user. The user IDs must have the UID field filled with the user ID, as well as other fields chosen to be used for LDAP authentication, such as the Common Name (CN), Surname (SN), and User Password. Make sure that user `sysadmin` is not created in the LDAP server, because you might need to log in to Tivoli Enterprise Portal Server if your LDAP configuration fails.
2. Stop all IBM Tivoli Monitoring agents that are installed in the Tivoli Enterprise Portal Server machine by using the command **`$ITMHOME/bin/itmcmd agent stop all`**.
3. Use this command to create a tar backup of `<itm install dir>/<interp>/iw` just in case you have any problem with the configuration in the embedded WebSphere Application Server:

```
tar -cvf /backup/iw_622_orig.tar $ITM_HOME/<interp>/iw
```
4. Use the **`./itmcmd agent start cq`** command to start Tivoli Enterprise Portal Server from the command line or use Manage Tivoli Enterprise Monitoring Services.
5. The first time that you want to enable the embedded WebSphere Application Server console, you must set `wasadmin`'s password in the file repository. You can use the Manage Tivoli Enterprise Monitoring Services or the command **`updateTEPSEPass.sh wasadmin <password>`**.
6. Enable the embedded WebSphere Application Server console; you must use the `wasadmin` user. Run **`$ITM_HOME/<interp>/iw/scripts/enableISCLite.sh true`**. Example 4-20 shows the execution.

Example 4-20 Enabling the embedded WebSphere Application Server console

```
[root@itm ~]# cd /opt/IBM/ITM/li6263/iw/scripts
[root@itm scripts]# ./enableISCLite.sh true
WASX7209I: Connected to process "ITMServer" on node ITMNode using
SOAP connector; The type of process is: UnManagedProcess
WASX7303I: The following options are passed to the scripting
environment and are available as arguments that are stored in the
argv variable: "[true]"
ISCLite started
```

7. Access the embedded WebSphere Application Server console and log in with user `wasadmin` and the password that you choose. The console resides in `http://<TEPS hostname>:15205/ibm/console`. Verify that you have access to WebSphere.

8. Reconfigure Tivoli Enterprise Portal Server for LDAP and SSO using the agent configuration dialog. Use the LDAP settings from Table 4-8 on page 121. In Linux, we issue **itmcmd config -A cq**. Example 4-21 shows the LDAP portion of the configuration.

Note: We found that the LDAP configuration from Tivoli Enterprise Portal Server is less error prone than entering the federated repositories configuration directly from the TEPS/e Administration.

Example 4-21 Configuring LDAP

```
[root@itm bin]# ./itmcmd config -A cq
Agent configuration started...

. . .

... DB2 instance connection parameters
. . .
... TEPS DB creation
. . .
... Warehouse DB connection parameters
. . .

LDAP Security: Validate User with LDAP?(1=Yes, 2=No)(Default is: 1):
LDAP type: [AD2000, AD2003, IDS6, OTHER](Default is: IDS6):
LDAP base(Default is: ou=SWG,o=IBM,c=US):
LDAP bind ID(Default is: cn=root):
LDAP bind password(Default is: *****):
Re-type: LDAP bind password(Default is: *****):
LDAP Port number(Default is: 389):
LDAP host name(Default is: security1):
Enable Single Sign On ? (1=Yes, 2=No)(Default is: 1):
Realm name(Default is: itsorealm):
Domain name(Default is: itso.ral.ibm.com):
. . .
Agent configuration completed...
```

9. Restart Tivoli Enterprise Portal Server and enable embedded WebSphere Application Server administration again. If you have an error, see 4.7.2, “Work-around for security problem” on page 126.
10. In embedded WebSphere Application Server, go to **Users and Groups** → **Manage Users**, and click **Search**. Verify that you can see your LDAP users. Otherwise, review the LDAP settings for the ITM_TEPS_LDAP repository.

11. The default base entry mapping for Tivoli Enterprise Portal Server is `o=ITMSSOEntry`. We decided early on to use a common mapping, so we use `ou=SWG,o=IBM,c=US` as the mapping to preserve the original suffix. We modify this mapping by clicking **Security** → **Secure administration, applications, and infrastructure** and clicking the Base Entry `o=ITMSSOEntry`. Change this entry to the mapping that is shown in Figure 4-50.

Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure

Secure administration, applications, and infrastructure > Federated repositories > `ou=SWG,o=IBM,c=US`

Specifies a set of identity entries in a repository that are referenced by a base entry into the directory information tree. If multiple repositories are included in the same realm, it might be necessary to define an additional distinguished name that uniquely identifies this set of entries within the realm.

Configuration

General Properties

* Repository
ITM_TEPS_LDAP Add Repository...

* Distinguished name of a base entry that uniquely identifies this set of entries in the realm
ou=SWG,o=IBM,c=US

Distinguished name of a base entry in this repository
ou=SWG,o=IBM,c=US

Apply OK Reset Cancel

Figure 4-50 Base entry mapping

12. Save the configuration changes in WebSphere Application Server.
13. From command line or embedded WebSphere Application Server, import the keys that are used to encrypt the LPTA tokens, including the key filename and a password to encrypt its key. The process is shown in Example 4-22.

Example 4-22 Importing the LTPA key

```
[root@itm scripts]# cd /opt/IBM/ITM/li6263/iw/scripts
[root@itm scripts]# ./importKeys.sh /tnmip-tip-ltpa itso4you
WASX7209I: Connected to process "ITMServer" on node ITMNode using
SOAP connector; The type of process is: UnManagedProcess
WASX7303I: The following options are passed to the scripting
environment and are available as arguments that are stored in the
argv variable: "[/tnmip-tip-ltpa, itso4you]"
```

14. Restart Tivoli Enterprise Portal Server.
15. Log in to Tivoli Enterprise Portal Server as sysadmin, click the **Users** icon, define a new user that already exists in LDAP server, clear the Distinguished

Name field, and click **Find**. A list of LDAP users appears. Select the correct DN defined in the LDAP Server, and click **OK**. See Figure 4-51.

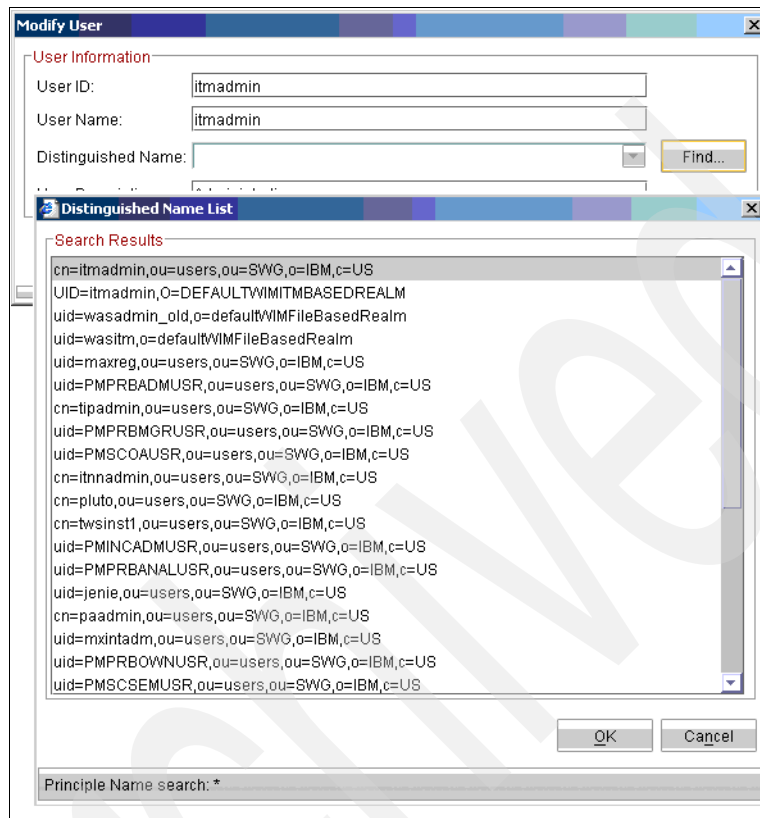


Figure 4-51 Defining the TEPS user from LDAP

16. Log out of Tivoli Enterprise Portal Server, and try to log in again by using the user ID that you previously defined in Tivoli Enterprise Portal Server.

4.7.2 Work-around for security problem

Use this work-around when access to the embedded WebSphere Application Server fails:

1. Stop Tivoli Enterprise Portal Server.
2. Edit the security.xml file from the `<itm_install_dir>/<interp>/iw/profiles/ITMProfile/config/cells/ITMCe11` directory and change the `enabled="true"` parameter to `enabled="false"`.
3. Restart Tivoli Enterprise Portal Server.

4. Enable the embedded WebSphere Application Server console with the **enableISCLite.sh** script.
5. Access the embedded WebSphere Application Server console with wasadmin user, go to **Security** → **Secure administration, applications, and infrastructure**, and click **Configure**. Check and verify the base entry mapping and repository settings for ITM_TEPS_LDAP. Do not remove or modify the default entries for DEFAULTWIMITMBASEDREALM and defaultWIMFileBasedRealm. Those entries are required for IBM Tivoli Monitoring and embedded WebSphere Application Server to work.
6. Re-enable security, either from the security.xml file or from the administration console.
7. Restart Tivoli Enterprise Portal Server.
8. If the problem persists, see the
/opt/IBM/ITM/<interp>/iw/profiles/ITMProfile/logs/ITMServer/SystemOut.log file to understand what is happening with embedded WebSphere Application Server.

4.8 IBM Tivoli Storage Productivity Center

The security setup of IBM Tivoli Storage Productivity Center data server must be performed to match the security settings of IBM Tivoli Storage Productivity Center with Tivoli Integrated Portal and to allow the Single Sign-On function with other Tivoli products.

Because we have already specified to use LDAP authentication for IBM Tivoli Storage Productivity Center server in the installation wizard, IBM Tivoli Storage Productivity Center is already using the LDAP server in security1 server. We must add the realm information and import/export the single sign-on key to allow single sign-on.

This configuration can be performed from Tivoli Integrated Portal. The step-by-step instruction is similar to 4.6.2, “Configuring Tivoli Integrated Portal LDAP” on page 103.

Archived

Data integration

In this chapter, we describe *data integration*, which is a way to exchange information between Tivoli products. Data integration allows consistent objects to be managed from multiple products. It also allows context to be preserved and used for navigating the user interface for various tools.

We discuss the following data integration topics:

- ▶ 5.1, “Data integration overview” on page 130
- ▶ 5.2, “Resource data integration” on page 135
- ▶ 5.3, “Event data integration” on page 176
- ▶ 5.4, “Reports integration” on page 218
- ▶ 5.5, “Other data integration” on page 229

5.1 Data integration overview

We separate the discussion of data integration into the following areas:

- ▶ *Resource data integration* involves discovering and correlating resource information across various products by using the Common Data Model and IDML exchanges. See 5.2, “Resource data integration” on page 135.
- ▶ *Event data integration* involves event data transfer from a product to another product, including status synchronization between events that reside in separate platforms. See 5.3, “Event data integration” on page 176.
- ▶ *Report integration* introduces and explains the platform to provide unified reports about how Tivoli products perform. See 5.4, “Reports integration” on page 218.
- ▶ Other data integration.

In this section, we also discuss the concept of the Common Data Model:

- ▶ 5.1.1, “Common Data Model” on page 130
- ▶ 5.1.2, “IDML data” on page 132

5.1.1 Common Data Model

The *Common Data Model* is an information model that provides consistent definitions for managed resources, business systems and processes, and other data, and the relationships between those elements.

The Common Data Model is used to integrate the understanding of data and the exchange of data between management products that concern the resources and components of a client's business. The Common Data Model is entirely composed of data definitions. These definitions are characteristics that identify resources, their meanings, and any restrictions on their lengths or values.

There are several industry data models, each one with a specific focus and specific implementation details. None of those industry models covers all client scenarios and needs; therefore, it is difficult to select just one.

The Common Data Model aims to absorb the best of each industry model and to provide a best-of-model solution. The content of the Common Data Model is obtained by merging applicable industry information, data model standards, and the data models that are used by current products into a single, converged model.

The applications that use the Common Data Model are able to share definitions and terminology for resource instance data that is common among them,

enabling the construction of higher level applications that encompass the overall management environment and share information between those systems.

The Common Data Model differs from a schema. A *schema* is usually associated with a database. It includes both the organization of data into a logical model and the specification of how that data is stored in specific columns of specific tables (also known as the *physical model* of the database).

The Common Data Model represents a logical model, which is composed of definitions, that enables the consistent identification of resource instances, information about them, and relationships between them. The data model links business and infrastructure processes with the systems that provide them, the users that invoke them, the policies that control them, the resources that processes use, and much more. The Common Data Model classifies and organizes the most commonly managed characteristics of users, resources, and business infrastructure information and processes and presents them in a way that all applications can use.

The Common Data Model has the following characteristics:

- ▶ It does not define the physical schema, and it does not define how a management system operates.
- ▶ It defines the resources and characteristics of a management environment that the management system monitors, analyzes, and controls.
- ▶ It is also in use when management applications exchange information about resource instances and their relationships to other resources.
- ▶ It standardizes the characteristics, the concepts of classes, attributes, interfaces, naming rules, and naming policies, and the data types that are in use.
- ▶ It provides consistent definitions of items, best practices for content, and guidelines for mapping resource instance data to the Common Data Model.

To foster integration among products, use the Common Data Model as the basis of your data modeling and interactive design. Because the Common Data Model is an information model, products are able to maintain their existing database schemas and also utilize the Common Data Model. When integrating with other products (such as when loading information into the IBM Tivoli Change and Configuration Management Database (CCMDB)), you need to use Common Data Model definitions and terminology. Using Common Data Model definitions and terminology fosters a consistent, one-time integration function that is reusable across multiple solutions.

We use the Common Data Model as an information model for data integration. When managed resources and business components are modeled using

Common Data Model specifications, Tivoli management products can understand and more easily exchange data across the enterprise.

This capability allows multiple IBM Tivoli management products that run in a single enterprise to work together. Although each of these products still maintains its own separate data that is related to the set of resources that it manages, the data maintenance and administration efforts for these multiple formats can be minimized.

The Common Data Model provides a language specification to describe infrastructure resources, their attributes, and relationships. It is based on Unified Modelling Language (UML) and includes influences from various management products, client solutions, and industry standards, including:

- ▶ Distributed Management Task Force (DMTF)
- ▶ Common Information Model (CIM)
- ▶ Business Process Execution Language (BPEL)
- ▶ IT Infrastructure Library (ITIL) specification
- ▶ Lightweight Directory Access Protocol (LDAP) directory schema
- ▶ TeleManagement Forum (TMf)
- ▶ Storage Networking Industry Association (SNIA)

The Common Data Model is used to define the external representation of Configuration Item (CI) information, as well as representations of infrastructure throughout the enterprise. It is used to exchange data across multiple applications, such as IBM Tivoli Application Dependency Discovery Manager, IBM Tivoli Change and Configuration Management Database, IBM Tivoli Network Manager for IP, IBM Tivoli Business Service Manager, and other applications that support the Common Data Model. For more information about the Common Data Model, refer to *IBM Tivoli Common Data Model: Guide to Best Practices*, REDP-4389.

5.1.2 IDML data

The Identity Markup Language (IDML) is an Extensible Markup Language (XML) dialect that is used to describe resource instances and their relationships according to the Common Data Model. Processes that produce these IDML files are called *Discovery Library Adapters (DLA)*. DLAs are an easy-to-develop, lightweight solution that allows for rapid integration between management products, customer data, and other third-party data sources. These IDML files are created by DLAs on a periodic frequency (set by the client) and then sent to a common location (set by the client) for multiple management products to consume the same set of IDML files.

As a discovery technology, IBM Tivoli Application Dependency Discovery Manager is a powerful tool for discovering information about hardware and software assets, including relationships and dependencies. Additional information about these assets from files, databases, and other sources is needed. This information is reformatted and written as Identity Markup Language (IDML) files that IBM Tivoli Application Dependency Discovery Manager can import.

IBM Tivoli Application Dependency Discovery Manager provides a generic IDML reader that can load several books into the database at a time. This loader is also referred to as the *BulkLoader*. Figure 5-1 shows the process of importing the IDML books into IBM Tivoli Application Dependency Discovery Manager.

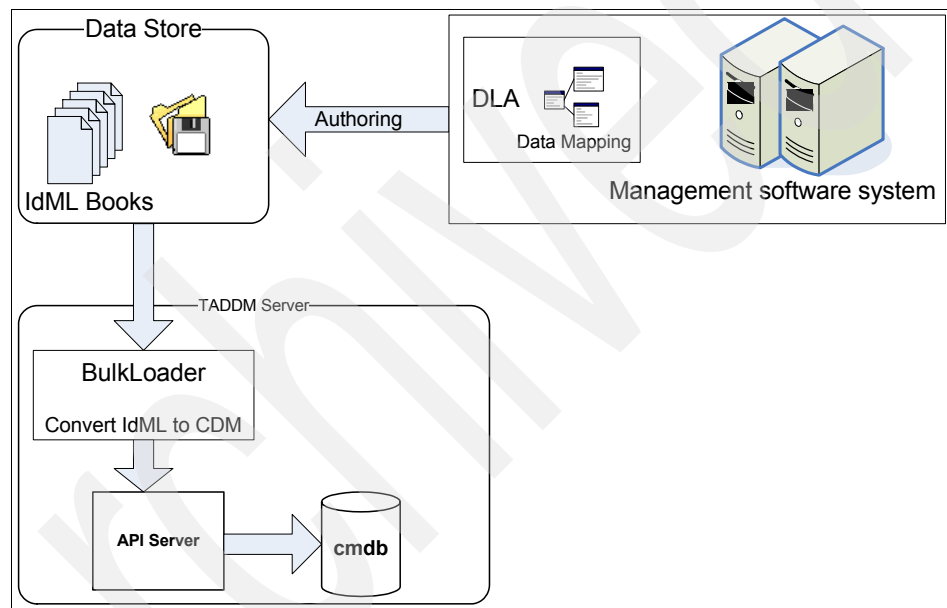


Figure 5-1 Importing IDML books inside IBM Tivoli Application Dependency Discovery Manager

As a result of execution, Discovery Library Adapters generate Identity Markup Language (IDML) files that contain the resource attributes and relationships that are known by a particular data source. These IDML files, otherwise known as *DLA books*, should be placed into a directory called the Discovery Library File Store (DLFS). These IDML-based files can be imported from the Discovery Library File Store into the discovered CI data space of the IBM Tivoli Change and Configuration Management Database. The bulk loader is run as a program that honors the reconciliation logic while batch importing the data. The bulk loader allows the data to arrive from separate data sources at various times to reconcile

together into a single representation, in order to guarantee the uniqueness of the Configuration Item resource data.

Some DLAs are written by IBM to extract information from management applications, while other DLAs are written by third-party providers. You can obtain a recent list of DLAs at this Web site:

<https://www.ibm.com/developerworks/wikis/display/tivoliaddm/Discovery+Library+Adapters>

Most of the DLA tools are recorded in the Tivoli Open Process Automation Library (OPAL) Web site:

<http://www.ibm.com/software/tivoli/opal>

The bulk load program is an efficient way to load large numbers of managed elements and relationship definitions into the IBM Tivoli Application Dependency Discovery Manager database. The bulk loader must be run by the user that starts and stops the IBM Tivoli Application Dependency Discovery Manager server. A sample command to run the bulk loader is shown in Example 5-1.

Example 5-1 Running bulk loader

```
su - cmdbadmin
export ${COLLATION_HOME}=/opt/IBM/cmdb/dist
cd $COLLATION_HOME/bin
./loadidml.sh -f <dlds> -h <hostname> -u <userid> -p <passwd>
```

The execution result of the bulk loader resides in these files:

- ▶ \$COLLATION_HOME/dist/bulk/results/xxxxx.results
- ▶ \$COLLATION_HOME/dist/log/bulkload.log

For additional details about the contents of the results file, you can turn on statistics data from the \$COLLATION_HOME/dist/etc/collation.properties file:

```
com.ibm.cdb.bulk.stats.enabled=true
```

Note: The working directory and the results directory must preexist, or the bulk loader does not run. It does not automatically create these directories.

The settings of the bulk loader are stored in the bulkload.properties file. You must ensure that the working directory and the results directory that are mentioned in the bulkload.properties file are valid.

For more information about IDML and its API, see this Web site:

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/DLADevGuide/c_cmdbsdk_dla_introducing.html

Several of the Common Data Model APIs can be invoked using the **api.sh** script. For example, you can query information in IBM Tivoli Application Dependency Discovery Manager by running the command:

```
./api.sh -u <taddm_admin> -p <password> find "select * from  
com.collation.platform.model.topology.process.ManagementSoftwareSystem  
where guid=='<GUID>'"
```

More information about the **api.sh** command can be found at this Web site:

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/SDKDevGuide/r_cmdbsdk_cliapi_commandsyntax.html

5.2 Resource data integration

Resource data integration allows the exchange of managed resource information between Tivoli products. This integration is typically performed by exporting and importing the IDML data format from one product to another product. IBM Tivoli Application Dependency Discovery Manager serves as the central contact point for this interaction. Data from IBM Tivoli Application Dependency Discovery Manager is then loaded to the IBM Tivoli Change and Configuration Management Database as actual configuration items (CIs).

The overall structure of the resource data integration scenario is depicted in Figure 5-2 on page 136.

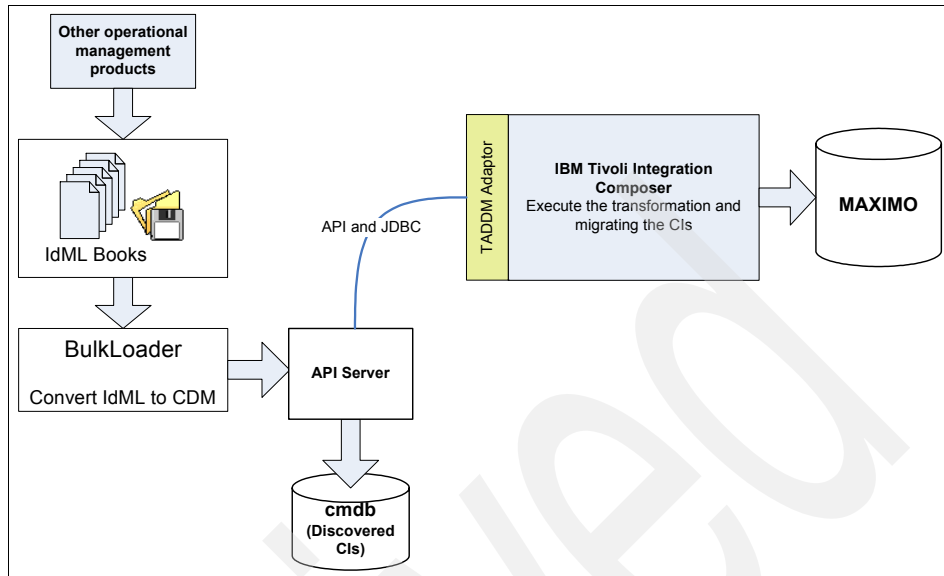


Figure 5-2 Resource data integration

We discuss resource data integration in the following sections:

- ▶ 5.2.1, “IBM Tivoli Integration Composer” on page 136
- ▶ 5.2.3, “IBM Tivoli Provisioning Manager integration” on page 142
- ▶ 5.2.4, “IBM Tivoli Monitoring integration” on page 150
- ▶ 5.2.5, “IBM Tivoli Business Service Manager integration” on page 155
- ▶ 5.2.6, “IBM Tivoli Network Manager for IP integration” on page 163

5.2.1 IBM Tivoli Integration Composer

IBM Tivoli Change and Configuration Management Database uses IBM Tivoli Integration Composer to read the discovered CIs from IBM Tivoli Application Dependency Discovery Manager. IBM Tivoli Integration Composer is a stand-alone integration application that migrates data from source to target. Its primary purpose in the IBM Service Management environment is to transfer discovered data into the Actual CI space of the environment.

IBM Tivoli Integration Composer processing is depicted in Figure 5-3 on page 137.



Figure 5-3 IBM Tivoli Integration Composer processing

Figure 5-3 demonstrates how IBM Tivoli Integration Composer extracts configuration information from IBM Tivoli Application Dependency Discovery Manager using API calls to populate IBM Tivoli Change and Configuration Management Database Actual CI information. We use this process in our data integration solution.

IBM Tivoli Integration Composer tools are highly dependent on the API call and database format to which it connects. Slight differences that are caused by varying product levels (a patch or fix pack) or differing sets of installed components can cause an existing mapping process to be unusable.

5.2.2 Promoting Actual CIs to Authorized CIs

After the CIs are loaded into IBM Tivoli Change and Configuration Management Database as Actual CIs, we need to make the Actual CIs into Authorized CIs. Only Authorized CIs can be manipulated with IBM Service Management tools. This transformation is performed in these sections:

- ▶ “Setting up a classification” on page 137
- ▶ “Define the CI classification to be used as a top level CI” on page 138
- ▶ “Map the new CI hierarchy to the Actual CI” on page 139
- ▶ “Promoting Actual CIs to CIs” on page 140

Setting up a classification

You must define the classes that you want to use for CI definition. In this section, we demonstrate the manual definition of a separate TOPCICLASS classification. This classification under TOPCICLASS will become the Authorized CIs. The Authorized CI object is called a CI, while the Actual CI is known as an ACTCI object. Follow these steps:

1. Select **Go To** → **Administration** → **Classification**, and create a new classification. Call it TOPCICLASS, and then, click the **Classifications** tab.
2. Click **New Row** in the Use With section. Click **CI** from the Use With Object field, as shown in Figure 5-4 on page 138, and click **Save**. *Do not set this to Top Level.*

The screenshot shows the IBM Classifications web interface. The top navigation bar includes links for Web Replay, Bulletins: (1), Go To, List, Reports, Start Center, Profile, Sign Out, and Help. The main content area is titled "Classifications" and contains a search bar, a "Select Action" dropdown, and a "List" tab. Below the tab, there are input fields for "Classification" (TOPCICLASS), "Classification Path" (TOPCICLASS), "Parent Classification", "Generate Description?" (checked), "Top level CI Classification", "Organization", "Site", "Owner Group", "Service Group", "Indicated Priority", and "Use Classification?" (checked). A "Use With" section shows a table with columns "Use With Object", "Description", and "Top Level?". The table has two rows: "ACTCI" with "Use with Actual Configuration Items" and "Top Level?" (unchecked), and "CI" with "Use with Configuration Items" and "Top Level?" (checked). A "Details" section shows "Use With Object" (CI) and "Use with Configuration Items". A "Children" section shows a table with columns "Classification", "Classification Desc", "Generate Description?", "Use Classification?", "Organization", and "Site". The table lists several classifications, including "APP.J2EE.JBOSS.JBOSSSERVER", "APP.SOFTWAREIMAGE", "APP.PACKAGEDAPP.MYSAP.BASIS.MYSAPJ2", "SYS.IVM", and "SYS.HPC.CM.CONFIGURATIONMANAGEMENT?".

Figure 5-4 Setting up TOPCICLASS classification

Define the CI classification to be used as a top level CI

We use the SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM as the base to create CI.LINUXUNITARYCOMPUTERSYSTEM, which becomes our new top level CI:

1. Select **Go To** → **Administration** → **Classification**, and filter by **COMPUTERSYSTEM**. Choose the hierarchy that you want to use for promotion. In our case, we choose **SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM**.
2. Go to the Select Action Menu and choose **Duplicate Classification**. Rename the new classification, such as **CI.LINUXUNITARYCOMPUTERSYSTEM**.
3. Change the parent to **TOPCICLASS** that we created in “Setting up a classification” on page 137. Remove the Use with ACTCI record, because this is a copy of the ACTCI object that we will use only with a CI object. Select the **Top Level** check box. The result is shown in Figure 5-5 on page 139.

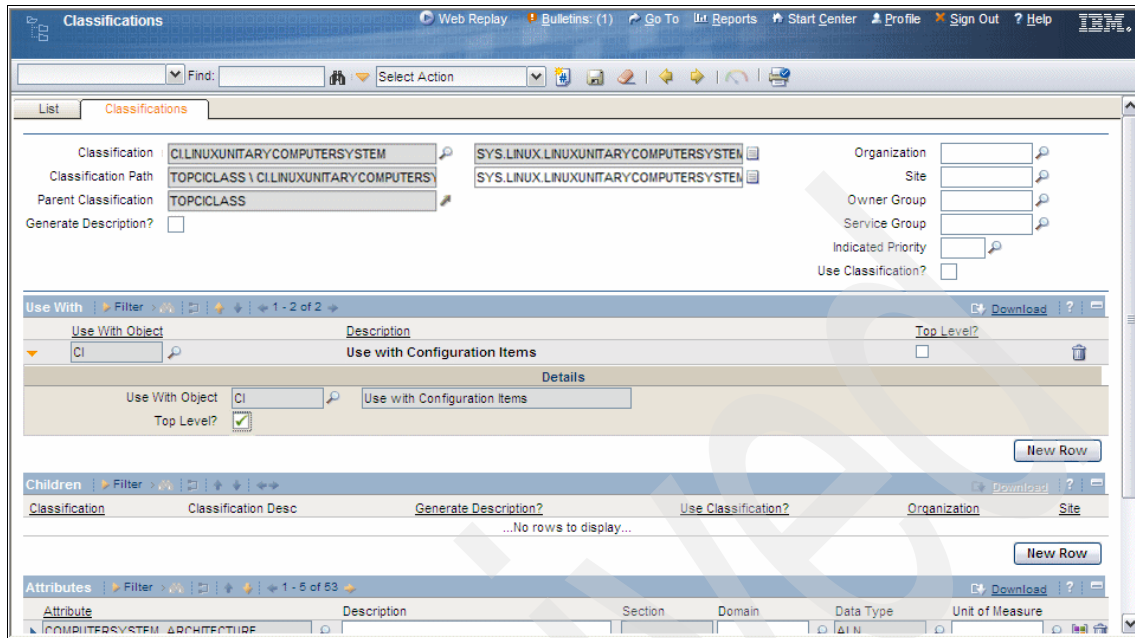


Figure 5-5 Setting up SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM classification

4. Save the **CI.LINUXUNITARYCOMPUTERSYSTEM** object. Now, you have two classifications: **CI.LINUXUNITARYCOMPUTERSYSTEM** for the CI and **SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM** for the Actual CI.

Map the new CI hierarchy to the Actual CI

Use the Manage CI Hierarchies dialog to map the new CI classification **CI.LINUXUNITARYCOMPUTERSYSTEM** to the Actual CI classification **SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM**:

1. Select the new classification **CI.LINUXUNITARYCOMPUTERSYSTEM**, and from the Select Action menu, click **Manage CI Hierarchies**.
2. In the Manage CI Hierarchies dialog, go to the Relationship section. See Figure 5-6 on page 140.

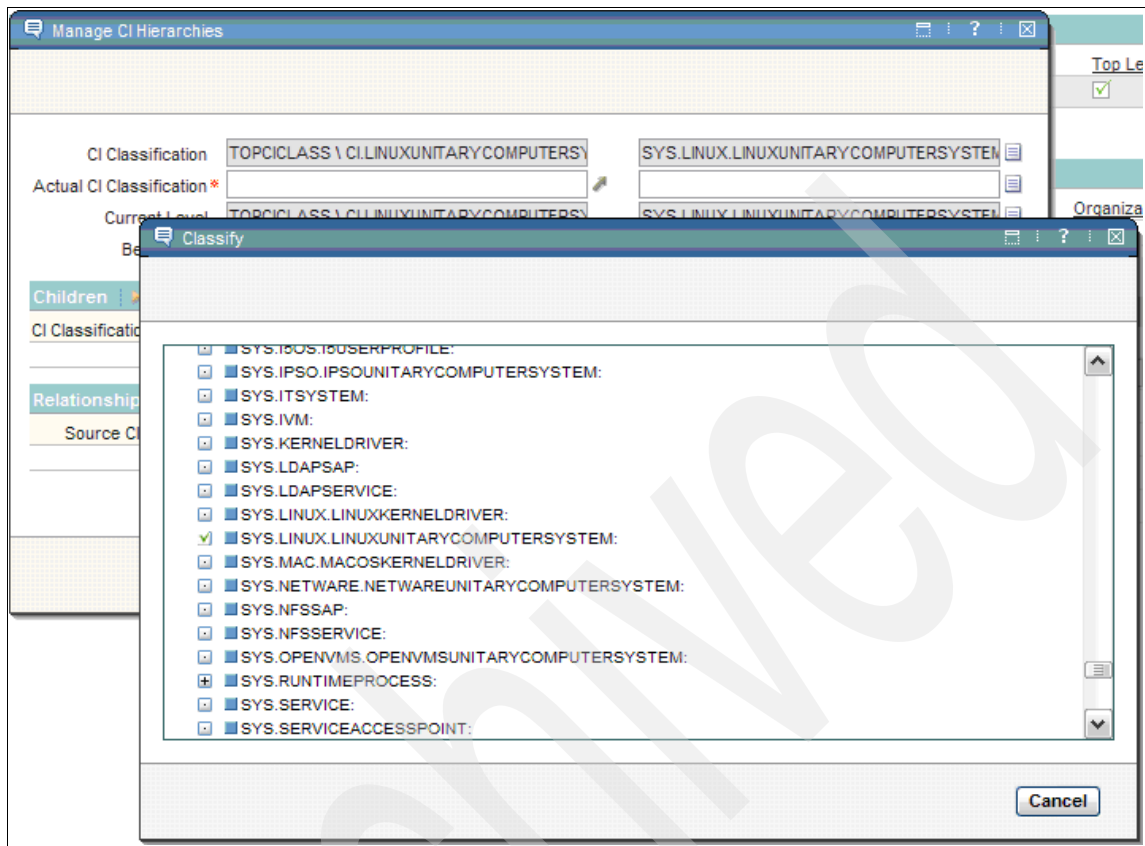


Figure 5-6 CI mapping

- For the Source CI, choose the actual CL classification **SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM**. Click **OK** to save the mapping.

Promoting Actual CIs to CIs

After we define the CI classes that can be used for the Authorized CI, we can promote the CIs and set their status as operational. Perform the following steps:

- Select **Go To** → **IT infrastructure** → **Actual Configuration Items**, type `sys.linux.linuxunitarycomputersystem` under Classifications, and then, press Enter.
- The Actual CIs that we want to promote are displayed, as shown in Figure 5-6 on page 140.

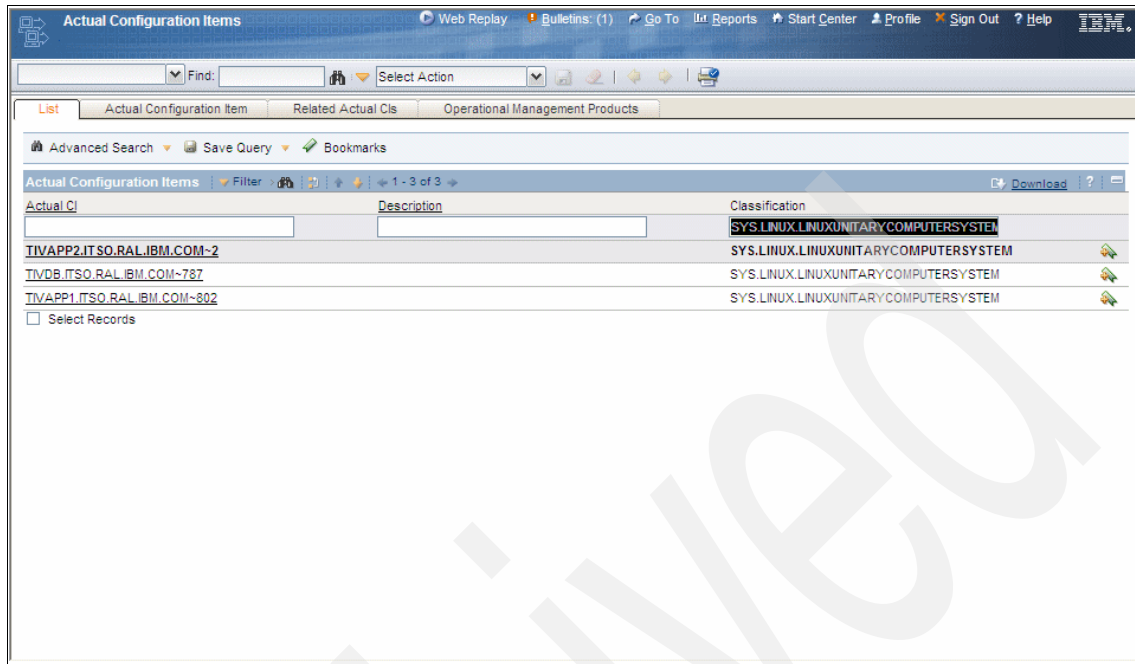


Figure 5-7 Actual CIs to promote

3. Choose the **Select Records** check box, and choose all computers that we want to promote to CI. Then, from the Select Action menu, select **Create Authorized Configuration Items**, as shown in Figure 5-8, and click **OK**.

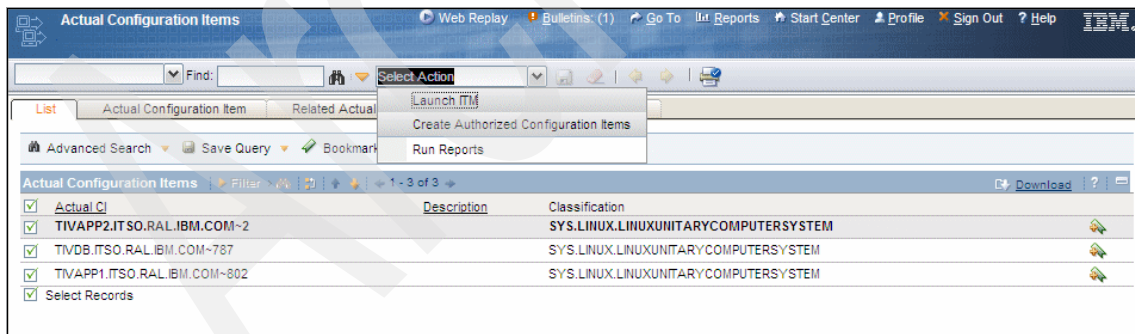
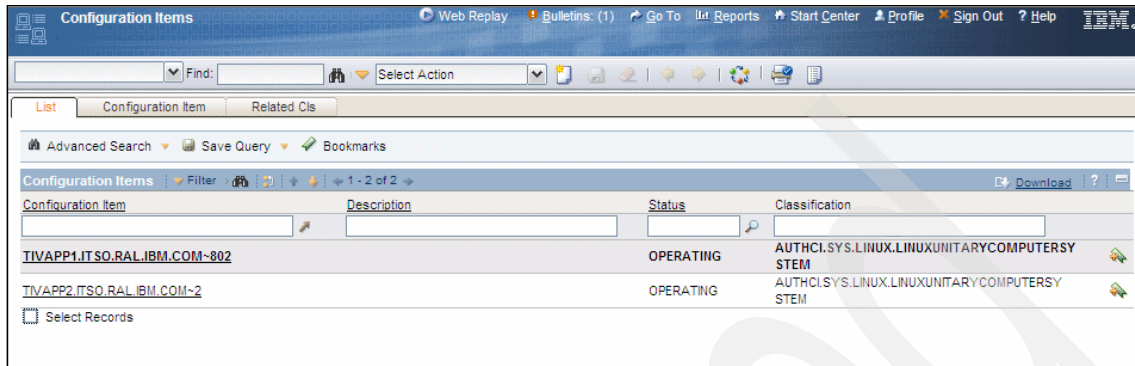


Figure 5-8 Promote Actual CIs to Authorized CIs

4. After the CIs are promoted, select **Go To** → **IT infrastructure** → **Configuration Items**, and choose the **Select Records** check box, and then, choose all CIs. Next, select **Change Status** from the **Select Action** menu. Select a New Status of **Operating** in the pop-up window, and click **OK**.

5. We can see the CIs' status changed to Operating, as shown in Figure 5-9 on page 142.



The screenshot shows the IBM Tivoli Configuration Items web interface. The top navigation bar includes links for Web Replay, Bulletins (1), Go To, Reports, Start Center, Profile, Sign Out, and Help. Below the navigation bar is a search and action area with a 'Find:' field, a 'Select Action' dropdown, and several icons. The main content area has tabs for 'List', 'Configuration Item', and 'Related CIs'. Under the 'List' tab, there are links for 'Advanced Search', 'Save Query', and 'Bookmarks'. A table titled 'Configuration Items' displays two items. The table has columns for 'Configuration Item', 'Description', 'Status', and 'Classification'. Both items have a status of 'OPERATING'. The first item is 'TIVAPP1.ITSO.RAL.IBM.COM~802' and the second is 'TIVAPP2.ITSO.RAL.IBM.COM~2'. Both are classified as 'AUTHCI.SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM'. There is a 'Download' link and a 'Select Records' checkbox at the bottom of the table.

Configuration Item	Description	Status	Classification
TIVAPP1.ITSO.RAL.IBM.COM~802		OPERATING	AUTHCI.SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM
TIVAPP2.ITSO.RAL.IBM.COM~2		OPERATING	AUTHCI.SYS.LINUX.LINUXUNITARYCOMPUTERSYSTEM

Figure 5-9 Change CIs' status to Operating

5.2.3 IBM Tivoli Provisioning Manager integration

With IBM Tivoli Provisioning Manager, you can leverage additional component data that is discovered by IBM Tivoli Application Dependency Discovery Manager. IBM Tivoli Provisioning Manager provides asset inventory data, which can be further enriched with the data discovered through IBM Tivoli Application Dependency Discovery Manager for the same CI. Also, you can use the discovery engine that is provided by IBM Tivoli Provisioning Manager to add more CIs to IBM Tivoli Application Dependency Discovery Manager database.

There is a two-way integration of IBM Tivoli Application Dependency Discovery Manager with IBM Tivoli Provisioning Manager:

- ▶ IBM Tivoli Provisioning Manager Discovery Library Adapter can generate an IDML book to populate IBM Tivoli Application Dependency Discovery Manager.
- ▶ IBM Tivoli Application Dependency Discovery Manager discovery can be triggered by using the IBM Tivoli Provisioning Manager GUI and can collect data directly into IBM Tivoli Provisioning Manager. This approach provides more granular details to IBM Tivoli Provisioning Manager for those CIs.

Discovery Library Adapter

IBM Tivoli Provisioning Manager must know about all systems in the network so that it can make sure that all systems contain the correct patch level. You achieve this end by getting discovery information and learning about existing systems from IBM Tivoli Application Dependency Discovery Manager or other sources, including IBM Tivoli Provisioning Manager's own discovery mechanism.

Data integration also plays an important role in GUI integration between other Operational Management Products or Process Management Products and IBM Tivoli Provisioning Manager. It makes sure that attributes, such as server name, of certain objects are consistent across all systems.

IBM Tivoli Provisioning Manager provides Discovery Library Adapter to export provisioning resources based on a provisioning data model into the discovery library book, which is based on the Common Data Model.

After the discovery library book is created, it can then be uploaded and consumed by other IBM Service Management products, such as IBM Tivoli Change and Configuration Management Database:

1. Open IBM Tivoli Provisioning Manager console, and navigate to **Go To → Administration → Provisioning → Provisioning Workflows**.
2. Type `DiscoveryLibrary` into the Provisioning Workflow entry, and press Enter.
3. You see the list of discovery library adapter-related workflows. Click the **DiscoveryLibraryAdapterExportByDevices** link.
4. Click the **Select Action** drop-down list and choose **Run Workflow**.
5. For the parameters for the workflow (Figure 5-10), enter these values:
 - `DiscoveryLibraryBookRepositoryPath`: This field is the target location for IDML books, we enter `C:\`.
 - `ServerList`: List of servers whose resource information will be written into the discovery library book. We specify: `tivapp1.itso.ral.ibm.com`, `tivapp2,itso.ral.ibm.com`, `tpm.itso.ral.ibm.com`.

Parameter name	Is Array	Parameter value
DiscoveryLibraryBookRepositoryPath	N	C:\
ServerList	Y	tivapp1.itso.ral.ibm.com,tivapp2,itso.ral.ibm.com,tpm.itso.ral.ibm.com

Hide Encrypted Input? ☒

Data Model Object Finder

Scheduling

Schedule

Run Cancel

Figure 5-10 DLA provisioning parameters

6. Click **Run** to execute the workflow, and then, click **Yes** to open Provisioning Task Tracking to obtain the status of the workflow.

7. In the Provisioning Task Tracking dialog, click the refresh icon to check the latest status. When the status is Success, then you can check the discovery library book in the destination path.

Example 5-2 shows the number of discovery library books produced by IBM Tivoli Provisioning Manager in the C:\ directory. The latest book that we have produced is at the bottom of the list.

Example 5-2 List of discovery library books

Directory of C:\

```
05/05/2009  11:51 AM 8,974 ITPM71.tpm.2009-05-05T15.51.52.484Z.xml
05/05/2009  01:29 PM 1,055 ITPM71.tpm.2009-05-05T17.29.09.250Z.xml
05/05/2009  01:31 PM 16,771 ITPM71.tpm.2009-05-05T17.31.03.437Z.xml
05/05/2009  02:26 PM 60,992 ITPM71.tpm.2009-05-05T18.26.25.625Z.xml
05/06/2009  12:07 PM 45,612 ITPM71.tpm.2009-05-06T16.07.34.968Z.xml
05/08/2009  05:48 PM 60,992 ITPM71.tpm.2009-05-08T21.48.17.718Z.xml
```

The next step is to copy the file to the IBM Tivoli Application Dependency Discovery Manager machine and to use the bulk upload API to upload the files into the discovery database:

1. Copy the DLA book that we created in Example 5-2 to a directory on your IBM Tivoli Application Dependency Discovery Manager server. We use the /TADDM_Integration path for storing the files.
2. Ensure that the bulk loading function of IBM Tivoli Application Dependency Discovery Manager server is configured by editing the \$COLLATION_HOME/dist/etc/bulkload.properties file, according to the instructions in the \$COLLATION_HOME/dist/etc/bulkload.readme file.
3. Run the Bulk loader command, as shown in Example 5-6 on page 152.

Example 5-3 Running the bulk loader

```
# su - cmdbadmin
> ./loadidm1.sh -f
/TADDM_Integration/ITPM71.tpm.2009-05-05T18.26.25.625Z.xml -u
administrator -p collation
```

```
Bulk Load Program starting.
Bulk Load Program running.
Bulk Load Program running.
Bulk Load Program succeeded. Return code is: 0
```

```
Bulk Load Program ending.
```

The discovery of a Windows machine is collected from the IBM Tivoli Provisioning Manager discovery, as shown in Figure 5-11.

The screenshot shows the 'Details' window in the IBM Tivoli Provisioning Manager. The 'Items' list shows 'tpm.itso.ra1.ibm.com'. The 'OS' tab is selected, displaying the following information:

OS Name:	Microsoft Windows Server 2003
OS Confidence Level:	
OS Version:	2003
Kernel Architecture:	
Kernel Version:	
System Id:	
DNS Resolve Entries:	

Figure 5-11 Windows machine imported into IBM Tivoli Application Dependency Discovery Manager

Also, you can see the software that is installed by looking at the IBM Tivoli Provisioning Manager Software Installed software inventory list, as shown in Figure 5-12.

The screenshot shows the 'Details' window in the IBM Tivoli Provisioning Manager. The 'Software Installed' tab is selected, displaying a table of installed software:

Software Name	Version	Architecture	Manufacturer	Path	Size
ECLIPSE SDK [Version 3.1.2.0]	3.1.2.0	unknown			
ECLIPSE SDK [Version 999.999.0.0]	999.999.0.0	unknown			
Executive Software Diskeeper [Version 5.2.3790.0]	5.2.3790.0	unknown			
Executive Software Diskeeper [Version 5.2.3790.1830]	5.2.3790.1830	unknown			
IBM DB2 Enterprise Server Edition [Version 9.1.400.359]	9.1.400.359	IBM			
IBM SDK 5.0 for Windows IA32, Java Technology [Version 5.0.0.0]	5.0.0.0	IBM			
IBM WebSphere Application Server Network Deployment [Version 6.1.0]	6.1.0	IBM			
IPX and SPX Protocol Stacks [Version 3.02.0.0]	3.02.0.0	unknown			
JRE [Version 1.4]	1.4	unknown			

Figure 5-12 Software discovery by IBM Tivoli Provisioning Manager

Using IBM Tivoli Application Dependency Discovery Manager discovery from IBM Tivoli Provisioning Manager

Another integration aspect is to bring the IBM Tivoli Application Dependency Discovery Manager data into IBM Tivoli Provisioning Manager. This integration uses the IBM Tivoli Application Dependency Discovery Manager APIs to load resources and populate IBM Tivoli Provisioning Manager:

1. From the IBM Tivoli Provisioning Manager Web interface, select **Go To** → **Discovery** → **Provisioning Discovery** → **Discovery Configuration**, as shown in Figure 5-13.

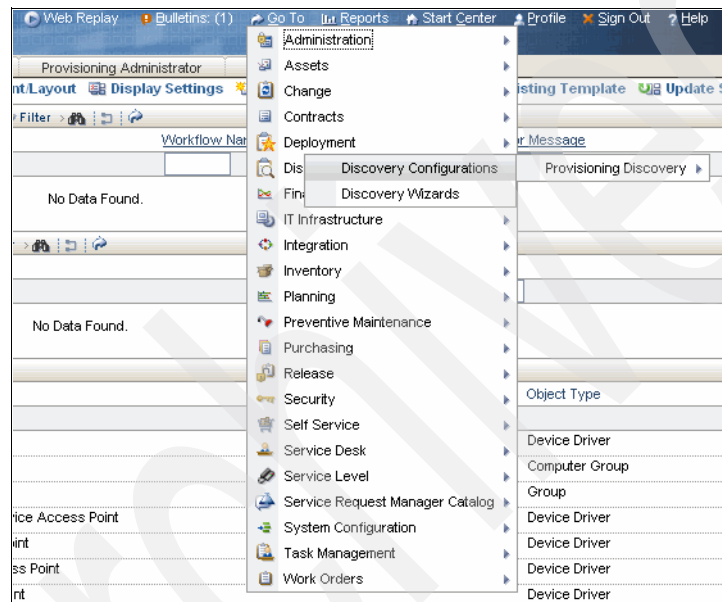


Figure 5-13 Select Discovery Configurations

2. Select the Tivoli Application Dependency Discovery Manager Discovery from the list, as shown in Figure 5-14 on page 147.

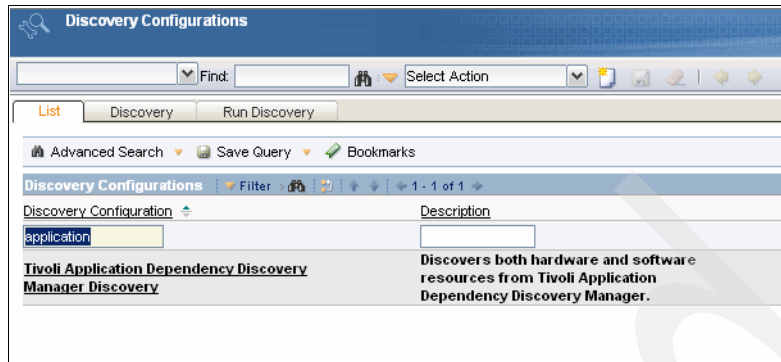


Figure 5-14 Select IBM Tivoli Application Dependency Discovery Manager discovery from the list

3. Type the IBM Tivoli Application Dependency Discovery Manager server host name, port, user ID, and password, as shown in Figure 5-15.

The screenshot shows the 'Discovery Parameters' window with the 'Server Information' tab selected. The 'Computer to be Discovered' and 'Scope' tabs are also visible. The 'Server Information' section contains four fields: 'Name' (taddm.itso.ral.ibm.com), 'User Id' (administrator), 'Port number' (9530), and 'Password' (masked with dots).

Server Information	
Name	taddm.itso.ral.ibm.com
User Id	administrator
Port number	9530
Password	

Figure 5-15 IBM Tivoli Application Dependency Discovery Manager discovery parameters

4. You can define a filter on the Computer to be Discovered tab, as shown in Figure 5-16.

The screenshot shows the 'Discovery Parameters' window with the 'Computer to be Discovered' tab selected. The 'Server Information' and 'Scope' tabs are also visible. The 'Computer to be Discovered' section contains a table with one row: 'Fully-qualified Server Name' with the value 'itm.itso.ral.ibm.com'.

Computer to be Discovered	
Fully-qualified Server Name	itm.itso.ral.ibm.com

Figure 5-16 Define a filter for discovery

5. On the Run Discovery tab, you can start the discovery by clicking **Submit**, as shown in Figure 5-17 on page 148.

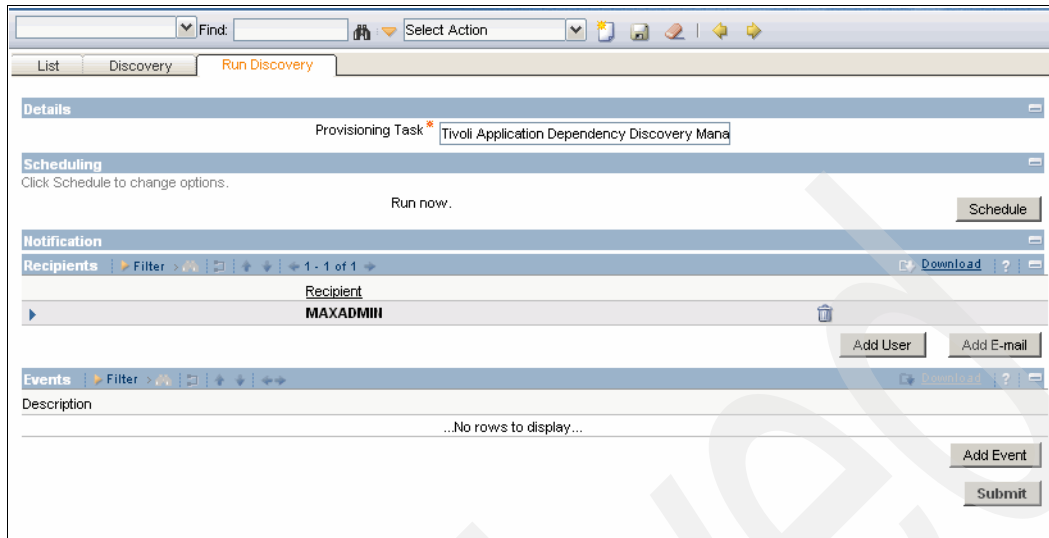


Figure 5-17 Run the discovery from IBM Tivoli Provisioning Manager

6. To check the discovery status, click **Go To** → **Task Management** → **Provisioning Tasks** → **Provisioning Task Tracking**. See Figure 5-18.

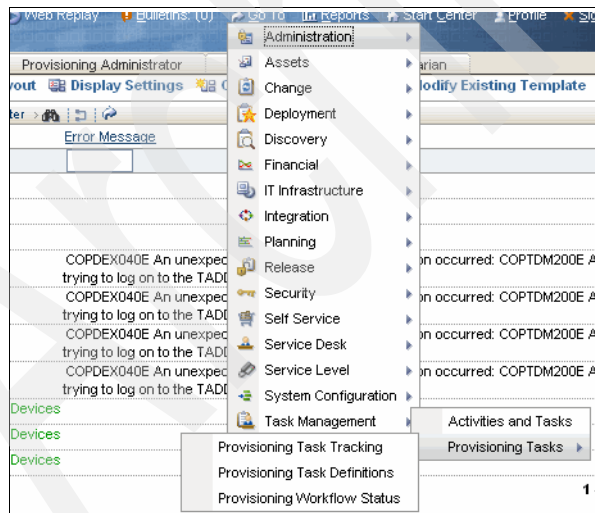


Figure 5-18 Open Provisioning Task Tracking

7. Filter the list to see IBM Tivoli Application Dependency Discovery Manager discovery, as shown in Figure 5-19 on page 149.

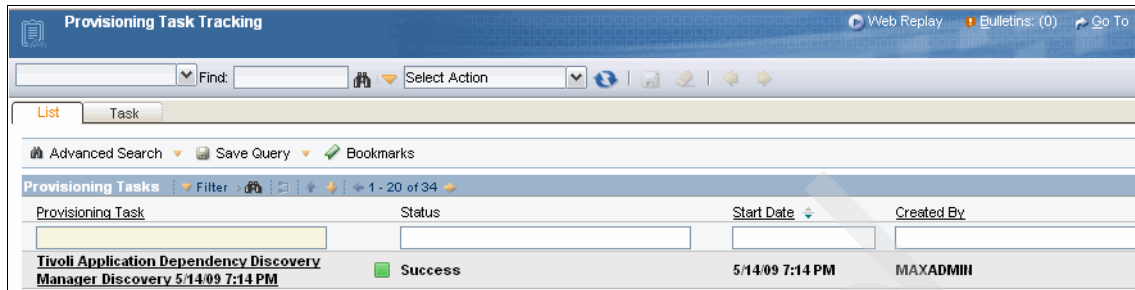


Figure 5-19 Check the Discovery status

8. To check the discovered CIs, click **Go To** → **IT Infrastructure** → **Provisioning Inventory** → **Provisioning Computer**. See Figure 5-20.

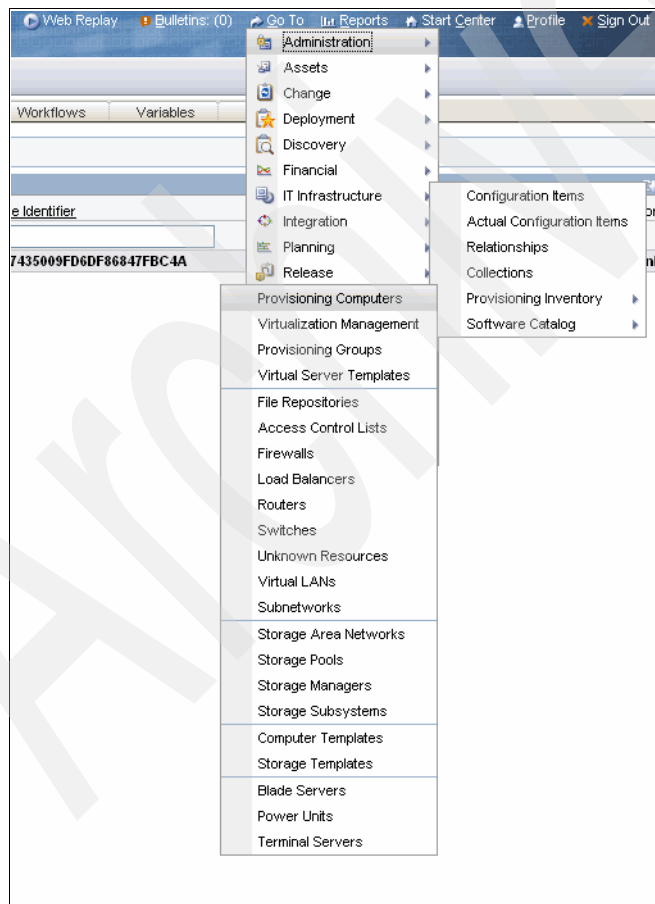


Figure 5-20 Open the Provisioning Computer

9. Check the newly discovered machine, as shown in Figure 5-21.

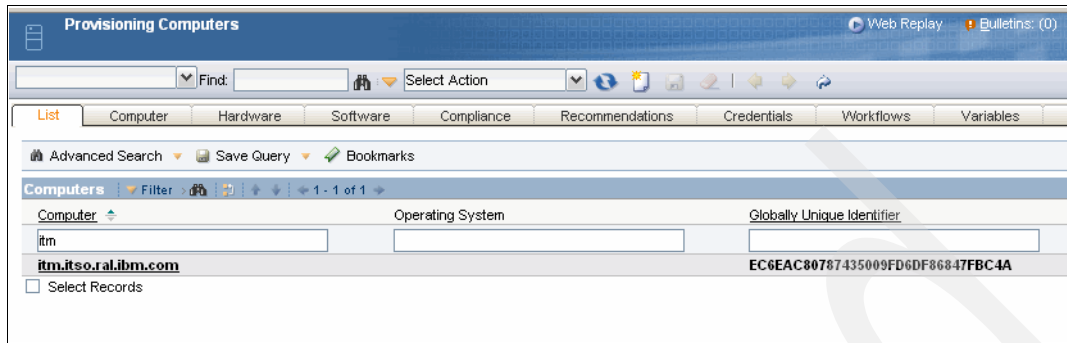


Figure 5-21 Check the newly discovered CI from IBM Tivoli Application Dependency Discovery Manager

Because the computer system has already been discovered from IBM Tivoli Application Dependency Discovery Manager, IBM Tivoli Provisioning Manager can use the data for this CI by using APIs and will not run its own discovery again.

5.2.4 IBM Tivoli Monitoring integration

In enterprise environments, there are instances when you add new servers or reinstall existing servers. You need to redeploy the IBM Tivoli Monitoring Tivoli Enterprise Monitoring Agents on these servers. In large environments, it becomes difficult to keep track of server targets that are not monitored. With IBM Tivoli Monitoring V6.2, you can now leverage the integration between IBM Tivoli Application Dependency Discovery Manager and IBM Tivoli Monitoring to discover targets that do not have IBM Tivoli Monitoring agents installed.

Discovery library adapter

The IBM Tivoli Application Dependency Discovery Manager monitoring coverage report highlights unmonitored resources by contrasting data that is extracted from IBM Tivoli Monitoring with other discovered data. In this section, we explain the steps that are necessary to use the IBM Tivoli Application Dependency Discovery Manager Monitoring coverage report. The report is intended for use in conjunction with the IBM Tivoli Monitoring discovery library adapter.

IBM Tivoli Application Dependency Discovery Manager is an extremely useful tool to dynamically discover and collect asset information from assets in the enterprise with no need for agents. IBM Tivoli Monitoring is a recognized tool that is able to monitor, manage, collect, and generate events for a huge number of IT resources.

It is necessary to export monitoring information from IBM Tivoli Monitoring to IBM Tivoli Application Dependency Discovery Manager by using a discovery library adapter (DLA). The DLA is installed with IBM Tivoli Monitoring. The data generated by the DLA is in IDML format and must be copied to the IBM Tivoli Application Dependency Discovery Manager. The DLA must then be loaded using the **loadidm1** script.

The information that is generated by the Tivoli Management Services DLA allows IBM Tivoli Application Dependency Discovery Manager to provide a contextual launch to Tivoli Enterprise Portal. You can also view the status of the discovered managed systems while in IBM Tivoli Application Dependency Discovery Manager Console.

When IBM Tivoli Change and Configuration Management Database synchronizes its CIs from IBM Tivoli Application Dependency Discovery Manager through IBM Tivoli Integration Composer, it is also possible to have a contextual launch from IBM Tivoli Change and Configuration Management Database to Tivoli Enterprise Portal:

1. Execute the **tmsd1a** script to export IBM Tivoli Monitoring information about managed systems. The DLA gathers information by querying the hub monitoring server for all managed systems and mapping them to Common Data Model resources based on the agent product code and managed system name format.
2. In our environment, it was necessary to adjust the **tmsd1a.sh** script that comes as part of IBM Tivoli Monitoring to work correctly. Example 5-4 shows the modified `/opt/IBM/ITM/li6263/cq/bin/tmsdla.sh` script. The changes are in bold.

Note: This changes is not needed on the GA version of IBM Tivoli Monitoring V6.2.2.

Example 5-4 Modified tmsdla.sh script

```
#!/bin/sh
export JAVA_HOME=/opt/IBM/ITM/JRE/li6263
export KFW_HOME=/opt/IBM/ITM
export CLASSPATH=${KFW_HOME}/classes:${KFW_HOME}/li6263/cq/lib/tmsdla.jar:
${KFW_HOME}/classes/cnp.jar:${KFW_HOME}/classes/cnp_vbjorball.jar:${KFW_HOME}/classes/kjrall.jar:${KFW_HOME}/classes/util.jar
for jarfile in ${KFW_HOME}/li6263/cq/bin/tmsdla/*.jar; do
    export CLASSPATH=${CLASSPATH}:${jarfile}
done
export CNPS_ADDR=localhost
```

```
export TRACEPARMS="ERROR (UNIT:PBasedRequest DETAIL) (UNIT:DataBus ALL)
(UNIT:TEPSRetriever ALL)"
```

```
${JAVA_HOME}/bin/java -Xms64m -Xmx128m -noverify -classpath
${CLASSPATH} -Dcnp.http.url.host=${CNPS_ADDR}
-Dvbroker.agent.enableLocator=false -Dkjr.trace.mode=LOCAL
-Dkjr.trace.file=FZZSRAS1.LOG -Dkjr.trace.params="${TRACEPARMS}"
-DUSER=sysadmin com.ibm.tivoli.monitoring.tmsdla.TmsDla $*
```

3. In Example 5-5, we show the execution of the **tmsdla.sh** script.

Example 5-5 Execution of the tmsdla.sh script

```
# cd /opt/IBM/ITM/li6263/cq/bin
# ./tmsdla.sh
No errors during run.
```

4. The results are an IDML file in the `/opt/IBM/ITM/<interp>/cq/bin/tmsdla` directory. In our environment, the path is `/opt/IBM/ITM/li6263/cq/bin/tmsdla/` with the name of `TMSDISC620-A.itm.itso.ral.ibm.com.2009-05-07T16.46.20Z.refresh.xml`. The log files are saved in the `/opt/IBM/ITM/li6263/cq/bin` directory. The log files generated are the `tmsdla.queries.log` file, the `tmsdla.log` file, and the `FZZSRAS1.LOG` file.
5. Copy the XML file to the IBM Tivoli Application Dependency Discovery Manager domain server. The XML file is transferred to the Discovery Library File Store path in the IBM Tivoli Application Dependency Discovery Manager domain server. In a production environment, this task must be automated with a scheduling system, such as Tivoli Workload Scheduler.
6. Ensure that the bulk loading function of IBM Tivoli Application Dependency Discovery Manager server is configured by editing the `$COLLATION_HOME/dist/etc/bulkload.properties` file, according to the instructions in the `$COLLATION_HOME/dist/etc/bulkload.readme` file.
7. Run the Bulk loader command, as shown in Example 5-6.

Example 5-6 Running the bulk loader

```
su - cmdbadmin
./loadidml.sh -f
/TADDM_Integration/TMSDISC620-A.itm.itso.ral.ibm.com.2009-05-07T16.46.2
0Z.refresh.xml -u administrator -p collation
```

```
Bulk Load Program starting.
Bulk Load Program running.
Bulk Load Program running.
```

Bulk Load Program succeeded. Return code is: 0

Bulk Load Program ending.

Running the monitoring report

With the monitoring environment information loaded in IBM Tivoli Application Dependency Discovery Manager, you can discover which system IBM Tivoli Monitoring is monitoring, and you can create these relationships in IBM Tivoli Application Dependency Discovery Manager using the Tivoli Common Data Model. Figure 5-22 on page 154 shows the Monitoring coverage report, which is retrieved by clicking **Analytics** → **Monitoring Coverage Report**.

Application Manager for Web Resources appears in the monitoring report, but it is not linked to the actual WebSphere Application Server CI object in IBM Tivoli Application Dependency Discovery Manager.

The IDML books are created based on the `klz_tmsdla.xml` file, the `knt_tmsdla.xml` file, and the `kux_tmsdla.xml` file that provide the operating system agent mapping of the `app.TMSAgent` object back to the actual computer system object.

5.2.5 IBM Tivoli Business Service Manager integration

This integration imports data from IBM Tivoli Application Dependency Discovery Manager into IBM Tivoli Business Service Manager. The data import includes IBM Tivoli Application Dependency Discovery Manager Business Applications and Business Services, which will be added to the IBM Tivoli Business Service Manager's list of services. The data import also includes the physical IBM Tivoli Application Dependency Discovery Manager CIs, which will be added to IBM Tivoli Business Service Manager's Service Component Repository.

The IBM Tivoli Business Service Manager Discovery Library Toolkit acts as a bridge between the IBM Common Data Model and the template-based model of IBM Tivoli Business Service Manager. The toolkit can either interface with IBM Tivoli Application Dependency Discovery Manager or read Discovery Library Adapter (DLA) books directly.

The toolkit consists of a process that collects information from the Common Data Model, transforms the information, and stores the data in the IBM Tivoli Business Service Manager's Service Component Repository. The Service Component Repository is then accessed using Enhanced SQL Data Adapter to show that data has been stored in IBM Tivoli Business Service Manager server's data store.

We have two integration options between IBM Tivoli Application Dependency Discovery Manager and IBM Tivoli Business Service Manager:

- ▶ Create the business service inside IBM Tivoli Business Service Manager and then link this business service to the CIs discovered by IBM Tivoli Application Dependency Discovery Manager.
- ▶ Create the business service inside IBM Tivoli Application Dependency Discovery Manager, which creates the service views into IBM Tivoli Business Service Manager automatically.

Defining business services in IBM Tivoli Business Service Manager allows the user to combine resource objects that were discovered by IBM Tivoli Application Dependency Discovery Manager with other resource objects created in IBM

Tivoli Business Service Manager through other methods, such as auto-population or RADShell.

Figure 5-23 on page 156 shows what we have discovered using IBM Tivoli Application Dependency Discovery Manager.

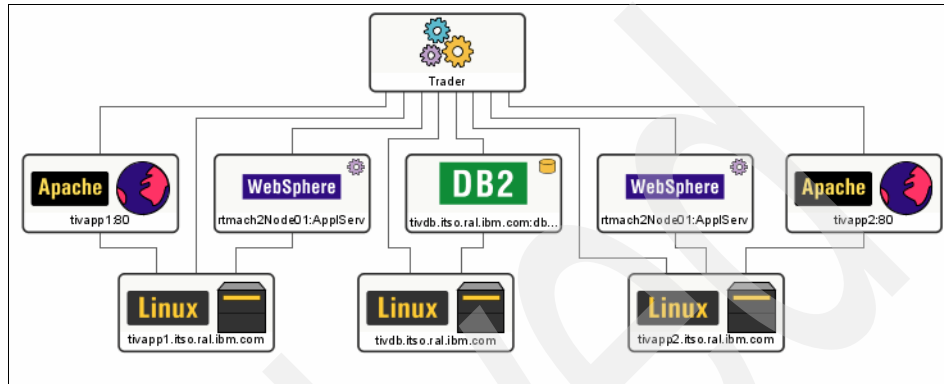


Figure 5-23 IBM Tivoli Application Dependency Discovery Manager discovered CIs

The IBM Tivoli Business Service Manager and IBM Tivoli Application Dependency Discovery Manager data integration is shown in Figure 5-24.

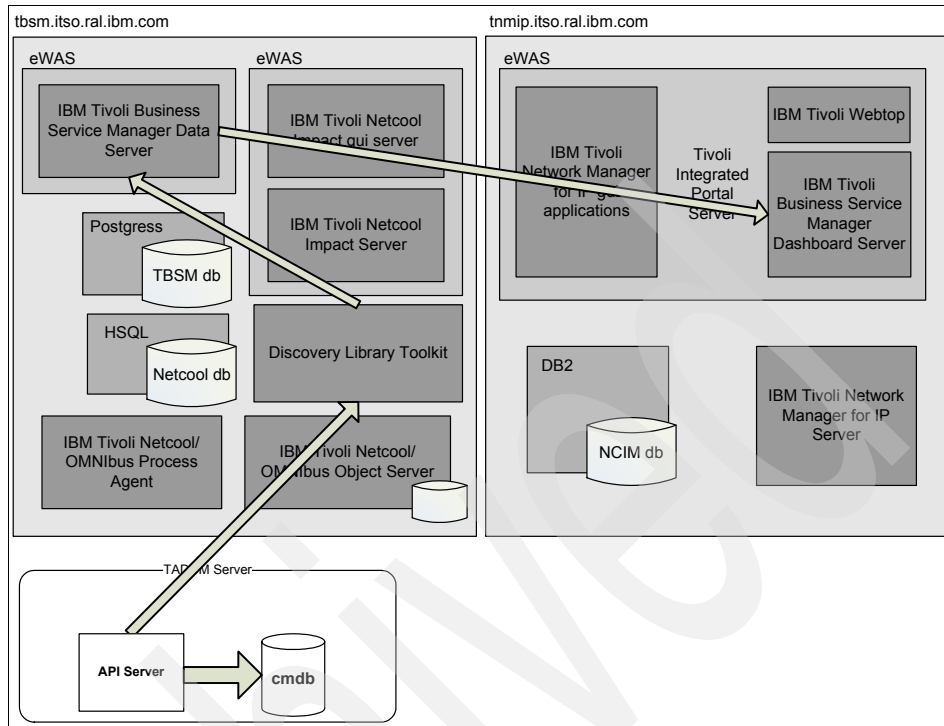


Figure 5-24 Installed architecture

The Discovery Library Toolkit imports data from IBM Tivoli Application Dependency Discovery Manager into IBM Tivoli Business Service Manager. The Discovery Library Toolkit runs as a process to automatically refresh the IBM Tivoli Business Service Manager data from IBM Tivoli Application Dependency Discovery Manager.

Implementation

The Discovery Library Toolkit installation process configures the toolkit to connect IBM Tivoli Application Dependency Discovery Manager and imports the data into IBM Tivoli Business Service Manager. To install the Discovery Library Toolkit, perform the following steps:

1. As the netcool user, from the IBM Tivoli Business Service Manager installation media/linux/DiscoveryLibrary directory, issue **./setup-linux.bin** to launch the Java installer.
2. Select the language to use during the installation, and click **OK**.
3. Click **Next** to begin the installation.
4. Accept the license agreement, and click **Next**.

5. Browse the installation directory, and click **Next**. The default value is the \$TBSM_HOME/XMLtoolkit directory.
6. If this installation is a new installation and there is no export directory, select “No, I do not have an “export” directory”, and click **Next**.
7. Specify these IBM Tivoli Business Service Manager configuration parameters, and click **Next**:
 - TBSM User ID: Administrative user ID for the Data Server
 - TBSM Password: Administrative user ID’s password for the Data Server
 - Data server host name: Fully qualified host name of the Data Server
 - Data server port: Data Server port specified in the \$TBSM_HOME/etc/rad/RAD_server.props file. The default value is 17310.

Figure 5-25 on page 158 illustrates the configured Data Server parameters.

IBM Tivoli Business Service Manager 4.2 Discovery Library toolkit

Configuration definitions used by the toolkit to connect to the TBSM Data server.
If a hostname is specified, it must be a fully qualified hostname.

Enter the TBSM User ID:
tipadmin

Enter the TBSM password:

Confirm the TBSM password:

Enter the TBSM data server hostname or IP address:
tbsm.itso.ral.ibm.com

Enter the TBSM data server port:
17310

InstallShield

< Back Next > Cancel

Figure 5-25 Data Server parameters

8. Select the data source to use. Specify the IBM Tivoli Application Dependency Discovery Manager data source when IBM Tivoli Application Dependency Discovery Manager is installed in the environment. The PostgreSQL IBM Tivoli Business Service Manager database is selected by default. Click **Next**.

Figure 5-26 on page 159 illustrates selecting IBM Tivoli Application Dependency Discovery Manager as the data source.

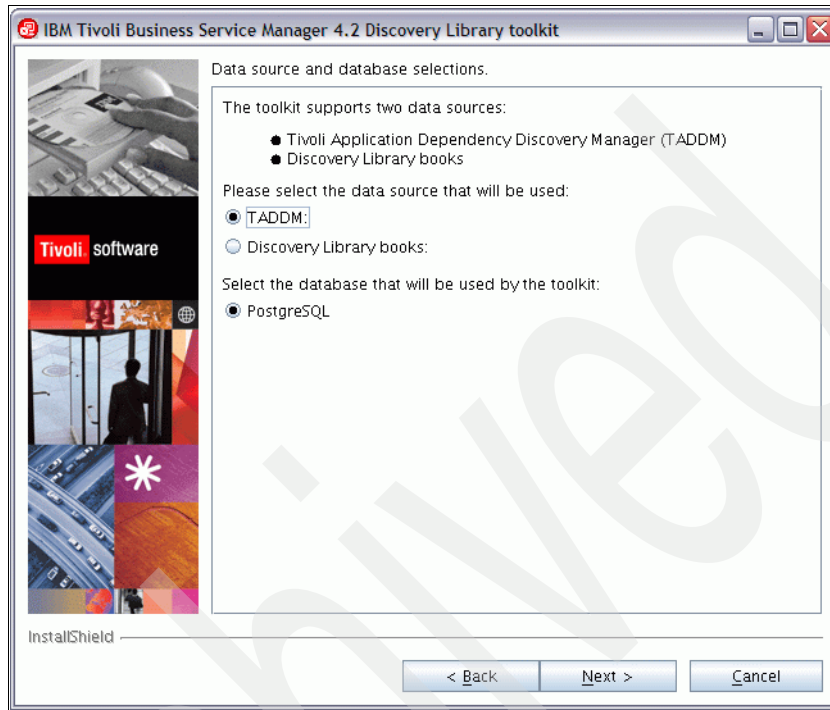


Figure 5-26 IBM Tivoli Application Dependency Discovery Manager data source

9. Specify the IBM Tivoli Business Service Manager PostgreSQL configuration parameters, and click **Next**:
 - Database User ID: For UNIX, this user ID is the system user who installed IBM Tivoli Business Service Manager.
 - Database password: Type the system password for the Database user ID.
 - Database server: Type the fully qualified host name for the PostgreSQL server.
 - Database port: Type the PostgreSQL database port. The default port is 5435.

Figure 5-27 on page 160 illustrates the database parameters.

IBM Tivoli Business Service Manager 4.2 Discovery Library toolkit

Database configuration. The database user is the internal PostgreSQL user, not the user that the service runs under. The default user on Unix is the user that the TBSM server was installed under, on Windows it is 'postgres'.

Enter the database User ID:
netcool

Enter the database password:

Confirm the database password:

Enter the database hostname or IP Address:
tbsm.itso.ral.ibm.com

Enter the port used by the database:
5435

InstallShield

< Back Next > Cancel

Figure 5-27 Database parameters

10. Specify the IBM Tivoli Application Dependency Discovery Manager configuration parameters, and click **Next**:

TADDM User ID	IBM Tivoli Application Dependency Discovery Manager user with supervisory authority.
TADDM Password	IBM Tivoli Application Dependency Discovery Manager user password.
TADDM Hostname	Fully qualified host name of IBM Tivoli Application Dependency Discovery Manager server.
TADDM Port	IBM Tivoli Application Dependency Discovery Manager Remote Method Invocation (RMI) port. The default port is 9530.

Figure 5-28 on page 161 illustrates the IBM Tivoli Application Dependency Discovery Manager parameters.

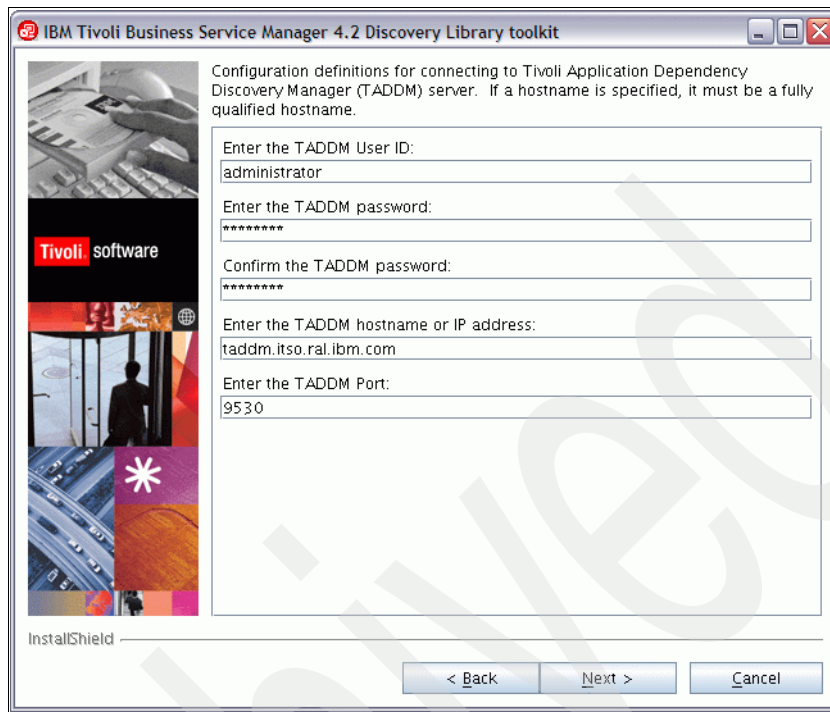


Figure 5-28 IBM Tivoli Application Dependency Discovery Manager parameters

11. Browse the Discovery Library book import directory, and click **Next**. The default value is the \$TBSM_HOME/discovery/dlbooks directory.
12. Browse the Discovery Library book export directory, which can be the same directory as the import directory. The default value is the \$TBSM_HOME/discover/dlbooks directory. Enter the fully qualified IBM Tivoli Business Service Manager Dashboard server host name and port. The default port is 16316. Click **Next**.
13. To add the Discovery Library Toolkit to start at system start-up, root authority is required. Select whether root authority can be used to configure the Discovery Library Toolkit automatic start-up now. If Discovery Library Toolkit automatic start-up is not configured now, you can run the tbsmrdr_enable.sh script later. Click **Next**.
14. Click **Next** to begin the installation.
15. When the Discovery Library Toolkit is installed, click **Finish**.

Configuring the Discovery Library Toolkit

To configure the Discovery Library Toolkit, you must perform the following tasks:

- ▶ From the IBM Tivoli Application Dependency Discovery Manager V7.1.1 server, copy the `/opt/IBM/cmdb/dist/sdk/clientlib/taddm-api-client.jar` file to the IBM Tivoli Business Service Manager Discovery Library Toolkit server in the `$TBSM_HOME/XMLtoolkit/sdk/clientlib` directory.
- ▶ Create the Discovery Library Toolkit service templates. On the IBM Tivoli Business Service Manager Data Server as the `netcool` user, issue the `cat $TBSM_HOME/install/BSM_Templates.radsh | $TBSM_HOME/bin/rad_radshe11` command.

Starting the Discovery Library Toolkit

To start the toolkit, from the `$TBSM_HOME/XMLToolkit/bin` directory, issue the `./tbsmrdr_start.sh` command. Example 5-7 illustrates starting the Discovery Library Toolkit in our environment.

Example 5-7 Discovery Library Toolkit start-up

```
[netcool@tbsm bin]$ ./tbsmrdr_start.sh
Starting tbsmrdrtk2: [ OK ]
```

The Discovery Library Toolkit automatically connects to IBM Tivoli Application Dependency Discovery Manager and performs a bulk insert of all discovered objects into IBM Tivoli Business Service Manager.

Verification

On the Discovery Library Toolkit server, check the `$TBSM_HOME/XMLtoolkit/log/msgGTM_XT.log` file. Successful bulk import is indicated by the following message:

“GTMCL5293I: CMDB import completed successfully.”

Verify that new services are available in IBM Tivoli Business Service Manager. New services must be subscribed to the service tree.

Note: If the import is unsuccessful due to an incorrect user ID or password, this access can be configured after installation with the `setxmlaccess.sh` command:

```
setxmlaccess.sh -U TBSMInstallUser:TBSMAdmin:TADDMAAdmin -P
TBSMInstallPassword:TBSMAdminPassword:TADDMPassWord
```


5.2.6 IBM Tivoli Network Manager for IP integration

The IBM Tivoli Network Manager for IP and IBM Tivoli Application Dependency Discovery Manager data integration allows objects in IBM Tivoli Network Manager for IP to be defined in IBM Tivoli Application Dependency Discovery Manager and objects in IBM Tivoli Application Dependency Discovery Manager to be defined in IBM Tivoli Network Manager for IP.

There are two options to export objects from IBM Tivoli Network Manager for IP to IBM Tivoli Application Dependency Discovery Manager:

- ▶ Manually run the IBM Tivoli Network Manager for IP Discovery Library Adapter to export the configuration and run the IBM Tivoli Application Dependency Discovery Manager bulk loader to import the objects.
- ▶ Automatically run the discovery components through IBM Tivoli Directory Integrator.

One of these data integrations must be performed prior to a Launch in Context integration from IBM Tivoli Network Manager for IP launching to IBM Tivoli Application Dependency Discovery Manager.

You must export objects from IBM Tivoli Application Dependency Discovery Manager and import them into IBM Tivoli Network Manager for IP. IBM Tivoli Network Manager for IP does not import IBM Tivoli Application Dependency Discovery Manager full entities. IBM Tivoli Network Manager for IP exports the network topology entities using the Discovery Library Adapter into IBM Tivoli Application Dependency Discovery Manager. The IBM Tivoli Application Dependency Discovery Manager shell script API is used to convert the GUID data for the entities originally sent by IBM Tivoli Network Manager for IP into a comma-separated value (CSV) file. The file is then imported by IBM Tivoli Network Manager for IP Discovery Library Adapter into an entityGUIDcache table in IBM Tivoli Network Manager for IP IBM Tivoli Netcool/Impact (NCIM) database. This GUID is then used by Webtop to dynamically build the URL to Launch In Context from IBM Tivoli Network Manager for IP into the IBM Tivoli Application Dependency Discovery Manager window.

The IBM Tivoli Application Dependency Discovery Manager to IBM Tivoli Network Manager for IP integration is required for an IBM Tivoli Business Service Manager launch to IBM Tivoli Network Manager for IP if IBM Tivoli Application Dependency Discovery Manager is integrated with IBM Tivoli Business Service Manager.

The IBM Tivoli Network Manager for IP and IBM Tivoli Application Dependency Discovery Manager automatic integration includes the following products:

- ▶ IBM Tivoli Directory Integrator

IBM Tivoli Directory Integrator is a stand-alone application that exchanges data between separate applications or data sources. IBM Tivoli Directory Integrator is used to automate the export of objects from IBM Tivoli Network Manager for IP and the import of objects into IBM Tivoli Application Dependency Discovery Manager. If the export and import will be performed manually, IBM Tivoli Directory Integrator is not required. IBM Tivoli Directory Integrator reads information from one or multiple data sources, reformats it in the specified format, and writes the information to one or multiple target applications. The IBM Tivoli Directory Integrator rules define a set of procedures that reformats the data in stages that are similar to an assembly line's stages. The integration currently supports IBM Tivoli Directory Integrator Version 6.1.1 with Fix Pack 1 or later.

- ▶ Discovery Library Integration Framework

Discovery Library Integration Framework provides IBM Tivoli Directory Integrator components, which can be used to import objects into IBM Tivoli Application Dependency Discovery Manager from multiple applications. This framework is available on the IBM Open Process Automation Library Web site.

- ▶ Discovery Library Integration Framework plug-in for IBM Tivoli Network Manager for IP

Discovery Library Integration Framework plug-in for IBM Tivoli Network Manager for IP provides the components that are used by the Discovery Library Integration Framework and IBM Tivoli Directory Integrator to export objects from IBM Tivoli Network Manager for IP. This framework is available on the IBM Open Process Automation Library Web site.

Figure 5-24 on page 157 illustrates IBM Tivoli Network Manager for IP and IBM Tivoli Application Dependency Discovery Manager manual data integration.

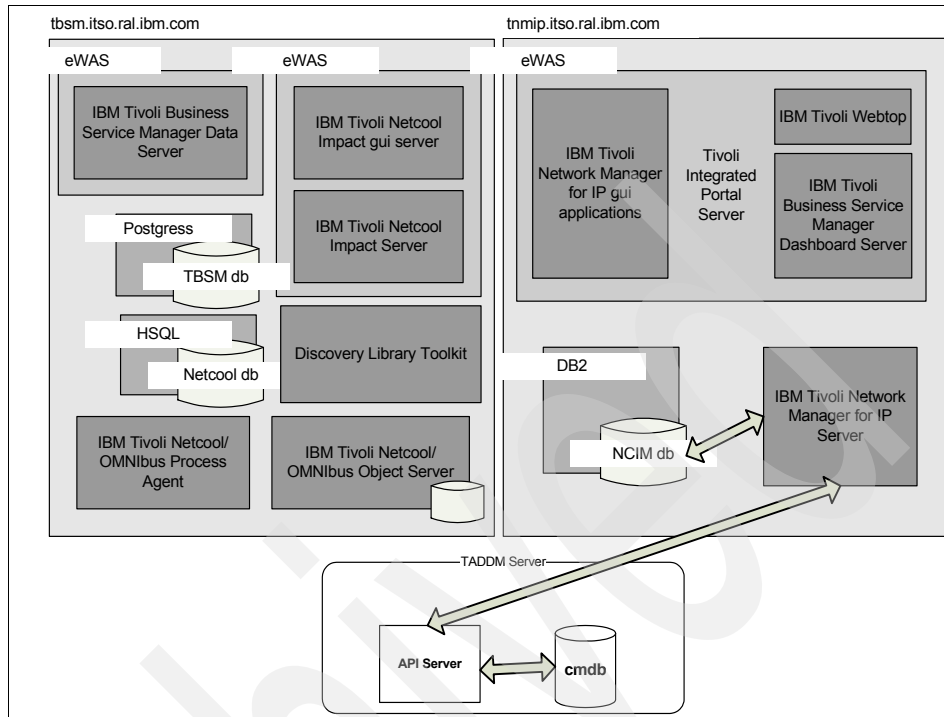


Figure 5-29 Installed architecture for manual integration

The data flow occurs this way in Figure 5-29:

1. On the IBM Tivoli Network Manager for IP system, export the devices manually by running Discovery Library Adapter to create an IDML book.
2. Copy the IDML book to the IBM Tivoli Application Dependency Discovery Manager server.
3. Import the IDML book using the IBM Tivoli Application Dependency Discovery Manager bulk loader.
4. On the IBM Tivoli Application Dependency Discovery Manager system, manually export the Computer System data.
5. Copy the exported data to the IBM Tivoli Network Manager for IP system.
6. Import the ComputerSystem data using the IBM Tivoli Network Manager for IP Discovery Library Adapter.
7. The process must be repeated manually to refresh the data.

Figure 5-30 illustrates the IBM Tivoli Network Manager for IP and IBM Tivoli Application Dependency Discovery Manager automatic data integration.

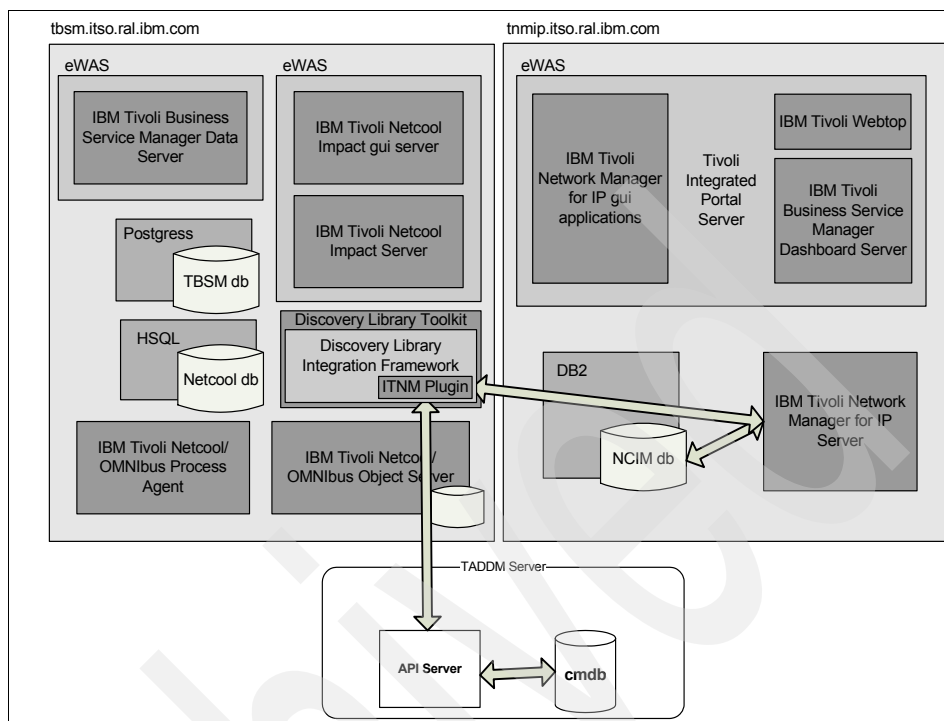


Figure 5-30 Installed architecture for automatic integration

In Figure 5-30, the data flows this way:

1. The IBM Tivoli Directory Integrator assembly line automatically exports IBM Tivoli Network Manager for IP devices.
2. The IBM Tivoli Directory Integrator assembly line automatically copies the IDML book to the IBM Tivoli Application Dependency Discovery Manager server.
3. The IBM Tivoli Directory Integrator assembly line automatically imports the IDML book using the IBM Tivoli Application Dependency Discovery Manager bulk loader.
4. The IBM Tivoli Directory Integrator assembly line runs on a configurable schedule to refresh data.

There is currently no automated option to automatically refresh data from IBM Tivoli Application Dependency Discovery Manager to IBM Tivoli Network Manager for IP.

Manually exporting data to IBM Tivoli Application Dependency Discovery Manager

The IDML book links objects within several Tivoli products. IDML books exported from IBM Tivoli Network Manager for IP can be imported into either IBM Tivoli Application Dependency Discovery Manager or IBM Tivoli Business Service Manager. If IBM Tivoli Business Service Manager is integrated with IBM Tivoli Application Dependency Discovery Manager, IBM Tivoli Network Manager for IP must be integrated with IBM Tivoli Application Dependency Discovery Manager as well to have a single object repository. If IBM Tivoli Business Service Manager is not integrated with IBM Tivoli Application Dependency Discovery Manager, IBM Tivoli Network Manager for IP can be directly integrated with IBM Tivoli Business Service Manager.

For this example, because we have integrated IBM Tivoli Business Service Manager and IBM Tivoli Application Dependency Discovery Manager (see 5.2.5, “IBM Tivoli Business Service Manager integration” on page 155), we integrate IBM Tivoli Network Manager for IP with IBM Tivoli Application Dependency Discovery Manager.

The Discovery Library Adapter is included with IBM Tivoli Network Manager for IP. The Discovery Library Adapter is used to import and export IDML books:

1. On the IBM Tivoli Network Manager for IP system, set the Discovery Library Adapter properties. Edit the `$NCHOME/precision/adapter/ncp_dla.properties` file.
2. Set the `ncp.dla.precisionDomain` property to the IBM Tivoli Network Manager for IP domain. If there are multiple domains, multiple property files can be created.
3. Set the `ncp.dla.contextualLaunchURL` property. IBM Tivoli Network Manager for IP provides three possible views, but only one view can be launched in context from IBM Tivoli Business Service Manager. Additional views can be added manually as generic URLs. You can use one of these views:
 - Hop View
`https://TIPhost:TIPport/ibm/console/ncp_topoviz/HopView.do?seed=`
 - Find in View
`https://TIPHost:TIPport/ibm/console/ncp_topoviz/FindInView.do?entityId=`
 - Structure Browser
`http://TIPHost:TIPport/ibm/console/ncp_structurereview/Launch.do?entityId=`

4. Run Discovery Library Adapter from the `$NCHOME/precision/adapters/ncp_dla` directory. Issue `./ncp_dla.sh ncp_dla.properties`. Example 5-8 illustrates exporting the IDML book.

Example 5-8 Exporting the IDML book

```
[netcool@tnmip ncp_dla]$ ./ncp_dla.sh ncp_dla.properties
ncp_DLA ( IBM Tivoli Network Manager IP Edition - Discovery Library
Adapter )
Copyright (C) 1997 - 2008 By IBM Corporation. All Rights Reserved.
See product license for details.

[IDML Generation Mode]
Initializing...
Loading properties from
/opt/IBM/tivoli/tip/profiles/TIPProfile/etc/tnm/tnm.properties
ConnectionPool 'READ' Initialised
JDBC Driver: com.ibm.db2.jcc.DB2Driver
JDBC URL   : jdbc:db2://tnmip:50000/NCIM
Working on domain 'ITS0'...
Processing 14 valid device(s)
% Complete: 0...10...20...30...40...50...60...70...80...90...100
Writing IDML Book to
'/opt/IBM/tivoli/netcool/var/precision/ccmdb/ITNMIP.9.42.171.29.2009
-05-08T19.04.18.682Z.refresh.xml'...
Shutting down...
Finished.
```

Import the IDML book into IBM Tivoli Application Dependency Discovery Manager:

1. From the IBM Tivoli Network Manager for IP system, copy the IDML book file to the IBM Tivoli Application Dependency Discovery Manager system.
2. Run the IBM Tivoli Application Dependency Discovery Manager bulk loader to import the IDML book. From the `$COLLATION_HOME/bin` directory, issue `./loadidml.sh -f <filename> -u <user> -p <password>`. Make sure that it gets a 0 return code.
3. If you have the IBM Tivoli Business Service Manager integration, you can now extract new objects that are loaded in IBM Tivoli Application Dependency Discovery Manager. You can force a refresh for IBM Tivoli Business Service Manager by issuing the `./cmdbdiscovery -r` command from the `$TBSM_HOME/XMLtoolkit/bin` directory.

Note: If loading IBM Tivoli Network Manager for IP data into IBM Tivoli Business Service Manager directly, copy the IBM Tivoli Network Manager for IP IDML book file to the IBM Tivoli Business Service Manager system in the \$TBSM_HOME/discovery/dlbooks directory. Ensure that the Discovery Library Toolkit is configured to not connect to IBM Tivoli Application Dependency Discovery Manager with the property DL_TADDM_Connect=false specified in the xmltoolkitsvc.properties file.

Verify that new objects are available in IBM Tivoli Application Dependency Discovery Manager from IBM Tivoli Network Manager for IP.

Automatically implementing IBM Tivoli Directory Integrator

The following steps show how to implement exporting objects into IBM Tivoli Application Dependency Discovery Manager by using IBM Tivoli Directory Integrator:

1. Download the required package and install IBM Tivoli Directory Integrator:
 - a. IBM Tivoli Directory Integrator V6.1.1 was previously installed with Fix Pack 6. This integration requires IBM Tivoli Directory Integrator V6.1.1 with Fix Pack 1 or later. For information about installing IBM Tivoli Directory Integrator, see the *IBM Tivoli Directory Integrator Administration Guide*, SC32-1716, or go to this Web site:
<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDI.doc/adminguide.htm>
 - b. From the IBM Open Process Automation Library Web site, download the Discovery Library Integration Framework package:
<http://www.ibm.com/software/tivoli/opal/details?catalog.label=1TW10CC1U>
 - c. From the IBM Open Process Automation Library Web site, download the Discovery Library Integration Framework plug-in for IBM Tivoli Network Manager for IP package:
<http://www.ibm.com/software/tivoli/opal/details?catalog.label=1TW10CC1Y>
2. Install the Discovery Library Integration Framework:
 - a. On the IBM Tivoli Directory Integrator server, copy the Discovery Library Integration Framework DLAtoCCMDB directory to the \$TDI_SOLUTION_DIR/DLAtoCCMDB directory.
 - b. Verify the Discovery Library Integration Framework version by viewing the \$TDI_SOLUTION_DIR/DLAtoCCMDB/DLAtoCCMDB.xml file. Find the line <MetamergeConfig line, IDIversion="TDI6.1.1-DICommon1.2".

- c. Install the Discovery Library Integration Framework plug-in for IBM Tivoli Network Manager for IP. Unzip the plug-in in the \$TDI_SOLUTION_DIR/DLAtoCCMDB directory. Verify that the tdi_ccmdb_itnm_plugin12.jar file exists in the \$TDI_SOLUTION_DIR/DLAtoCCMDB/TMS directory.
3. Configure the Discovery Library Integration Framework:
 - a. Launch the IBM Tivoli Directory Integrator configuration editor. Open the \$TDI_SOLUTION_DIR/DLAtoCCMDB/DLAtoCCMDB.xml file.
 - b. Right-click the “Includes” directory, and select **Include** → **New Include**. Enter the name ITNMDLA, and click **OK**.
 - c. Select ibmdiXML as the Config Driver. Under file, enter the full path to the ITNMDLA.xml file. The password is only required if the XML is password-protected; leave this field blank for the OPAL package.
 - d. Save the selected configuration, and exit the IBM Tivoli Directory Integrator configuration editor.
 - e. Edit the \$TDI_SOLUTION_DIR/DLAtoCCMDB/DLAtoCCMDB.properties file.
Edit the EnabledALs property. If IBM Tivoli Network Manager for IP is the only Discovery Library Integration Framework plug-in, change the contents to ITNMDLA:/AssemblyLines/ITNMtoCCMDB. If there are other Discovery Library Integration Framework plug-ins installed, append the ITNMDLA configuration to the end of the property. Example 5-9 illustrates the EnabledALs property.

Example 5-9 EnabledALs

```
#Enabled plug-ins
EnabledALs=ITNMDLA:/AssemblyLines/ITNMtoCCMDB
```

Remove invalid characters <93> and <94> from the header comments. IBM Tivoli Directory Integrator will not be able to start with these invalid characters. Example 5-10 illustrates the invalid characters <93> and <94>.

Example 5-10 Remove invalid characters

```
#Licensed Materials - Property of IBM <93>Restricted Materials of
IBM<94>
```

- f. Create a Java wrapper script. Create \$TDI_SOLUTION_DIR/DLAtoCCMDB/calljava with the contents, as shown in Example 5-11 on page 171. Edit the executable location of Java Runtime Environment (JRE™) 1.5 and the full path to the TIPProfile for your environment.

Example 5-11 The calljava wrapper script

```
#!/bin/sh
/opt/IBM/tdi/jvm/jre/bin/java
-Duser.install.root=/opt/IBM/tivoli/tip/profiles/TIPProfile $*
```

- g. Edit the \$TDI_SOLUTION_DIR/DLAtoCCMDB/DLAtoCCMDB.properties file. Add the following properties:

itnmHost	Host name of IBM Tivoli Network Manager for IP server
itnmDLADataSinkDir	This directory must match the ncp.dla.datasink.targetDirectory directory in the \$NCHOME/precision/adapters/ncp_dla/ncp_dla.properties file. This directory must be empty every time that the assembly line executes.
itnmRemoteUser	System user with access to the itnmDLADataSinkDir
itnmRemotePassword	Password for the system user. To encrypt, prefix the property as {protect}-itnmRemotePassword. The password will be encrypted when the Discovery Library Integration Framework process is started.
itnmConnectionType	Protocol to connect to the itnmHost. The options are ANY, SSH, RSH, REXEC, and WIN.
itnmJavaCommand	The full path to the java wrapper script that was created in previous step.
itnmDLAJarName	The file name of the ncp_dla jar file in the \$NCHOME/precision/adapters/ncp_dla/ directory.
itnmDLAProperties	The file name of the ncp_dla properties file in the \$NCHOME/precision/adapters/ncp_dla/ directory.
itnmDLAPath	The path to the ncp_dla directory.

Example 5-12 on page 171 illustrates the additional properties that have been added to the DLAtoCCMDB.properties file.

Example 5-12 Added properties in the DLAtoCCMDB.properties file

```
#ITNM properties
itnmHost=tnmip.itso.ral.ibm.com
```

```

itnmDLADataSinkDir=/opt/IBM/tivoli/netcool/var/precision/ccmdb
itnmRemoteUser=netcool
{protect}-itnmRemotePassword=itso4you
itnmConnectionType=SSH
itnmJavaCommand=/opt/IBM/tdi/DLatoCCMDB/calljava
itnmDLAJarName=ncp_DLA.jar
itnmDLAProperties=ncp_dla.properties
itnmDLAPath=/opt/IBM/tivoli/netcool/precision/adapters/ncp_dla

```

- h. Edit the \$TDI_SOLUTION_DIR/DLatoCCMDB/DLatoCCMDB.properties file. Edit the following properties:

ccmdbHost	Host name of IBM Tivoli Application Dependency Discovery Manager system. This name is not used unless isTdiOnCcmdb is set to false.
isTdiOnCcmdb	This property must be set to false when IBM Tivoli Directory Integrator is on a separate system.
ccmdbStagingDir	The full path to the remote staging directory on the IBM Tivoli Application Dependency Discovery Manager server. This path must be created for this integration. Nothing else can exist in this directory, because the ccmdbUsername will delete the contents.
ccmdbUsername	System user for connecting to the IBM Tivoli Application Dependency Discovery Manager system and invoking the bulk loader. This user must be the user running IBM Tivoli Application Dependency Discovery Manager.
{protect}-ccmdbPassword	The System password for ccmdbUsername. This password will be encrypted when IBM Tivoli Directory Integrator is started.
ccmdbComProtocol	Protocol to connect to the ccmdbHost. The options are ANY, SSH, RSH, REXEC, and WIN.
ccmdbProcessedFilesDir	The directory on the IBM Tivoli Application Dependency Discovery Manager server that contains the processedfiles.list file.
bulkLoaderPath	The full path to the loadidml executable.
useBulkLoadGraphWritingAlg	This property specifies whether the -g option will be used in the bulk load command. If you do not

	want to use the -g option in the bulk import command, enter <code>false</code> .
logLevel	IBM Tivoli Directory Integrator IBM Tivoli Application Dependency Discovery Manager integration log level. The options are <code>DEBUG</code> , <code>INFO</code> , <code>WARN</code> , <code>ERROR</code> , or <code>FATAL</code> .
logFilePath	The log file path relative to the <code>\$TDI_HOME</code> directory.
tdiStagingDir	The staging directory for IBM Tivoli Directory Integrator to store IDML files relative to the <code>\$TDI_HOME</code> directory.
tdiBackupDir	The backup directory for IDML files relative to the <code>\$TDI_HOME</code> directory.
scheduleMonth	The month to run the bulk import. January-December or enter <code>*</code> (asterisk) for every month.
scheduleDay	The day of the month to run the integration. 1-31 or <code>*</code> is every day.
scheduleWeekday	The day of the week to run the integration. Sunday-Monday or <code>*</code> is every weekday.
scheduleHour	The hour to run the bulk import. 0-24 or <code>*</code> is every hour.
scheduleMinute	The minute to run the bulk import.

Optional notification properties

These properties can be configured to send an e-mail when the import fails or succeeds. The `notificationEvents` property must be set to blank to disable notification.

Example 5-13 illustrates the properties that were edited in the `DLAtoCCMDB.properties` file.

Example 5-13 Edited properties in the `DLAtoCCMDB.properties` file

```
#CCMDB server details
ccmdbHost=taddm.itso.ral.ibm.com
ccmdbStagingDir=/opt/IBM/cmdb/dist/bulk/ITNMDLstaging
ccmdbUsername=cmdbadmin
{protect}-ccmdbPassword=itso4you
ccmdbComProtocol=SSH
ccmdbProcessedFilesDir=/opt/IBM/cmdb/dist/bulk
bulkLoaderPath=/opt/IBM/cmdb/dist/bin/loadidml.sh
useBulkLoadGraphWritingAlg=false
```

```
isTdiOnCcldb=false

#TDI details
logLevel=INFO
logFilePath=DLAtocCmdb/log/DLAtocCmdb.log
tdiStagingDir=DLAtocCmdb/staging
tdiBackupDir=DLAtocCmdb/backup

#Scheduling configuration
scheduleMonth=*
scheduleDay=*
scheduleWeekday=Sunday
scheduleHour=3
scheduleMinute=0
```

- i. On the IBM Tivoli Network Manager for IP system, edit the `$NCHOME/precision/adapters/ncp_dla/ncp_dla.properties` file. Ensure the `ncp.dla.precisionDomain` property is set to the IBM Tivoli Network Manager for IP domain from which to export objects.
- j. Start IBM Tivoli Directory Integrator. To run the IBM Tivoli Network Manager for IP export to IBM Tivoli Application Dependency Discovery Manager on a scheduled basis as defined in the `DLAtocCmdb.properties` file, start IBM Tivoli Directory Integrator:

```
./ibmdisrv -c DLAtocCmdb/DLAtocCmdb.xml -r "Schedule Discoveries" &
```

To run a single IBM Tivoli Network Manager for IP export to IBM Tivoli Application Dependency Discovery Manager now and shut down, start IBM Tivoli Directory Integrator:

```
./ibmdisrv -c DLAtocCmdb/DLAtocCmdb.xml -r "Perform Discoveries"
```

4. Verification

To verify a successful import, view the `$TDI_SOLUTION_DIR/DLAtocCmdb/log/DLAtocCmdb.log` file. Look for a message similar to Example 5-14 on page 174 that indicates a successful bulk load.

Example 5-14 DLAtocCmdb.log file

```
2009-05-15 13:28:51,916 INFO
[AssemblyLine.AssemblyLines/ToCmdb.789982998] - CTJDI0047I The
processedfiles.list file indicates that the following file was
successfully loaded by the TADD Bulk Loader:
ITNMIP.9.42.171.29.2009-05-15T17.28.35.377Z.refresh.xml
```

Verify that the new objects are available in IBM Tivoli Application Dependency Discovery Manager from IBM Tivoli Network Manager for IP.

Manually importing from IBM Tivoli Application Dependency Discovery Manager

Export ComputerSystem data from IBM Tivoli Application Dependency Discovery Manager:

1. On the IBM Tivoli Network Manager for IP system, copy the `$NCHOME/precision/adapters/ncp_dla/integration/itnm_guids.xml` file to the IBM Tivoli Application Dependency Discovery Manager system in the `$COLLATION_HOME/sdk/bin` directory.
2. On the IBM Tivoli Application Dependency Discovery Manager system, issue the `$COLLATION_HOME/sdk/bin/api.sh -u <username> -p <password> find ComputerSystem > itnm_guids.xml` command.
3. Extract the entityIds and GUIDs to a CSV file. Ensure both the `itnm_guids.xml` file and the `itnm_guids.xml` file exist in the `$COLLATION_HOME/sdk/bin` directory. Issue the `$COLLATION_HOME/sdk/bin/xslt.sh -XSL ./itnm_guids.xml > itnm_guids.csv` command.

Import ComputerSystem data to IBM Tivoli Network Manager for IP:

1. Copy the `itnm_guids.csv` file from the IBM Tivoli Application Dependency Discovery Manager system to the IBM Tivoli Network Manager for IP system in the `$NCHOME/precision/adapters/ncp_dla` directory.
2. From the `$NCHOME/precision/adapters/ncp_dla` directory, issue the `./ncp_dla.sh -import -file itnm_guids.csv ncp_dla.properties` command. Example 5-15 illustrates the ComputerSystem data import into IBM Tivoli Network Manager for IP.

Example 5-15 Import from IBM Tivoli Application Dependency Discovery Manager

```
[root@tnmip ncp_dla]# ./ncp_dla.sh -import -file itnm_guids.csv
ncp_dla.properties
ncp_DLA (IBM Tivoli Network Manager IP Edition - Discovery Library
Adapter)
Copyright (C) 1997 - 2008 By IBM Corporation. All Rights Reserved.
See product license for details.
```

```
[GUID Import Mode]
Initializing...
Loading properties from
/opt/IBM/tivoli/tip/profiles/TIPProfile/etc/tnm/tnm.properties
ConnectionPool 'READ' Initialised
```

```
JDBC Driver: com.ibm.db2.jcc.DB2Driver
JDBC URL   : jdbc:db2://tnmip:50000/NCIM
Importing GUIDs from 'itnm_guids.csv'...
Imported 14 GUID(s) into NCIM.
Shutting down...
Finished.
```

Verification

Perform the Launch in Context from IBM Tivoli Network Manager for IP to IBM Tivoli Application Dependency Discovery Manager to verify a successful import.

5.3 Event data integration

In this section, we discuss event data as the integration point. *Event data*, as opposed to resource data, represents a volatile alert that can flow from one application to another application to indicate that operator intervention might be required. Figure 5-31 on page 176 shows the event data integration flow that we discuss.

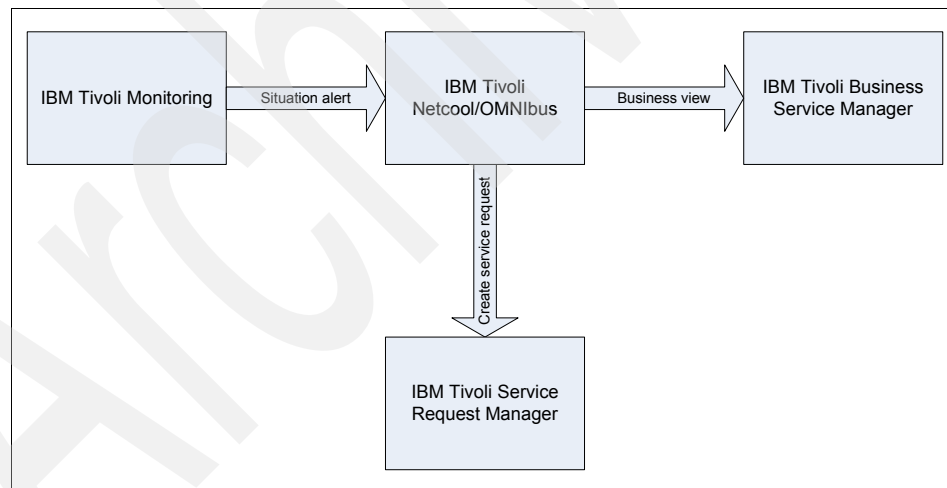


Figure 5-31 Event data integration

The event data integration starts from IBM Tivoli Monitoring situations as monitoring alerts. These alerts are sent to IBM Tivoli Netcool/OMNIBus as the primary event processor; see 5.3.1, “IBM Tivoli Netcool/OMNIBus and IBM Tivoli Monitoring” on page 177.

IBM Tivoli Netcool/OMNIBus events then can be passed on to other applications for further processing, such as:

- ▶ IBM Tivoli Business Service Manager for showing business services, which is a default behavior, so no customization is needed
- ▶ IBM Tivoli Service Request Manager for opening a service incident (refer to 5.3.2, “IBM Tivoli Netcool/OMNIBus and IBM Tivoli Service Request Manager integration” on page 194)

5.3.1 IBM Tivoli Netcool/OMNIBus and IBM Tivoli Monitoring

In this section, we describe the requirements for data integration between IBM Tivoli Monitoring and IBM Tivoli Netcool/OMNIBus. The IBM Tivoli Netcool/OMNIBus and IBM Tivoli Monitoring data integration allows IBM Tivoli Netcool/OMNIBus events to create events in IBM Tivoli Monitoring. The Integration is bidirectional and allows IBM Tivoli Monitoring to send Resolution and Closure event information to IBM Tivoli Netcool/OMNIBus.

Also, IBM Tivoli Monitoring can be configured to create events in IBM Tivoli Netcool/OMNIBus:

- ▶ IBM Tivoli Netcool/OMNIBus to IBM Tivoli Monitoring
This configuration represents event synchronization when an IBM Tivoli Monitoring event that is forwarded to IBM Tivoli Netcool/OMNIBus is modified.
- ▶ IBM Tivoli Monitoring to IBM Tivoli Netcool/OMNIBus
IBM Tivoli Netcool/OMNIBus is the recommended event processing platform. The data integration between IBM Tivoli Monitoring and IBM Tivoli Netcool/OMNIBus allows you to consolidate the event management in IBM Tivoli Netcool/OMNIBus and includes monitoring situations from IBM Tivoli Monitoring.
By default, all situation events are forwarded to the IBM Tivoli Netcool/OMNIBus event server. IBM Tivoli Monitoring allows you to specify situations that send situation events to IBM Tivoli Netcool/OMNIBus through the Event Information Facility tab of the Situation editor in Tivoli Enterprise Portal.

The IBM Tivoli Netcool/OMNIBus and IBM Tivoli Monitoring integration includes the following components:

- ▶ Situation Update Forwarder
Situation Update Forwarder sends information from IBM Tivoli Netcool/OMNIBus or IBM Tivoli Enterprise Console to IBM Tivoli Monitoring. Situation Update Forwarder Version 6.2.1 was used in this integration, because Situation Update Forwarder Version 6.2.2 is currently a Beta version.

- IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility
IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility receives event information from any application that can send events in Event Integration Facility format and sends the event to IBM Tivoli Netcool/OMNIBus. Currently, IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility requires the IBM Tivoli Netcool/OMNIBus Nonnative Base Probe and Java 1.5. IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility V7.0.5806 and IBM Tivoli Netcool/OMNIBus Nonnative Base Probe V4.0.5837 were used in this integration.

Figure 5-32 on page 178 shows the data flow and the components that were used by the IBM Tivoli Netcool/OMNIBus and IBM Tivoli Monitoring integration. For this integration, we install the Situation Update Forwarder and the IBM Tivoli Netcool/OMNIBus components on tbsm. IBM Tivoli Netcool/OMNIBus was previously installed on tbsm, and IBM Tivoli Monitoring was previously installed on itm. The platform for all installations was Red Hat Enterprise Linux 4.

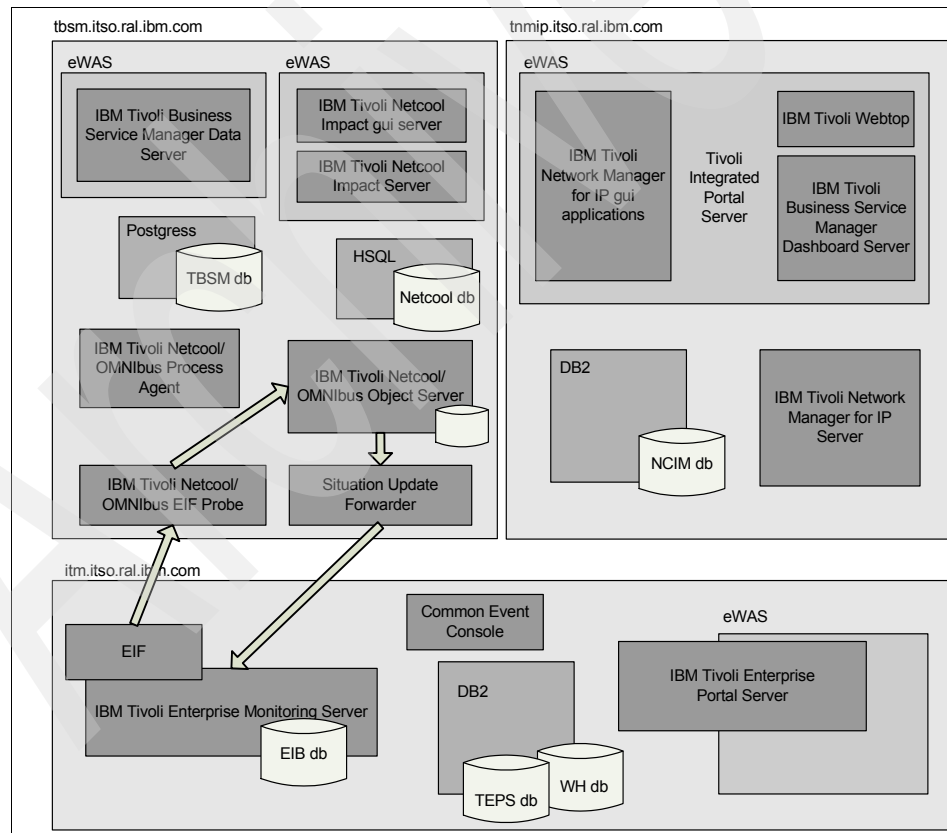


Figure 5-32 Event flow

These steps describe the outbound event flow between IBM Tivoli Monitoring and IBM Tivoli Netcool/OMNIBus:

1. Event insert, update, or close occurs in IBM Tivoli Monitoring.
2. Tivoli Enterprise Monitoring Server situation event forwarding matches the event and sends the event to IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility.
3. IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility reformats the received event and sends the event to IBM Tivoli Netcool/OMNIBus.

These steps describe the inbound event flow between IBM Tivoli Netcool/OMNIBus and IBM Tivoli Monitoring:

1. An event originating from IBM Tivoli Monitoring is acknowledged, closed, or reopened in IBM Tivoli Netcool/OMNIBus.
2. IBM Tivoli Netcool/OMNIBus automation sends the event to the Situation Update Forwarder.
3. Situation Update Forwarder sends the event information to IBM Tivoli Monitoring.
4. The event is acknowledged, closed, or reopened in IBM Tivoli Monitoring.

Table 5-1 illustrates the default severity mapping between IBM Tivoli Monitoring and IBM Tivoli Netcool/OMNIBus. This mapping is set in the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility rules file.

Table 5-1 Event severity mapping

IBM Tivoli Netcool/OMNIBus severity	IBM Tivoli Monitoring severity
Critical (Severity 5)	Fatal
Major (Severity 4)	Critical
Minor (Severity 3)	Minor
Warning (Severity 2)	Warning
Indeterminate (Severity 1)	Unknown or Informational or Harmless

The event message format can be modified by performing the Event Information Facility slot customization of the message slot. The slot can be customized to contain a literal text string or to include values of other slots from the event.

You can write rules, procedures, or triggers in IBM Tivoli Netcool/OMNIBus for situation events received from IBM Tivoli Monitoring. The configuration of the

event synchronization and Event Integration Facility on the event server can be changed, as well.

Configuring event synchronization

Perform the following steps to configure event synchronization from IBM Tivoli Netcool/OMNIbus to IBM Tivoli Monitoring:

1. Install Situation Update Forwarder:
 - a. On the IBM Tivoli Netcool/OMNIbus server as the netcool user, change the directory to the IBM Tivoli Monitoring 6.2.1 Tools install media\tec directory.
 - b. Export the NCHOME environment variable.
 - c. Issue the `./ESync2100<platform>.bin -console` command to start the text-based installation. Example 5-16 on page 180 illustrates running the console installation.

Example 5-16 Console installation

```
[netcool@tbsm tec]$ ./ESync2100Linux.bin -console
InstallShield Wizard
```

```
Initializing InstallShield Wizard...
```

```
Preparing Java(tm) Virtual Machine...
```

```
.....
.....
.....
```

```
Welcome to the InstallShield Wizard for IBM Tivoli Monitoring and
Tivoli Event Synchronization
The InstallShield Wizard will install IBM Tivoli Monitoring and Tivoli
Event Synchronization on your computer.
To continue, choose Next.
IBM Tivoli Monitoring and Tivoli Event Synchronization
Press 1 for Next, 3 to Cancel or 4 to Redisplay [1] 1
```

- d. Press 1 to start the installation.
- e. Press 1 to accept the license agreement, and press 1 to continue.
- f. Enter the directory name for the installation, and press 1 to continue. The default directory is the `/opt/IBM/SitForwarder` directory. Example 5-17 illustrates the directory name default selection.

Example 5-17 Install location

IBM Tivoli Monitoring and Tivoli Event Synchronization Install Location
Please specify a directory or press Enter to accept the default directory.

Directory Name: [/opt/IBM/SitForwarder]

Press 1 for Next, 2 for Previous, 3 to Cancel or 4 to Redisplay [1] 1

- g. Enter the name of the configuration file, the number of seconds to sleep when there are no new situation updates, the number of bytes to use to save the last event, the URL of the Tivoli Enterprise Monitoring Server SOAP server, the rate for sending SOAP requests to Tivoli Enterprise Monitoring Server, and the level of debug. Accept the default values if desired. Press 1 to continue. Example 5-18 on page 181 illustrates the default configuration selection.

Example 5-18 Configuration selection

Name of configuration file [situpdate.conf]

Number of seconds to sleep when no new situation updates [3]

Number of bytes to use to save last event [50]

URL of the TEMS SOAP server [cms/soap]

Rate for sending SOAP requests to TEMS from Event Sync via Web Services [10]

Level of debug detail for log

[X] 1 - low

[] 2 - med

[] 3 - verbose

To select an item enter its number, or 0 when you are finished: [0]

Press 1 for Next, 2 for Previous, 3 to Cancel or 4 to Redisplay [1] 1

- h. Enter the maximum size of a single cache file in bytes, the maximum number of cache files, and the directory in which the cache file will reside. Accept the default values if desired. Press 1 to continue. Example 5-19 illustrates the default cache configuration selection.

Example 5-19 Cache configuration

Maximum size of any single cache file, in bytes [50000]

Maximum number of cache files [10]

Directory for cache files to reside [/opt/IBM/SitForwarder/persistence]

Press 1 for Next, 2 for Previous, 3 to Cancel or 4 to Redisplay [1] 1

- i. Enter the host name, the user ID, and the password for the Tivoli Enterprise Monitoring Server. You will be prompted for this information 10 times in case there are multiple Tivoli Enterprise Monitoring Servers. If

there is only one Tivoli Enterprise Monitoring Server, enter blank information on the remaining lines. Press 1 to continue.

Example 5-20 illustrates a single Tivoli Enterprise Monitoring Server configuration. We enter the remaining Tivoli Enterprise Monitoring Server server lines as blanks.

Example 5-20 Tivoli Enterprise Monitoring Server configuration

```
--- Tivoli Enterprise Monitoring server 1 ---
Host name [] itm.itso.ral.ibm.com
User ID [] itmuser
Password:
Confirmation:
```

- j. Press 1 to begin the installation. Wait until the installation completes and press 3 to finish. Example 5-21 illustrates a successful IBM Tivoli Netcool/OMNIbus Probe for Tivoli Event Information Facility installation.

Example 5-21 Situation Update Forwarder installation

```
IBM Tivoli Monitoring and Tivoli Event Synchronization will be
installed in the following location:
/opt/IBM/SitForwarder
for a total size: 129.8 MB
Press 1 for Next, 2 for Previous, 3 to Cancel or 4 to Redisplay [1] 1
Installing $L(com.ibm.opms.strings.OPMS_Inst_Strings,PRODUCT_NAME_ES).
Please
wait...
|-----|-----|-----|-----|
0%      25%      50%      75%      100%
||||||||||||||||||||||||||||||||||||||||||||||||||||||||
-----
The InstallShield Wizard has successfully installed IBM Tivoli
Monitoring and Tivoli Event Synchronization. Choose Finish to exit the
wizard.
Press 3 to Finish or 4 to Redisplay [3] 3
```

2. Configure IBM Tivoli Netcool/OMNIbus to run external procedures:

If not already configured, IBM Tivoli Netcool/OMNIbus must be configured to run external procedures through Process Control. For details about configuring Process Control to run external procedures, see this Web site:

http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.netcool_OMNIbus.doc_7.2.1/admin/concept/omn_adm_pa_usingpctrltorunxtrnlprocs.html

3. Update the IBM Tivoli Netcool/OMNIBus database with the columns and triggers that are required for this integration:

- a. Change the directory to the \$OMNIHOME/bin directory.

- b. Issue the **nco_sql** command redirecting input from itm_proc.sql:

```
./nco_sql -user "username" -password "password" <
/opt/IBM/SitForwarder/omnibus/itm_proc.sql
```

Example 5-22 on page 183 illustrates the successful itm_proc.sql Object Server update.

Example 5-22 Running itm_proc.sql

```
[netcool@tbsm bin]$ ./nco_sql -user root -password '' <
/opt/IBM/SitForwarder/omnibus/itm_proc.sql
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
```

- c. Issue the **nco_sql** command redirecting input from the IBM Tivoli Monitoring Tools provided itm_db_update.sql:

```
./nco_sql -user "username" -password "password" <
/opt/IBM/SitForwarder/omnibus/itm_db_update.sql
```

Errors will be received for objects that already exist. Ignore these "Object Exists" and "Attempt to insert duplicate row" error messages.

Example 5-23 illustrates the successful itm_db_update.sql Object Server update.

Example 5-23 Running itm_db_update.sql

```
[netcool@tbsm bin]$ ./nco_sql -user root -password '' <
/opt/IBM/SitForwarder/omnibus/itm_db_update.sql
ERROR=Object exists on line 48 of statement
'-----...',
at or near 'TECHostname'
```

```

ERROR=Object exists on line 2 of statement 'alter table alerts.status
add column TECFQHostname varchar(64);...', at or near 'TECFQHostname'
ERROR=Object exists on line 2 of statement 'alter table alerts.status
add column TECDate varchar(64);...', at or near 'TECDate'
ERROR=Object exists on line 2 of statement 'alter table alerts.status
add column TECRepeatCount int;...', at or near 'TECRepeatCount'
ERROR=Object exists on line 7 of statement
'-----...',
at or near 'ITMStatus'
ERROR=Object exists on line 2 of statement 'alter table alerts.status
add column ITMDisplayItem varchar(128);...', at or near
'ITMDisplayItem'
ERROR=Object exists on line 2 of statement 'alter table alerts.status
add column ITMEventData varchar(3096);...', at or near 'ITMEventData'
ERROR=Object exists on line 2 of statement 'alter table alerts.status
add column ITMTime varchar(23);...', at or near 'ITMTime'
ERROR=Object exists on line 2 of statement 'alter table alerts.status
add column ITMHostname varchar(64);...', at or near 'ITMHostname'
(0 rows affected)
ERROR=Object exists on line 3 of statement 'alter table alerts.status
add column ITMIntType varchar(1);...', at or near 'ITMIntType'
ERROR=Object exists on line 2 of statement 'alter table alerts.status
add column ITMResetFlag varchar(1);...', at or near 'ITMResetFlag'
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
ERROR=Attempt to insert duplicate row on line 8 of statement
'-----...'
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)

```

- d. Issue the command redirecting input from the IBM Tivoli Monitoring Tools provided itm_sync.sql:

```

./nco_sql -user "username" -password "password" <
/opt/IBM/SitForwarder/omnibus/itm_sync.sql

```

Example 5-24 illustrates the successful itm_sync.sql Object Server update.

Example 5-24 Running itm_sync.sql

```

[netcool@tbsm bin]$ ./nco_sql -user root -password '' <
/opt/IBM/SitForwarder/omnibus/itm_sync.sql

```

(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)

- e. Restart the Object Server.
4. Configure IBM Tivoli Netcool/OMNIBus external procedures to run as the non-root user. By default, the created IBM Tivoli Netcool/OMNIBus external procedures run as the root user. If Process Control is not running as root, the external procedure fails, because a non-root user cannot run a command as a root user. If the Process Agent is running as a non-root user, complete the following steps to modify the external procedure:
 - a. Open the IBM Tivoli Netcool/OMNIBus Administrator.
 - b. Go to **Automation** → **Procedures**.
 - c. Double-click the **eventcmd** external procedure.
 - d. Modify the User ID and Group ID to the non-root system user ID and group ID. Click **OK**. Figure 5-33 illustrates the modified external procedure.

Edit Procedure (NCOMS on tbsm.itso.ral.ibm.com:4100)

External Procedure Details

Name:

Parameters

```
in send_event_cmd Char(1000)
in send_event_args Char(8192)
```

In/Out: Name: Data Type: Array: ☐

Executable:

Arguments:

Host:

User ID: Group ID:

Figure 5-33 External procedure with non-root user

5. Configure IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility:

- Copy the `tivoli_eif.rules` file in the `/opt/IBM/SitForwarder/omnibus` directory to the `$OMNIHOME/probes/<arch>/itm_tivoli_eif.rules` file.
- Copy the `$OMNIHOME/probes/<arch>/tivoli_eif.props` file to the `itm_tivoli_eif.props` file, updating the following properties:

PortNumber	The port number to receive events from IBM Tivoli Monitoring. Nothing else can use this port.
Inactivity	The time in seconds that the probe allows a port to be inactive before disconnecting. Consider the idle time between IBM Tivoli Monitoring events.
EIFCacheFile	The full path to the probe cache file.
Server	Object Server name.

When running multiple instances of the probe, all probe files specified must be unique. Be sure to update the EIFCacheFile file, MessageLog file, PidFile file, PropsFile file, RulesFile file, and SAFFileName file to a unique value with the full path. Example 5-25 illustrates the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility property settings.

Example 5-25 The itm_tivoli_eif.props settings

```
EIFCacheFile : '/opt/IBM/tivoli/netcool/omnibus/var/itm_tivoli_eif_probe.cache'
Inactivity : 0
Manager : 'itm_tivoli_eif'
MessageLog : '/opt/IBM/tivoli/netcool/omnibus/log/itm_tivoli_eif.log'
PidFile : '/opt/IBM/tivoli/netcool/omnibus/var/itm_tivoli_eif'
PortNumber : 5539
PropsFile : '/opt/IBM/tivoli/netcool/omnibus/probes/linux2x86/itm_tivoli_eif.props'
RulesFile : '/opt/IBM/tivoli/netcool/omnibus/probes/linux2x86/itm_tivoli_eif.rules'
SAFFileName : '/opt/IBM/tivoli/netcool/omnibus/var/itm_tivoli_eif.store'
```

6. Configure Situation Update Forwarder events:

Situation Update Forwarder can be configured to send events directly to IBM Tivoli Netcool/OMNIBus when it encounters problems. To configure this functionality, update the /opt/IBM/SitForwarder/omnibus/errorevent.conf file with the ServerName and ServerPort of the Object Server. Example 5-26 illustrates the configured errorevent.conf file.

Example 5-26 errorevent.conf

```
ServerName=tbsm.itso.ral.ibm.com
ServerPort=5539
BufferEvents=YES
BufEvtPath=/tmp/itm_sync_error.cache
BufEvtMaxSize=4096
```

7. Start Situation Update Forwarder:

On the IBM Tivoli Netcool/OMNIBus server, from the /opt/IBM/SitForwarder/bin directory, issue **./startSUF.sh &**. Example 5-27 illustrates starting Situation Update Forwarder.

Example 5-27 Situation Update Forwarder start-up

```
[root@tbsm bin]# ./startSUF.sh &
[1] 24424
nohup: appending output to `nohup.out'
[1]+  Done                  ./startSUF.sh
```

Verify the Situation Update Forwarder process is running. Example 5-28 illustrates that the Situation Update Forwarder process is running.

Example 5-28 Situation Update Forwarder process

```
[root@tbsm bin]# ps -ef|grep SituationUp
root      24425      1  7 16:02 pts/9    00:00:00 ../jre/bin/java
-Djlog.logCmdPort=0 -Dsuf.config.path=/opt/IBM/SitForwarder/etc -Xrs
-cp ../jars/situpdate.jar:../jars/jlog.jar
com.tivoli.candlenet.SituationUpdateForwarder start OMNIBUS
root      24470 24059  0 16:02 pts/9    00:00:00 grep SituationUp
```

8. Verify that the IBM Tivoli Monitoring and IBM Tivoli Netcool/OMNIBus integration is working as expected by performing the following tests:
 - Insert an event into IBM Tivoli Monitoring. Verify an event is created in IBM Tivoli Netcool/OMNIBus according to Table 5-1 on page 179.
 - Acknowledge, close, or open an event in IBM Tivoli Netcool/OMNIBus. Verify that the event is updated in IBM Tivoli Monitoring.
 - Acknowledge, close, or open the event in IBM Tivoli Monitoring. Verify that the event is updated in IBM Tivoli Netcool/OMNIBus.

The default IBM Tivoli Netcool/OMNIBus and IBM Tivoli Monitoring event severity mapping is based on the severity name and not the severity color. If the IBM Tivoli Monitoring operators will observe the IBM Tivoli Netcool/OMNIBus console, the severity mapping is usually changed to be based on color. Table 5-2 on page 188 lists the severity and its color.

Table 5-2 New event severity mapping

IBM Tivoli Netcool/OMNIBus severity	IBM Tivoli Monitoring severity
Critical (Severity 5) - Red	Fatal - Black
Critical (Severity 5) - Red	Critical - Red
Major (Severity 4) - Orange	Minor - Orange
Minor (Severity 3) - Yellow	Warning - Yellow
Warning (Severity 2) - Blue	Unknown - Gray or Informational - Blue
Clear (Severity 0) - Green	Harmless - Green

The severity mapping is stored in the `$OMNIHOME/probes/<arch>/itm_tivoli_eif.rules` file. Edit the severity lines and change the severity as desired. Example 5-29 illustrates the new event severity mapping as defined in Table 5-2.

```
if( exists( $severity ) )
{
    switch ( $severity )
    {
        case "FATAL" :
            @Severity = 5
        case "60":
            @Severity = 5
        case "CRITICAL":
            @Severity = 5
        case "50":
            @Severity = 5
        case "MINOR":
            @Severity = 4
        case "40":
            @Severity = 4
        case "WARNING":
            @Severity = 3
        case "30":
            @Severity = 3
        default:
            @Severity = 2
    }
}
```

When you change this severity mapping, you must restart the probe process to load the rules file.

Configuring event forwarding to IBM Tivoli Netcool/OMNibus

Follow these steps to forward events to IBM Tivoli Netcool/OMNibus:

1. Configure the Hub Tivoli Enterprise Monitoring Server to forward situation events to IBM Tivoli Netcool/OMNibus:
 - a. Configure the Hub Tivoli Enterprise Monitoring Server by using the **\$CANDLEHOME/bin/itmcmd config -S -t HUB_itm** command, which starts the configuration process.
 - b. Accept the default configured values for most of the steps as shown in Example 5-30.

Example 5-30 Configuring Tivoli Enterprise Monitoring Server: Part 1

Configuring TEMS...

Hub or Remote [1=*LOCAL, 2=*REMOTE] (Default is: 1):

TEMS Host Name (Default is: itm.itso.ral.ibm.com):
Network Protocol 1 [ip, sna, ip.pipe or ip.spipe] (Default is:
ip.pipe):

Now choose the next protocol number from one of these:

- ip
- sna
- ip.spipe
- 0 for none

Network Protocol 2 (Default is: 0):
IP.PIPE Port Number (Default is: 1918):
Enter name of KDC_PARTITION (Default is: null):
Enter path and name of KDC_PARTITIONFILE (Default is:
/opt/IBM/ITM/tables/HUB_itm/partition.txt):
Configuration Auditing? [1=YES, 2=NO] (Default is: 1):
Standby TEMS Site or type 0 for "none" (Default is: 0):
Enter Optional Primary Network Name or type 0 for "none" : (Default is:
0):
Security: Validate User ? [1=YES, 2=NO] (Default is: 1):
LDAP Security: Validate User with LDAP ? [1=YES, 2=NO] (Default is: 2):

- c. Change the EIF option to YES by selecting **1** and pressing Enter in the next prompt (see Example 5-31 on page 190). Then, type the following values (we install the EIF probe in the same host as IBM Tivoli Netcool/OMNIBus):

- IP address or host name of the EIF probe and press Enter
- EIF port for the EIF probe process and press Enter

Example 5-31 EIF settings

Tivoli Event Integration Facility? [1=YES, 2=NO] (Default is: 1):
EIF Server? (Default is: tbsm.itso.ral.ibm.com):
EIF Port? (Default is: 5539):

- d. Accept the default values for the rest of the configuration items.

Note: Unless you have already added user authority, you might want to give QueryAdd and UpdateAdd user access rights to the itmuser user and to the sysadmin user.

2. Configure the Common Event Console:

- a. You can use the Manage Tivoli Enterprise Monitoring Services by issuing the command `/opt/IBM/ITM/bin/itmcmd manage`.

- b. In the Manage Enterprise Tivoli Monitoring Services window, right-click **Tivoli Enterprise Portal Server** and select **Configure**.
- c. Click **New**, select the **IBM Tivoli Netcool/OMNibus Connector** tab, and fill the fields to configure the connector. Table 5-3 shows the connector settings that we use for our environment.

Table 5-3 Connector settings

Field	Value	Comment
Connector name	itso_omnibus	The name to display in the common event console for this connector
Maximum number of events for this connector	100	The maximum number of events that are available in the common event console for this connector
Computer name of event system	tbsm.itso.ral.ibm.com	The computer name of the event system that is associated with this connector
Port number of event system	4100	The port number that is used by the connector to retrieve events from IBM Tivoli Netcool/OMNibus event system
User name for accessing event system	root	The user name when accessing the event system that is associated with this connector
Password		The password that is associated with the user name
SQL WHERE clause	ITMStatus = "	Restricts events for the common event console. If you do not define a clause, all IBM Tivoli Netcool/OMNibus events are available in the common event console. We use a clause that restricts IBM Tivoli Netcool/OMNibus events in the common event console to only those events that do not originate as Tivoli Monitoring events, which helps us to retrieve only one copy of the same event, avoiding duplicate event information in the common event console.
View cleared events	Yes	A value of Yes means that cleared events for this connector are available in the common event console.
Time interval for polling event system	1	The number of minutes between each poll of the event system for new or changed events. IBM Tivoli Netcool/OMNibus Object Server automatically sends new or changed events to the common event console as they become available. Therefore, the primary purpose of this checking is to ensure that the server and the connection to the server function properly.

Field	Value	Comment
Time interval between reconnection attempts	20	The number of seconds of delay between reconnection attempts when the connector loses its connection to the event system
Number of reconnection attempts	10	The maximum number of consecutive reconnection attempts to make if the connector loses its connection to the event system. If this value is set to 0 and the connector loses its connection, the connector remains inoperable indefinitely. If this value is set to -1 and the connector loses its connection, the connector attempts to reconnect indefinitely.
Information for extra table columns		Field type and field name that identify the field to map to each of the extra table columns
Names of Extra Columns tab		Information to map to each of the extra table columns

3. Install IBM Tivoli Netcool/OMNIbus Probe for Tivoli Event Information Facility. On the IBM Tivoli Netcool/OMNIbus server, install the IBM Tivoli Netcool/OMNIbus Gateway for Tivoli Event Information Facility:
 - a. Change the directory to the \$OMNIHOME/install directory.
 - b. As the IBM Tivoli Netcool/OMNIbus installation user, issue the `./nco_patch -install <patch to nco-g-tivoli-eif package>` command.

Example 5-32 illustrates the successful installation of the IBM Tivoli Netcool/OMNIbus Probe for Tivoli Event Information Facility.

Example 5-32 Gateway installation

```
[netcool@tbsm install]$ ./nco_patch -install
/opt/code/tsrmtoolkit/omnibus-3.6-linux2x86-gateway-nco-g-tivoli-eif-2_
0
Installing Patch
"/opt/code/tsrmtoolkit/omnibus-3.6-linux2x86-gateway-nco-g-tivoli-eif-2
_0" ...

----- End of README -----
Are you sure you want to install this patch? (y/n)? [default: y] y
Patch "gateway-nco-g-tivoli-eif-2" is successfully installed.
```

4. On the IBM Tivoli Netcool/OMNIBus installation, install the IBM Tivoli Netcool/OMNIBus Nonnative Base Probe and IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility. If IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility is already installed, multiple instances of the probe can run from a single installation:
 - a. Change the directory to the \$OMNIHOME/install directory.
 - b. The IBM Tivoli Netcool/OMNIBus Nonnative Base Probe is a prerequisite for the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility. As the IBM Tivoli Netcool/OMNIBus installation user, issue the **`./nco_patch -install <patch to nco-p-nonnative-base package>`** command.

Example 5-33 illustrates the successful installation of IBM Tivoli Netcool/OMNIBus Nonnative Base Probe.

Example 5-33 Nonnative base probe installation

```
[netcool@tbsm install]$ ./nco_patch -install
/opt/code/tsrmtoolkit/omnibus-linux2x86-probe-nonnative-base-4_0
Installing Patch
"/opt/code/tsrmtoolkit/omnibus-linux2x86-probe-nonnative-base-4_0" ...

----- End of README -----
Are you sure you want to install this patch? (y/n)? [default: y] y
Patch "probe-nonnative-base-4" is successfully installed.
```

- c. As the IBM Tivoli Netcool/OMNIBus installation user, issue the **`./nco_patch -install <patch to nco-p-tivoli-eif package>`** command.

Example 5-34 on page 193 illustrates the successful installation of IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility.

Example 5-34 Event Information Facility probe installation

```
[netcool@tbsm install]$ ./nco_patch -install
/opt/code/tsrmtoolkit/omnibus-linux2x86-probe-nco-p-tivoli-eif-7_0
Installing Patch
"/opt/code/tsrmtoolkit/omnibus-linux2x86-probe-nco-p-tivoli-eif-7_0 "
...

----- End of README -----
Are you sure you want to install this patch? (y/n)? [default: y] y
Patch "probe-nco-p-tivoli-eif-7" is successfully installed.
```

If IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility is already installed, multiple instances of the probe can run from a single

installation. IBM Tivoli Monitoring events sent to IBM Tivoli Netcool/OMNIBus can be accessed by using the IBM Tivoli Netcool/OMNIBus **nco_event** command or by going to the Tivoli Integrated Portal main menu and clicking **Availability → Events**.

5.3.2 IBM Tivoli Netcool/OMNIBus and IBM Tivoli Service Request Manager integration

The IBM Tivoli Netcool/OMNIBus and IBM Tivoli Service Request Manager data integration allows IBM Tivoli Netcool/OMNIBus events to open tickets or service requests in IBM Tivoli Service Request Manager. The Integration is bidirectional and allows IBM Tivoli Service Request Manager to send Resolution and Closure event information to IBM Tivoli Netcool/OMNIBus.

The IBM Tivoli Netcool/OMNIBus and IBM Tivoli Service Request Manager integration includes the following components:

- ▶ IBM Tivoli Directory Integrator

IBM Tivoli Directory Integrator is a stand-alone application that exchanges data between applications or data sources. IBM Tivoli Directory Integrator reads information from one or multiple data sources, reformats it in the specified format, and writes the information to one or multiple target applications. The IBM Tivoli Directory Integrator rules that reformat the data are assembly lines. The integration currently supports IBM Tivoli Directory Integrator Version 6.1.1. The IBM Tivoli Service Request Manager Version 7.1 integration will install IBM Tivoli Directory Integrator Version 6.1.1 using the IBM Tivoli Service Request Manager Integration Toolkit. If you have IBM Tivoli Service Request Manager Version 6.2, manually install IBM Tivoli Directory Integrator 6.1.1 using IBM Tivoli Netcool/OMNIBus Integration with IBM Tivoli Service Request Manager product documentation.

- ▶ IBM Tivoli Service Request Manager Integration Toolkit

The IBM Tivoli Service Request Manager Integration Toolkit provides IBM Tivoli Directory Integrator Version 6.1.1 and the configuration files that are necessary for IBM Tivoli Service Request Manager and IBM Tivoli Directory Integrator data integration. The IBM Tivoli Service Request Manager Integration Toolkit is only provided for IBM Tivoli Service Request Manager Version 7.1. For more information about integration with IBM Tivoli Service Request Manager Version 6.2, see IBM Tivoli Netcool/OMNIBus Integration with IBM Tivoli Service Request Manager product documentation.

- ▶ IBM Tivoli Netcool/OMNIBus Gateway Integration Module for IBM Tivoli Service Request Manager

The IBM Tivoli Netcool/OMNibus Gateway Integration Module for IBM Tivoli Service Request Manager provides the configuration files for IBM Tivoli Netcool/OMNibus, IBM Tivoli Netcool/OMNibus Probe for Tivoli Event Information Facility, IBM Tivoli Netcool/OMNibus Gateway for Tivoli Event Information Facility, and IBM Tivoli Directory Integrator in this integration. We use IBM Tivoli Netcool/OMNibus Gateway Integration Module for IBM Tivoli Service Request Manager Version 2.0.5869 in this integration.

Figure 5-34 on page 195 shows the data flow and components that are used by the IBM Tivoli Netcool/OMNibus and IBM Tivoli Service Request Manager integration. For this integration, we install the IBM Tivoli Directory Integrator on the tnmp machine and install the IBM Tivoli Netcool/OMNibus components on the tbsm machine. IBM Tivoli Netcool/OMNibus was installed on tbsm, and IBM Tivoli Service Request Manager was installed on ccmbd.

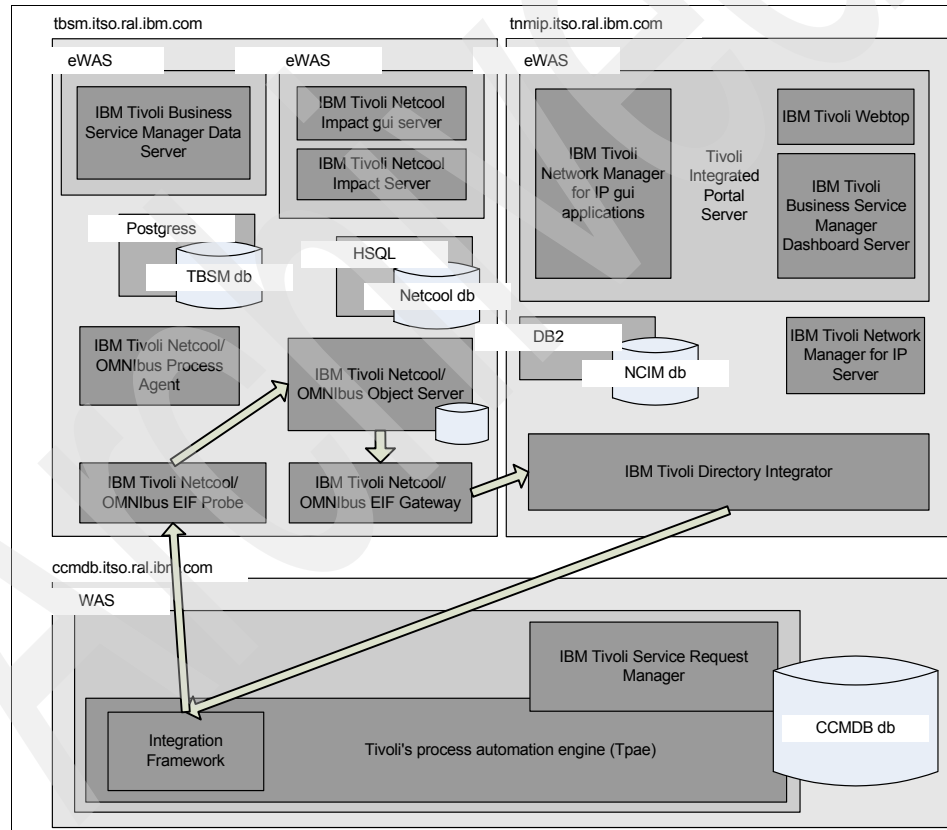


Figure 5-34 Data integration flow

These steps describe the outbound event flow between IBM Tivoli Netcool/OMNIBus and IBM Tivoli Service Request Manager:

1. The event is inserted in the IBM Tivoli Netcool/OMNIBus alerts.status table.
2. The IBM Tivoli Netcool/OMNIBus Gateway for Tivoli Event Information Facility filter matches the event and sends the event to IBM Tivoli Directory Integrator.
3. IBM Tivoli Directory Integrator reformats the received event and sends the event to IBM Tivoli Service Request Manager.
4. IBM Tivoli Service Request Manager creates a ticket or service request as specified by the event severity.

These steps describe the inbound event flow between IBM Tivoli Netcool/OMNIBus and IBM Tivoli Service Request Manager:

1. IBM Tivoli Service Request Manager ticket or service requests closure on an event originated by IBM Tivoli Netcool/OMNIBus.
2. IBM Tivoli Directory Integrator reformats the received event and sends the event to the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility.
3. IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility receives the event and sends the updated event severity to IBM Tivoli Netcool/OMNIBus.
4. IBM Tivoli Netcool/OMNIBus sets the event's severity, and the event is deleted by IBM Tivoli Netcool/OMNIBus automation.

Table 5-4 describes the event severity mapping between IBM Tivoli Netcool/OMNIBus and IBM Tivoli Service Request Manager.

Table 5-4 Event severity mapping

IBM Tivoli Netcool/OMNIBus severity	IBM Tivoli Service Request Manager incident or service request priority
Critical Severity (Severity=5)	Incident Priority 1
Major Severity (Severity=4)	Service Request Priority 1
Minor Severity (Severity=3)	Service Request Priority 2
Warning Severity (Severity=2)	Service Request Priority 3
Indeterminate Severity (Severity=1)	Service Request Priority 4
Clear Severity (Severity=0)	Service Request Priority 5

Note: When an event is deleted from IBM Tivoli Netcool/OMNIBus, the corresponding alert is set to Resolved in IBM Tivoli Service Request Manager.

When the severity of an event is updated in IBM Tivoli Netcool/OMNIBus, the priority of the corresponding alert is updated in IBM Tivoli Service Request Manager. However, the initial form of the IBM Tivoli Service Request Manager Incident or Service Request will not be changed; only the priority is changed.

When an event is Resolved or Closed in IBM Tivoli Service Request Manager, the severity of the corresponding event changes to Clear (Severity=0) in IBM Tivoli Netcool/OMNIBus.

IBM Tivoli Netcool/OMNIBus configuration

The integration is configured on both products. In this section, we discuss the IBM Tivoli Netcool/OMNIBus side of the implementation:

1. Install the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility, as discussed in “Configuring event forwarding to IBM Tivoli Netcool/OMNIBus” on page 189.
2. Install the IBM Tivoli Netcool/OMNIBus Gateway Integration Module for IBM Tivoli Service Request Manager:
 - a. Change the directory to the \$OMNIBUS_HOME/install directory.
 - b. As the IBM Tivoli Netcool/OMNIBus installation user, issue the `./nco_patch -force -install <path to nco-tdi-tsrn package>` command.

We use the `-force` option, because the software package incorrectly requires gateway-nco-g-tivoli-eif-1. gateway-nco-g-tivoli-eif-2 is installed, which supersedes gateway-g-tivoli-eif-1. Example 5-35 illustrates the successful installation of the IBM Tivoli Netcool/OMNIBus Gateway Integration Module for IBM Tivoli Service Request Manager.

Example 5-35 Module installation

```
[netcool@tbsm install]$ ./nco_patch -install
/opt/code/tsrntoolkit/omnibus-3.6-linux2x86-integration-nco-tdi-tsrn-2_
0
Installing Patch
"/opt/code/tsrntoolkit/omnibus-3.6-linux2x86-integration-nco-tdi-tsrn-2_
0" ...
```

```
----- End of README -----
Are you sure you want to install this patch? (y/n)? [default: y]
```

```
ERROR : Cannot install patch "integration-nco-tdi-tsrn-2" because of
the following reasons(s)
ERROR :      * Requires patch "gateway-nco-g-tivoli-eif-1" to be
installed
ERROR :      Contact IBM Support for patch "gateway-nco-g-tivoli-eif-1"
... Continuing installation/removal as the force flag is set!
Patch "integration-nco-tdi-tsrn-2" is successfully installed.
```

3. Update the IBM Tivoli Netcool/OMNIBus database with the columns and triggers that are required for this integration:

- a. Change the directory to the \$OMNIBUS/bin directory.
- b. Issue the **nco_sql** command to redirect input from the IBM Tivoli Netcool/OMNIBus Gateway Integration Module for IBM Tivoli Service Request Manager provided **tsrm_eif.sql**:

```
./nco_sql -user "username" -password "password" <
$OMNIBUS/gates/tdi_tsrn/os/tsrm_eif.sql
```

Example 5-36 illustrates the successful Object Server update.

Example 5-36 Running tsrm_eif.sql

```
[netcool@tbsm bin]$ ./nco_sql -user root -password '' <
/opt/IBM/tivoli/netcool/omnibus/gates/tdi_tsrn/os/tsrm_eif.sql
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
(0 rows affected)
```

- c. Configure IBM Tivoli Netcool/OMNIBus Gateway for Tivoli Event Information Facility to send events to IBM Tivoli Directory Integrator:
 - a. Copy the files in the \$OMNIBUS/gates/tdi_tsrn/gateway/ directory to the \$OMNIBUS/gates/tivoli_eif/ directory.
 - b. Copy the \$OMNIBUS/gates/tdi_tsrn/gateway/NCI_GATE.props to \$OMNIBUS/etc properties file.
 - c. Modify the \$NCHOME/etc/omni.dat file with the host name and port information for NCI_GATE. The port must be an unused port on the system. Example 5-37 illustrates a modified \$NCHOME/etc/omni.dat file for NCI_GATE.

Example 5-37 The omni.dat file

```
[NCO_GATE]
{
    Primary: tbsm.itso.ral.ibm.com 4300
}
```

- d. Issue the **\$NCHOME/bin/nco_igen** command to generate the interface files from the updated omni.dat file.
- e. Run the **\$OMNIHOME/bin/nco_g_crypt** command to encrypt the Object Server gateway user password. The password must be encrypted in the IBM Tivoli Netcool/OMNIBus Gateway for Tivoli Event Information Facility properties file. Example 5-38 on page 199 illustrates encrypting the Object Server gateway user password with the **nco_g_crypt** command.

Example 5-38 Encrypting the password with the nco_g_crypt command

```
[netcool@tbsm bin]$ ./nco_g_crypt
Password:
ZZ
```

- f. Edit the IBM Tivoli Netcool/OMNIBus Gateway for Tivoli Event Information Facility **\$OMNIHOME/etc/NCO_GATE.props** properties file to ensure that the following properties are already set. If they are not, ensure that you have copied the correct **NCO_GATE.props** file from the **\$OMNIHOME/gates/tdi_tsrm/gateway/** directory. These properties are correct:
 - Gate.TIVOLI{EIF.InsertClass: 'OMNIBus_Insert'
 - Gate.TIVOLI{EIF.DeleteClass: 'OMNIBus_Delete'
 - Gate.TIVOLI{EIF.Resync: FALSE
 - Gate.TIVOLI{EIF.MinimumSeverity: 0
 - Gate.RDRWTR.Password: Object Server gateway user password encrypted with **nco_g_crypt**
 - Gate.RDRWTR.Server: Object Server name
 - Gate.RDRWTR.Username: Object Server gateway user name
 - Gate.TIVOLI{EIF.ConfigFile: Full path to the **tivoli_eif_config** file

Note: The Gate.TIVOLI{EIF.ConfigFile property does not expand any operating system variables. Be sure to replace **\$OMNIHOME** with the full path.

Example 5-39 illustrates the updated property setting in the NCO_GATE.props file.

Example 5-39 NCO_GATE.props file

```
Gate.TIVOLI{EIF.InsertClass           : 'OMNIBUS_Insert'
Gate.TIVOLI{EIF.DeleteClass          : 'OMNIBUS_Delete'
Gate.TIVOLI{EIF.Resync               : FALSE
Gate.TIVOLI{EIF.MinimumSeverity      : 0
Gate.RDRWTR.Password                 : 'ZZ'
Gate.RDRWTR.Server                   : 'NCOMS'
Gate.RDRWTR.Username                 : 'root'
Gate.TIVOLI{EIF.ConfigFile           :
'/opt/IBM/tivoli/netcool/omnibus/gates/tivoli_eif/tivoli_eif_config'
```

g. Edit the \$OMNIHOME/gates/tivoli_eif/tivoli_eif_config file to update the following entries:

- c1ServerLocation: Host name of the IBM Tivoli Directory Integrator server
- c1Port: Port number to communicate with IBM Tivoli Directory Integrator
- BufEvtPath: Full path to the buffer file

Note: The c1Port in tivoli_eif_config must be an unused port and match the eif.recv.port that is configured later in the tsrm.properties file on the IBM Tivoli Directory Integrator server.

Example 5-40 illustrates the tivoli_eif_config file.

Example 5-40 The tivoli_eif_config file

```
TransportList=t1
t1Type=SOCKET
t1Channels=c1
c1ServerLocation=tnmip.itso.ral.ibm.com
c1Port=5529
BufferEvents=YES
BufEvtPath=/opt/IBM/tivoli/netcool/omnibus/var/tivoli_eif_gateway.cache
```

4. Configure IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility to receive events from IBM Tivoli Directory Integrator:
 - a. Copy the tivoli_eif.rules file and the tivoli_eif.props file in the \$OMNIHOME/gates/tdi_tsrm/probe/ directory to

`$OMNIHOME/probes/<arch>/tsrm_tivoli_eif.rules` file and the `tsrm_tivoli_eif.props` file.

- b. Edit the `$OMNIHOME/probes/<arch>/tivoli_eif.props` file to update the following properties:

- **PortNumber:** Port number to receive events from IBM Tivoli Directory Integrator

Note: The `PortNumber` in the `tsrm_tivoli_eif.props` file must be an unused port and must match the `eif.sent.port` that is configured later in the `tsrm.properties` file on the IBM Tivoli Directory Integrator server.

- **Inactivity:** The time in seconds that the probe allows a port to be inactive before disconnecting.

Note: You must set the `Inactivity` timeout to a value larger than the longest expected interval between receiving IBM Tivoli Service Request Manager Resolution or Closure updates. When reached, the probe will shut down and be automatically restarted if running under the Process Agent. If the probe is not running under the Process Agent, set the value to 0 and the probe will never time out.

- **EIFCache File:** Full path to the probe cache file
- **Server:** Object Server name
- **PropsFile:** Full path to probe properties file
- **RulesFile:** Full path to probe rules file

When running multiple instances of the probe, all probe files that are specified must be unique. Be sure to update the `EIFCacheFile`, `MessageLog`, `PidFile`, `PropsFile`, `RulesFile`, and `SAFFilename` files to a unique value with the full path. Example 5-41 illustrates the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility property settings.

Example 5-41 The tsrm_tivoli_eif.props file

```
EIFCacheFile : '/opt/IBM/tivoli/netcool/omnibus/var/tivoli_eif_probe.cache'  
Inactivity : 0  
PortNumber : 5530  
Server : 'NCOMS'  
PropsFile :  
'/opt/IBM/tivoli/netcool/omnibus/probes/linux2x86/tsrm_tivoli_eif.props'  
RulesFile :  
'/opt/IBM/tivoli/netcool/omnibus/probes/linux2x86/tsrm_tivoli_eif.rules'
```

Implementing IBM Tivoli Directory Integrator

Follow these steps:

1. IBM Tivoli Directory Integrator is a stand-alone component and can be installed on any server. Install the IBM Tivoli Directory Integrator:
 - a. Change the directory to the TDI/<arch> directory under the location of the untarred IBM Tivoli Service Request Manager Integration Toolkit package.
 - b. As the root user, run the following installation command:

```
./installTDI.sh -silent TDIServerDirectory TDIWorkingDirectory  
\\\\"ServiceRequestManagerHostNames\\\\" nonTMELogfileAdapter  
QueuingEnabled
```

Use the following values:

- TDIServerDirectory: Installation home for IBM Tivoli Directory Integrator
 - TDIWorkingDirectory: Working directory for IBM Tivoli Directory Integrator. Typically, this directory is a subdirectory of the TDIServerDirectory.
 - ServiceRequestManagerHostNames: The host name and HTTP listener port number of the IBM Tivoli Service Request Manager in the format *hostname:port*
 - nonTMELogfileAdapter: The full directory path for the nonTMELogfileAdapter to use with Tivoli Enterprise Console. Enter N/A for not applicable, because this value is not used in this implementation.
 - QueuingEnabled: Specified to use a common queue in a multiple IBM Tivoli Directory Integrator server environment. The default is Y.
2. Install the latest IBM Tivoli Directory Integrator fix pack as the root user. For this integration, we install 6.1.1-TIV-TDI-FP0006:
 - a. Issue the `/usr/ibm/common/ci/gmi/bin/gmi` command to launch the Update Installer tool GUI.

- b. Select **Next**.
- c. Select **Install maintenance packages such as fixes, fix packs or refresh packs**. Select **Next**. Figure 5-35 on page 203 illustrates the Install maintenance selection.

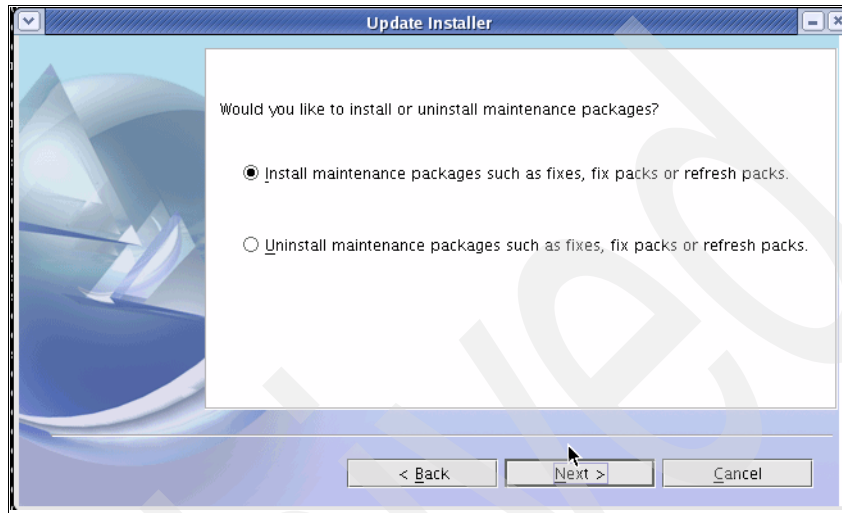


Figure 5-35 Install maintenance

- d. Select **IBM Tivoli Directory Integrator v6.1.1** as the product to be updated. Select **Next**. Figure 5-36 illustrates selecting IBM Tivoli Directory Integrator v6.1.1 as the product update.

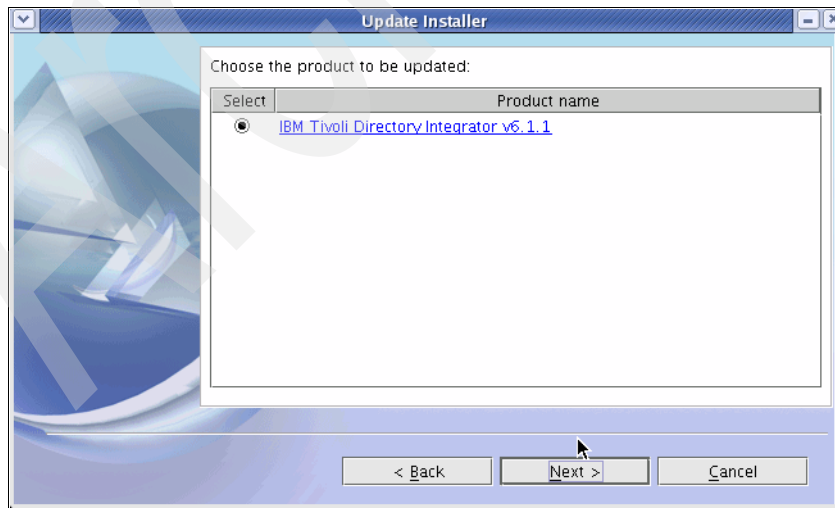


Figure 5-36 Product update

- e. Select **Edit** to search for the installation package. Locate the 6.1.1-TIV-TDI-FP0006 Fix Pack directory, and select **Add** to add it to the search paths. Select **OK**. Figure 5-37 illustrates the Fix Pack directory selection.

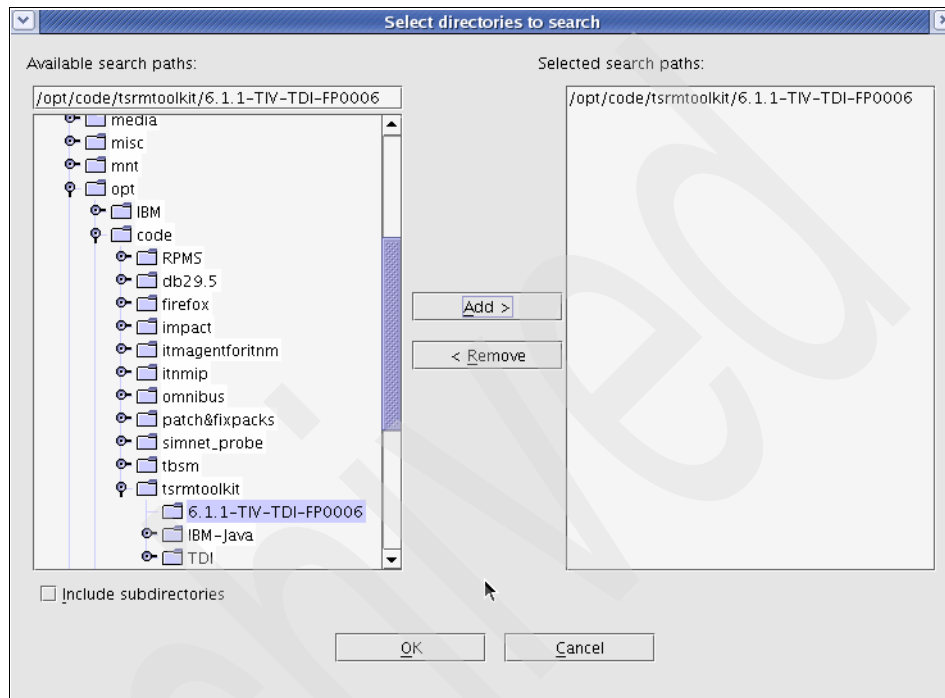


Figure 5-37 Directory to search

- f. Ensure that the fix pack directory path is correct, and select **Next**.
- g. Select **IBM Tivoli Directory Integrator v6.1.1 - FP0006** as the package to install, and select **Next**. Figure 5-38 on page 205 illustrates the fix pack selection.

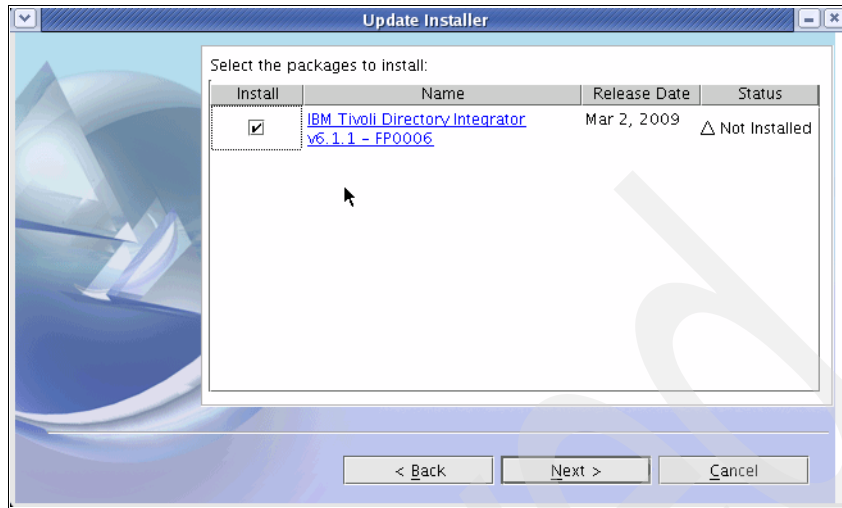


Figure 5-38 Fix pack to install

- h. Select **Install maintenance on this computer**, and select **Next**. Figure 5-39 illustrates the selection to install maintenance.

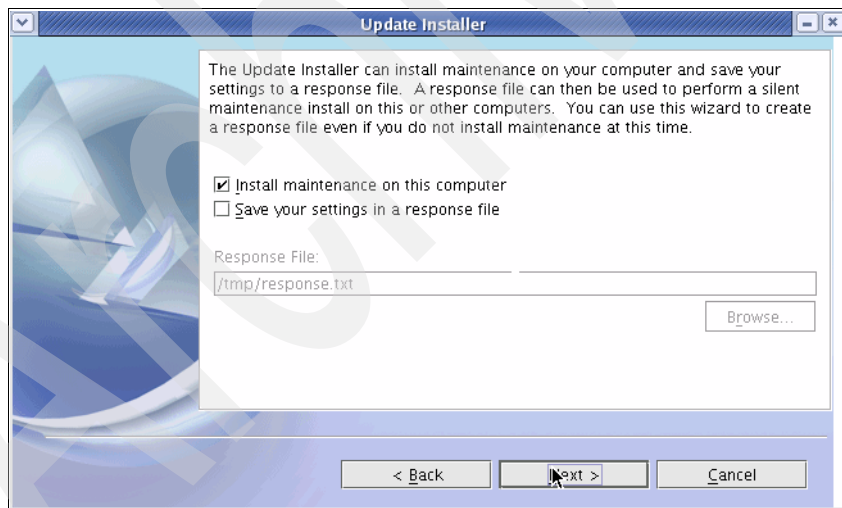


Figure 5-39 Install maintenance

- i. Verify the installation details, and select **Install**. Figure 5-40 on page 206 illustrates the pre-installation details.

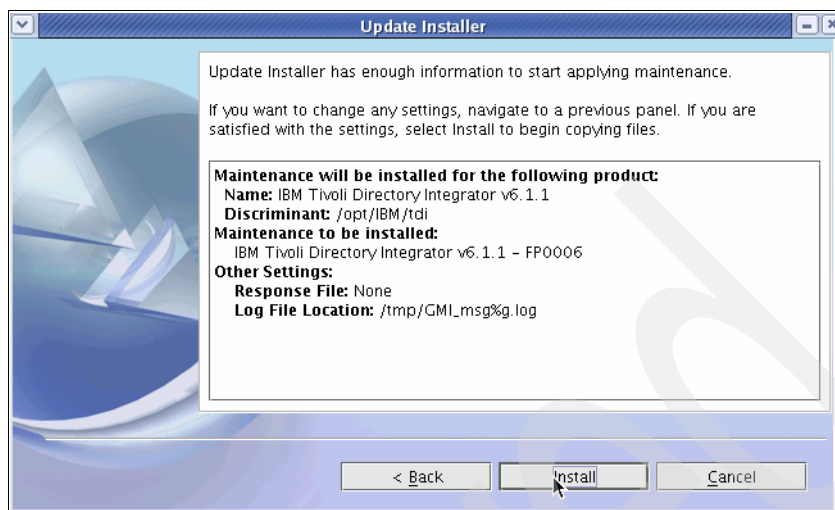


Figure 5-40 Pre-installation details

- j. Verify that the installation completes successfully. Figure 5-41 illustrates a successful fix pack installation.

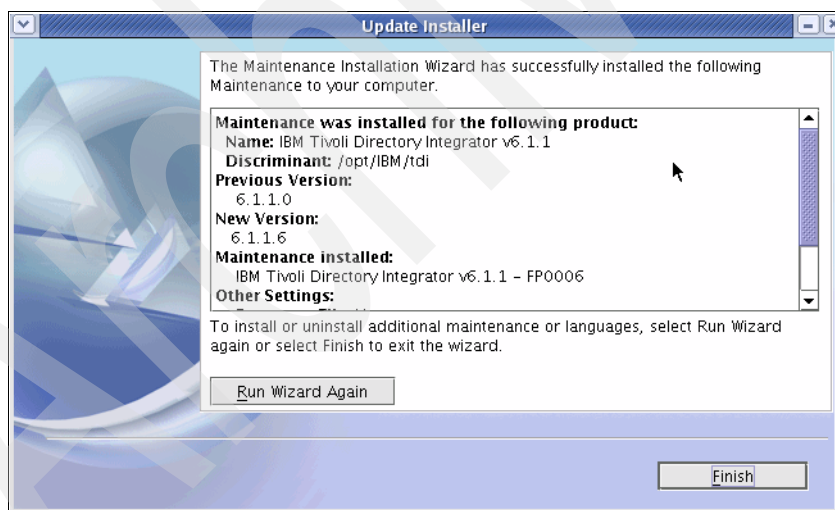


Figure 5-41 Successful installation

3. Install the latest IBM Tivoli Service Request Manager Connector from the currently installed IBM Tivoli Service Request Manager Fix Pack. In this integration, we install the latest IBM Tivoli Service Request Manager Connector from 7.1.0.3-TIV-SRM-FP0003.

- a. As the root user, rename the `$TDI_HOME/jars/connectors/generic-maximo-connector-1.0.0.jar` file to the `generic-maximo-connector-1.0.0.jar.old` file.
 - b. Change the directory to `tdi` in the 7.1.0.3-TIV-SRM-FP0003 package.
 - c. Copy the `generic-maximo-connector-1.0.1.jar` file to the `$TDI_HOME/jars/connectors` directory.
4. Configure IBM Tivoli Directory Integrator. From the IBM Tivoli Netcool/OMNIBus server, copy the required IBM Tivoli Directory Integrator configuration files to the IBM Tivoli Directory Integrator server:
 - a. From the IBM Tivoli Netcool/OMNIBus server in the `$OMNIHOME/gates/tdi_tsrn/tdi` directory, copy the `tsrm.properties` file and the `tsrm.xml` file to the IBM Tivoli Directory Integrator server in the `$TDI_HOME` directory.
 - b. On the IBM Tivoli Directory Integrator server, edit the `$TDI_HOME/tsrm.properties` file to remove any `^M` characters. In vi, this can be done with the `:%s/control-V control-M//g` command.
 - c. On the IBM Tivoli Directory Integrator server, edit the `$TDI_HOME/tsrm.properties` file, setting the following properties for IBM Tivoli Service Request Manager Version 7.1:
 - `default.maximo.url`: The root URL of the IBM Tivoli Service Request Manager Web services, for example, `http://servername:port`.
 - `default.maximo.user`: IBM Tivoli Service Request Manager user.
 - `default.maximo.password`: IBM Tivoli Service Request Manager user's password.
 - `EIF.recv.port`: The port of the IBM Tivoli Netcool/OMNIBus Gateway for Tivoli Event Information Facility, which is the `c1Port` port in the `tivoli_eif_config` file.
 - `EIF.send.port`: The port of the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility, which is the `PortNumber` in the `tivoli_eif.props` file.
 - `EIF.send.host`: The host name of the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility.

Example 5-42 illustrates the configured `tsrm.properties` file.

Example 5-42 The tsrm.properties file

```
default.maximo.url=http://ccmdb
default.maximo.user=maxadmin
default.maximo.password=maxadmin
default.maximo.authentication.required=true
eif.recv.port=5529
eif.recv.cache.location=eif_input.cache
eif.recv.cache.size=1024
eif.send.port=5530
eif.send.cache.location=eif_output.cache
eif.send.cache.size=1024
eif.send.trace.on=false
eif.send.host=tbsm.itso.ral.ibm.com
default.maximo.clear.status=RESOLVED
default.log.toEIF.log=toEIF
default.log.toTSRM.log=toTSRM
default.log.level=WARN
```

- d. On the IBM Tivoli Directory Integrator server in the \$TDI_HOME/jars file, rename the evd.jar file to the evd.jar.old file and the myEIFConnector.jar file to the myEIFConnector.jar.old file.
- e. From the IBM Tivoli Netcool/OMNIBus server in the \$OMNIHOME/gates/tdi_tsrm/tdi directory, copy the evd.jar, log.jar, and myEIFConnector.jar files to the IBM Tivoli Directory Integrator server in the \$TDI_HOME/jars directory.

Implementing IBM Tivoli Service Request Manager

Follow these steps on the IBM Tivoli Service Request Manager side:

1. Configuring IBM Tivoli Service Request Manager consists of generating the XML Schema Document (XSD) for all the objects to be exported using the Web application interface. Follow these steps to generate the required XSDs:
 - a. Start the Web application interface for IBM Change and Configuration Management Database. Select **Go To** → **Integration** → **Object Structures**. See Figure 5-42 on page 209.

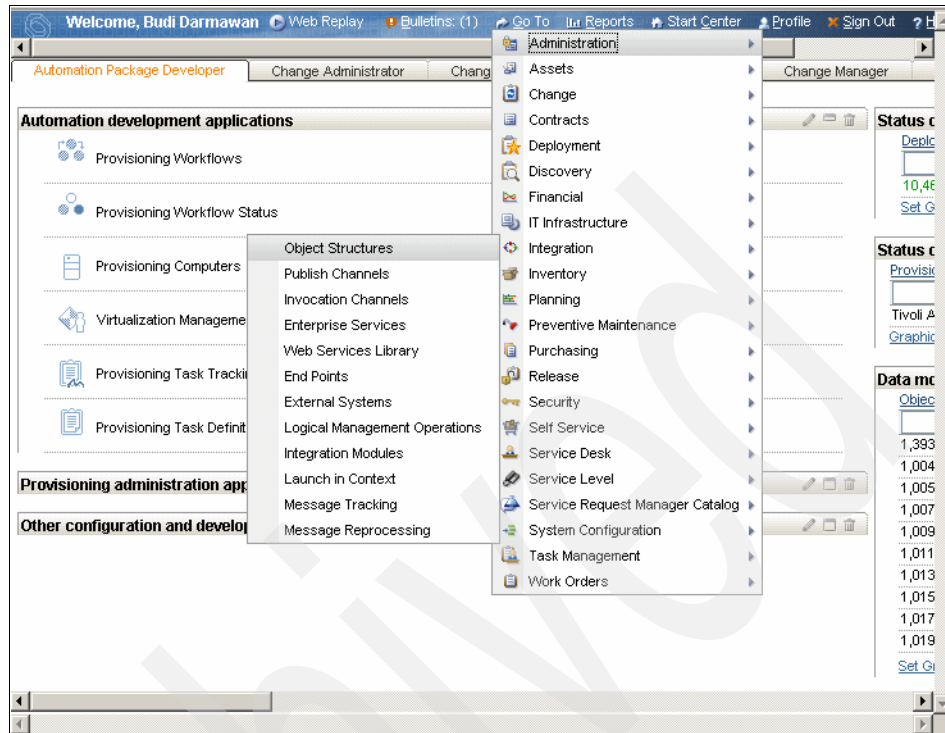


Figure 5-42 Launching Object Structures

- b. In the Object Structures filter field, enter MX0S and press Enter. As shown in Figure 5-43 on page 210, this action lists the six object structures for integrating with IBM Tivoli Service Request Manager using the Integration Framework.

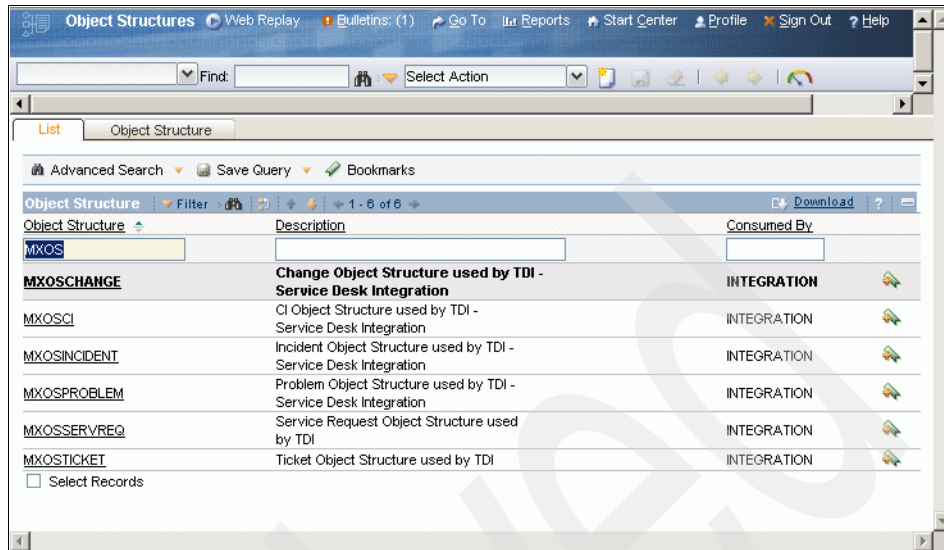


Figure 5-43 Filter MXOS Object Structures

- c. For each of the six object structures in the list, double-click the object structure to display the Object Structure details. From the **Select Action** pull-down, select **Generate Schema/View XML**, as shown in Figure 5-44.

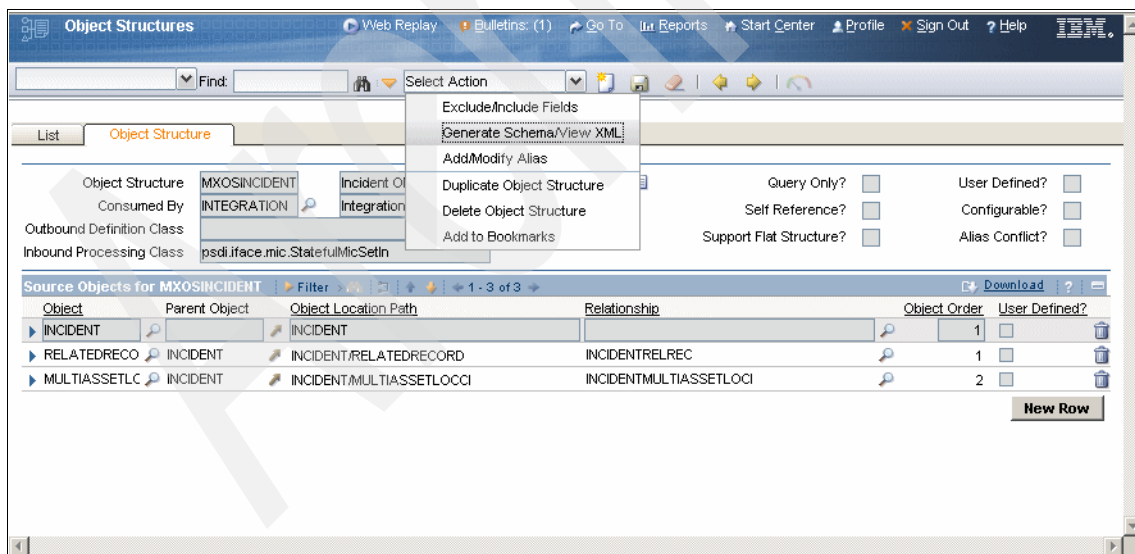


Figure 5-44 Generate Schema/View XML for Object Structure

- d. As shown in Figure 5-45, verify the BMXAA1245E message and click **OK** to continue.

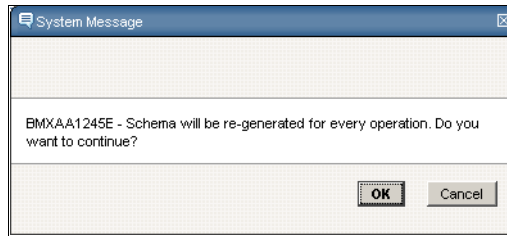


Figure 5-45 Regenerate

- e. The actions will generate the XML. The window in Figure 5-46 shows the XML that was generated.

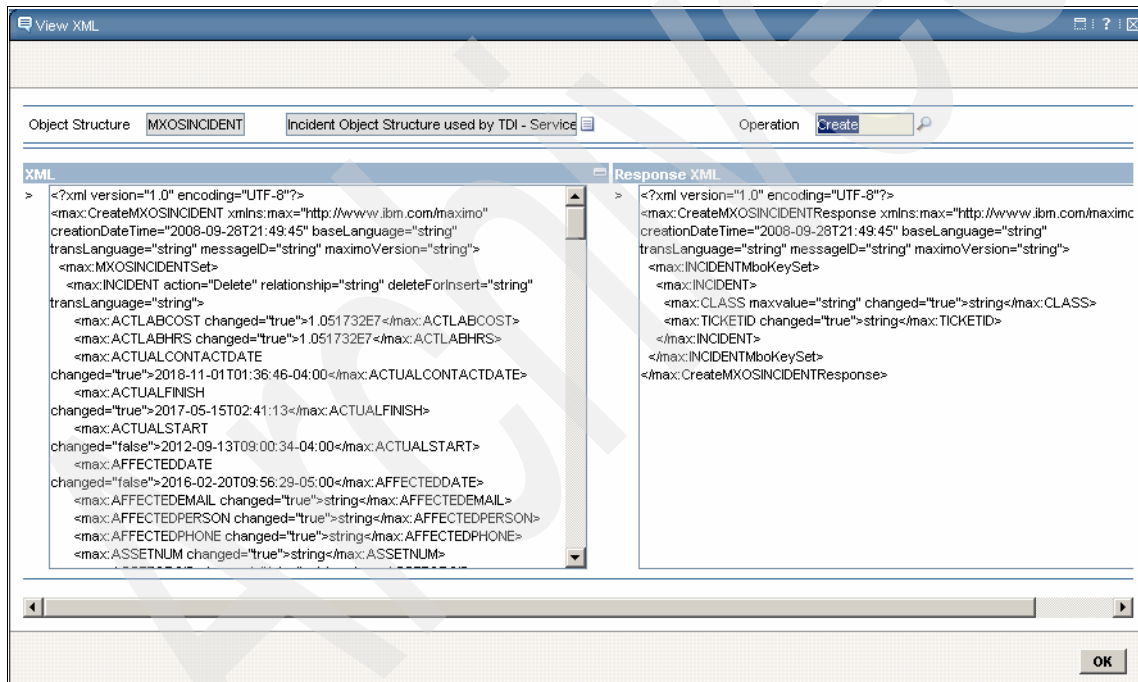


Figure 5-46 XML is displayed

Note: If you get the BMXAA5798E - Invalid integration web application url:http://localhost:7001/meaweb/ error, the Web application URL is incorrect. Update the Web application URL with the correct host name and port number by updating the mxe.int.webappurl file in **Go to → System Configuration → Platform Configuration → System Properties.**

- f. You must perform steps c on page 210 to step e on page 211 for all six object structures that are displayed in Figure 5-43 on page 210.
2. Start IBM Tivoli Directory Integrator:
 - a. From the IBM Tivoli Netcool/OMNIBus server, copy the `start_tdi.sh` script from the `$OMNIHOME/gates/tdi_tsrm/tdi` directory to the IBM Tivoli Directory Integrator server in the `$TDI_HOME` directory.
 - b. Ensure that the `tsrm.properties` and `tsrm.xml` files are also in `$TDI_HOME` directory on the IBM Tivoli Directory Integrator server.
 - c. On the IBM Tivoli Directory Integrator server, edit the `$TDI_HOME/start_tdi.sh` script, setting `TDIHOME` to the correct `$TDI_HOME` path.
 - d. From `$TDI_HOME` as the root user, issue the `./start_tdi.sh &` command.
 - e. Verify that the following process is running:

```
root      21439 21438  0 09:55 pts/1      00:00:00 /bin/sh
/opt/IBM/tdi//ibmdisrv -c tsrm.xml
```
3. Configure Process Control for probes and gateways:
 - a. Configure Process Control to start automatically at system start-up. On the IBM Tivoli Netcool/OMNIBus Object Server, in the `$OMNIHOME/install/startup` directory, issue the `./linux2x86install` command.
 - b. Enter `y` to continue the installation.
 - c. Enter the Process Agent name, and enter `y` to run in secure mode.
 - d. Enter blank for the Netcool License Server, which is not required in IBM Tivoli Netcool/OMNIBus Version 7.2.1. Example 5-43 on page 213 illustrates running the `linux2x86install` script to automatically start Process Control at system start-up.

Example 5-43 Running the linux2x86install script

```
[root@tbsm startup]# ./linux2x86install
This script copies a startup script into the /etc/init.d
directory to enable
you to automatically start and stop Netcool/OMNIBus processes.
```

It does this by:

```
Copying linux2x86/etc/rc.d/init.d/nco to /etc/init.d/nco
Running "/sbin/chkconfig --add nco"
```

```
Do you wish to continue (y/n)? [y] y
Name of the Process Agent Daemon [NCO_PA]:
Should NCO_PA run in secure mode (y/n)? [y]
Enter value for environment variable NETCOOL_LICENSE_FILE
if required [27000@localhost]:
Scripts installed.
```

- e. If running Process Control as a non-root user, edit the `/etc/init.d/nco` script to execute the `nco_pad` command as a non-root user. Example 5-44 illustrates executing `nco_pad` as a non-root user on Red Hat Linux.

Example 5-44 Executing the nco_pad start-up as non-root user

```
if [ "$SECURE" = "Y" ]; then
    daemon --user=netcool ${OMNIHOME}/bin/nco_pad -name ${NCO_PA}
    -authenticate PAM -secure > /dev/null 2> /dev/null
else
    daemon --user=netcool ${OMNIHOME}/bin/nco_pad -name ${NCO_PA}
    -authenticate PAM > /dev/null 2> /dev/null
fi
```

- f. As the root user, execute the `/etc/init.d/nco start` command to start Process Control.
4. Start the IBM Tivoli Netcool/OMNIBus Gateway for Tivoli Event Information Facility. You can start the IBM Tivoli Netcool/OMNIBus Gateway for Tivoli Event Information Facility from the command line for testing or from Process Control for automatic restart if necessary:
- To run IBM Tivoli Netcool/OMNIBus Gateway for Tivoli Event Information Facility from the command line, from the `$OMNIHOME/bin` directory as the netcool user, issue the `./nco_g_tivoli_eif &` command.

- To run IBM Tivoli Netcool/OMNIBus Gateway for Tivoli Event Information Facility from Process Control, perform the following steps:
 - i. Edit `$OMNIHOME/etc/nco_pa.conf`. Add a process definition for the IBM Tivoli Netcool/OMNIBus Gateway for Tivoli Event Information Facility. Example 5-45 illustrates the `nco_pa.conf` process configuration for the IBM Tivoli Netcool/OMNIBus Gateway for Tivoli Event Information Facility.

Example 5-45 IBM Tivoli Netcool/OMNIBus Gateway for Tivoli EIF process

```
nco_process 'nco_gw_eif_tsrn'
{
    Command '$OMNIHOME/bin/nco_g_tivoli_eif &' run as 501
    Host      =      'tbsm.itso.ral.ibm.com'
    Managed   =      True
    RestartMsg =      '${NAME} running as ${EUID}
has been restored on ${HOST}.'
    AlertMsg  =      '${NAME} running as ${EUID}
has died on ${HOST}.'
    RetryCount =      0
    ProcessType =      PaPA_AWARE
}
```

- ii. Edit `$OMNIHOME/etc/nco_pa.conf`. Add a process definition for the IBM Tivoli Netcool/OMNIBus Gateway for Tivoli Event Information Facility. Example 5-46 illustrates the `nco_pa.conf` service configuration for the IBM Tivoli Netcool/OMNIBus Gateway for Tivoli Event Information Facility with a dependency that the gateway starts after the Object Server is running.

Example 5-46 IBM Tivoli Netcool/OMNIBus Gateway for Tivoli EIF service

```
nco_service 'Core'
{
    ServiceType      =      Master
    ServiceStart     =      Auto
    process 'MasterObjectServer' NONE
    process 'nco_gw_eif_tsrn' 'MasterObjectServer'
}
```

- iii. You must restart the Process Agent for the new configuration to take effect. As the root user, issue the `/etc/init.d/nco restart` command. Example 5-47 illustrates restarting Process Control.

Example 5-47 Restart Process Control

```
[root@tbsm init.d]# /etc/init.d/nco restart
Netcool/OMNIBus : Stopping Process Control ...      [ OK
]
Netcool/OMNIBus : Starting Process Control ...      [ OK
]
```

- g. Issue the **\$OMNIHOME/bin/nco_pa_status** command to verify that the IBM Tivoli Netcool/OMNIBus Gateway for Tivoli Event Information Facility is running. Example 5-48 illustrates the IBM Tivoli Netcool/OMNIBus Gateway for Tivoli Event Information Facility running from Process Control.

Example 5-48 The nco_pa_status command

```
[netcool@tbsm bin]$ ./nco_pa_status
Login Password:
-----
Service Name      Process Name      Hostname  User      Status
PID
-----
Core              MasterObjectServer  tbsm.itso.ral.ibm.comnetcool
RUNNING          3354
nco_gw_eif_tsrm   tbsm.itso.ral.ibm.comnetcool
RUNNING          3594
-----
```

5. Start the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility. You can start the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility from a command line for testing or from Process Control for automatic restart if necessary. To run IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility from a command line, perform the following steps:
- Set the environment variable **PATH** to include Java 1.5:
`export PATH=$NCHOME/platform/linux2x86/jre_1.5.6/jre/bin:$PATH`
 - Set the environment variable **CLASSPATH** to include the `nco_p_tivoli_eif.jar` file:
`export CLASSPATH=$OMNIHOME/probes/java/nco_p_tivoli_eif.jar:$CLASSPATH`
 - As the `netcool` user, from the `$OMNIHOME/probes` directory, issue the `./nco_p_tivoli_eif &` command.

To run IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility from Process Control, perform the following steps:

- a. As the netcool user, edit the `$OMNIHOME/probes/java/nco_p_tivoli_eif.env` file. Add the `PATH` environment variable to include the Java 1.5 path, and add the `CLASSPATH` environment variable to include the `nco_p_tivoli_eif.jar`, `evd.jar`, `log.jar`, and `NSProbe.jar` files. Export the environment variables. Example 5-49 illustrates the `PATH` and `CLASSPATH` environment variable settings in the `$OMNIHOME/probes/java/nco_p_tivoli_eif.env` file.

Example 5-49 The `nco_p_tivoli_eif.env` file

```
PATH=/opt/IBM/tivoli/netcool/platform/linux2x86/jre_1.5.6/jre/bin:$PATH
CLASSPATH=/opt/IBM/tivoli/netcool/omnibus/probes/java/nco_p_tivoli_eif.
jar:/opt/IBM/tivoli/netcool/omnibus/probes/java/NSProbe.jar:/opt/IBM/ti
voli/netcool/omnibus/probes/java/evd.jar:/opt/IBM/tivoli/netcool/omnibu
s/probes/java/log.jar:$CLASSPATH

export PATH CLASSPATH
```

- b. Edit `$OMNIHOME/etc/nco_pa.conf`. Add a process definition for the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility. Example 5-50 illustrates the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility process definition.

Example 5-50 IBM Tivoli Netcool/OMNIBus Probe for Tivoli EIF process definition

```
nco_process 'nco_probe_eif_tsrm'
{
    Command '$OMNIHOME/probes/nco_p_tivoli_eif -propsfile
$OMNIHOME/probes/linux2x86/tivoli_eif.props &' run as 501
    Host = 'tbsm.itso.ra1.ibm.com'
    Managed = True
    RestartMsg = '${NAME} running as ${EUID} has been
restored on ${HOST}.'
    AlertMsg = '${NAME} running as ${EUID} has died on
${HOST}.'
    RetryCount = 0
    ProcessType = PaPA_AWARE
}
```

- c. Edit `$OMNIHOME/etc/nco_pa.conf`. Add a process definition for the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility. Example 5-51 illustrates the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility process definition.

Example 5-51 IBM Tivoli Netcool/OMNIBus Probe for Tivoli EIF process definition

```
nco_service 'Core'
{
    ServiceType      =      Master
    ServiceStart     =      Auto
    process 'MasterObjectServer' NONE
    process 'nco_gw_eif_tsrm' 'MasterObjectServer'
    process 'nco_probe_eif_tsrm' NONE
}
```

- d. You must restart the Process Agent for the new configuration to take effect. As the root user, issue the `/etc/init.d/nco restart` command.
- e. Issue the `$OMNIHOME/bin/nco_pa_status` command to verify that the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility is running. Example 5-52 illustrates the IBM Tivoli Netcool/OMNIBus Probe for Tivoli Event Information Facility running under Process Control.

Example 5-52 The `nco_pa_status` command

```
[netcool@tbsm bin]$ ./nco_pa_status
Login Password:
```

Service Name	Process Name	Hostname	User	Status	PID
Core	MasterObjectServer	tbsm	netcool	RUNNING	6495
	nco_gw_eif_tsrm	tbsm	netcool	RUNNING	6606
	nco_probe_eif_tsrm	tbsm	netcool	RUNNING	6496

Verify that the IBM Tivoli Netcool/OMNIBus and IBM Tivoli Service Request Manager integration works as expected by performing the following tests:

1. Insert an event into IBM Tivoli Netcool/OMNIBus. Verify that an incident or service request is opened in IBM Tivoli Service Request Manager according to Table 5-4 on page 196.
2. Update the severity of an event in IBM Tivoli Netcool/OMNIBus. Verify that the priority of the event is updated in IBM Tivoli Service Request Manager according to Table 5-4 on page 196. The priority of the event will change; however, the original form as an incident or service request will not change.
3. Resolve or close an event in IBM Tivoli Service Request Manager. Verify that the event severity is set to 0 or clear in IBM Tivoli Netcool/OMNIBus.

Configuring event filtering

By default, the IBM Tivoli Netcool/OMNIbus and IBM Tivoli Service Request Manager integration will cause all events in IBM Tivoli Netcool/OMNIbus to open incidents or service requests in IBM Tivoli Service Request Manager. You can configure a filter in the IBM Tivoli Netcool/OMNIbus Gateway for Tivoli Event Information Facility so that only certain events will be sent from IBM Tivoli Netcool/OMNIbus to IBM Tivoli Service Request Manager:

1. On the IBM Tivoli Netcool/OMNIbus Gateway for Tivoli Event Information Facility server, edit the Table Replication Definition file:
\$OMNIHOME/gates/tivoli_eif/tivoli_eif.rdrwtr.tblrep.def.
2. By default, a filter already exists for UpdatedByTSRM not equal to 1. This filter will not resend events that were already sent to IBM Tivoli Service Request Manager. Add an additional Object Server field to the filter.

Example 5-53 illustrates adding the SendToTSRM field to the gateway filter. An automation or tool can be created to update this field to 1 for events that must be ticketed in IBM Tivoli Service Request Manager.

Example 5-53 Updating SendToTSRM field

```
REPLICATE ALL FROM TABLE 'alerts.status'  
  USING MAP 'StatusMap'  
  FILTER WITH 'UpdatedByTSRM <> 1 and SendToTSRM = 1'
```

5.4 Reports integration

In this section, we discuss report integration. Reports from Tivoli products are currently integrated into a common platform.

5.4.1 Tivoli Common Reporting

Tivoli Common Reporting is a reporting tool available to Tivoli products. It consists of a Web user interface that is based on Tivoli Integrated Portal that presents scheduled or on demand reports by obtaining data from Tivoli products. User access to the reports' other parameters can be defined separately. The main interface for Tivoli Common Reporting is shown in Figure 5-47 on page 219.

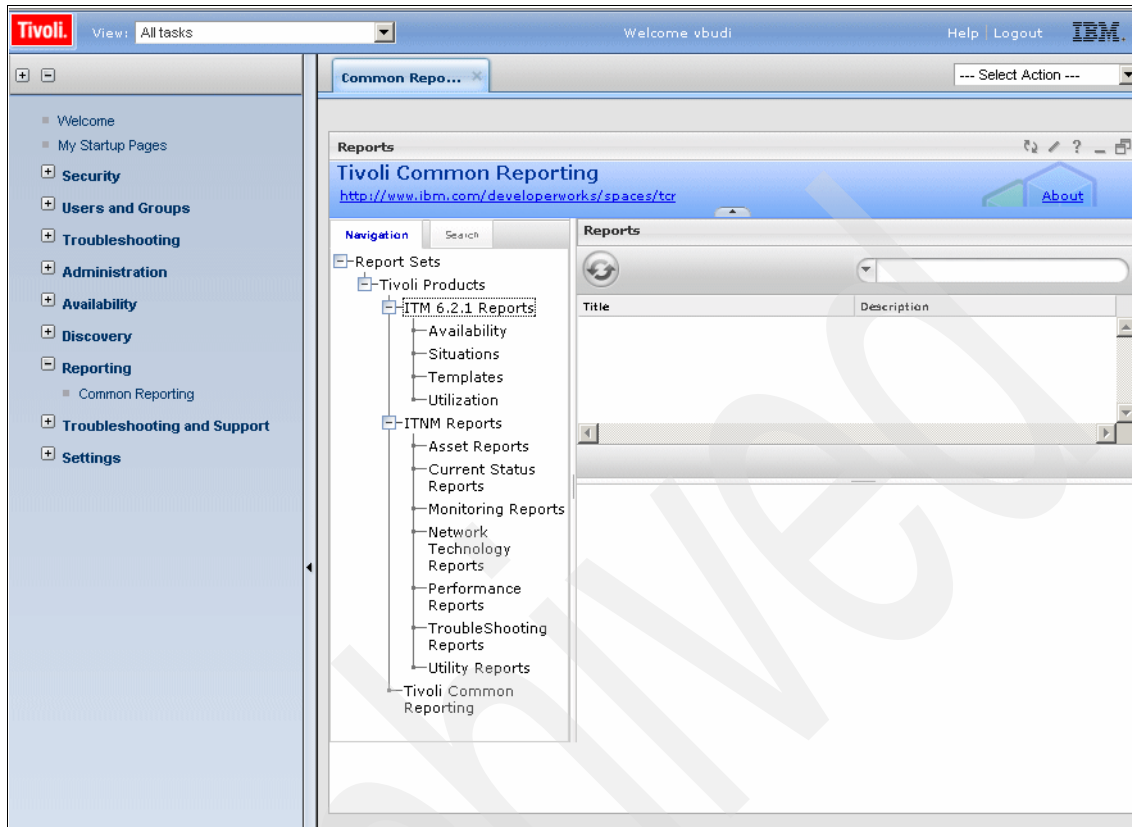


Figure 5-47 Tivoli Common Reporting

Tivoli Common Reporting accepts command-line interface and also contains a data store for storing and organizing reports, report designs, and related resources. Currently, Tivoli Common Reporting is based on the Business Intelligence Reporting Tool (BIRT) engine. BIRT is used to generate the formatted reports. The reports can be created or modified through using the BIRT report designer. This tool is downloadable from this Web site:

<http://www.ibm.com/software/tivoli/opal/details?catalog.label=1TW100T02>

More information and links are available in the Related Information tab under the About link in Tivoli Integrated Portal. It is shown in Figure 5-48 on page 220.

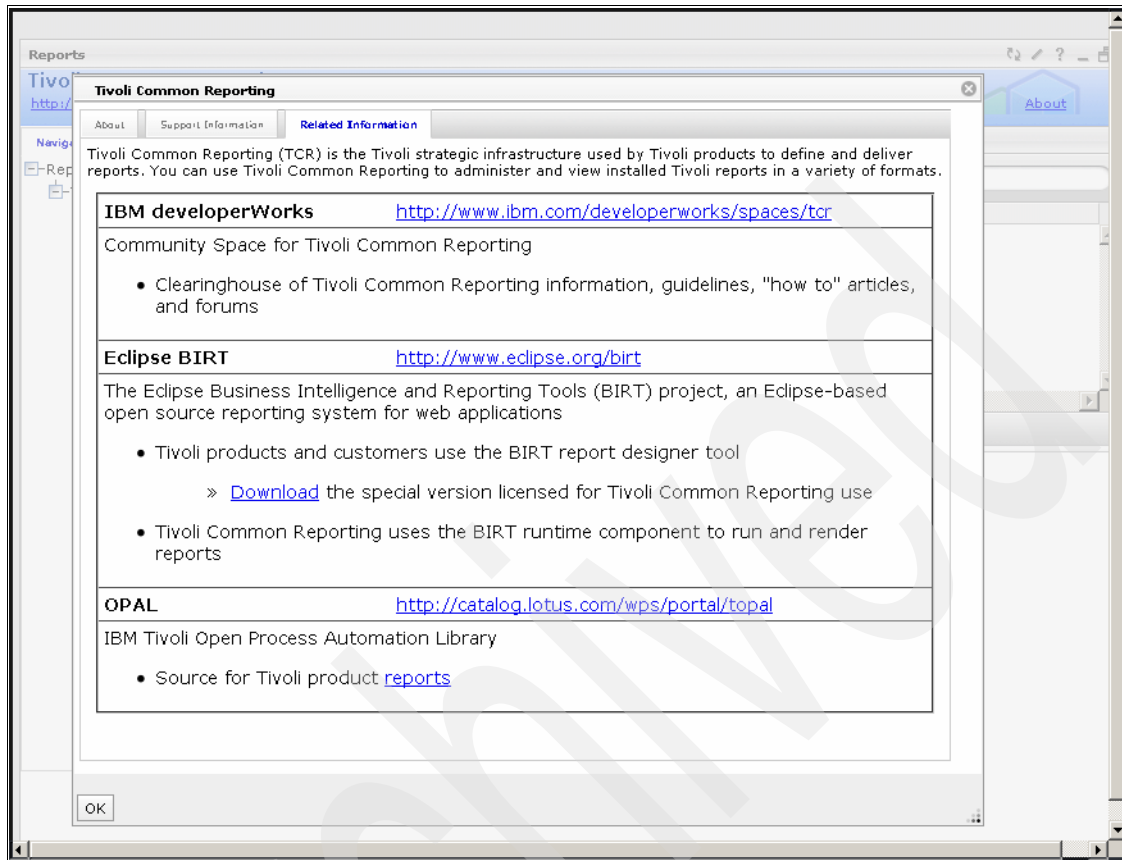


Figure 5-48 The related information links

When Tivoli Integrated Portal is installed through IBM Tivoli Business Service Manager or IBM Tivoli Network Manager for IP installation, Tivoli Common Reporting is also installed. This capability allows you to consolidate report data from various Tivoli products.

Certain Tivoli products come with readily available report design files whereas other products need the report to be downloaded from the Tivoli Open Process Automation Library (OPAL). Or, you can design the report manually using BIRT Report Designer.

The main page that contains all of the references to available reports and tools is available at this Web site:

<http://www.ibm.com/developerworks/spaces/tcr>

Check its Tivoli Report Catalog link to understand which products have pre-designed report packages.

5.4.2 Importing the report package

In this section, we demonstrate how to import and run report packages into Tivoli Integrated Portal from a product report package or custom report package files. We use IBM Tivoli Monitoring reports as an example.

The *Tivoli Common Reporting Guide* is available in the Tivoli Integrated Portal installation path at `/opt/IBM/tivoli/tip/products/tcr/docs`. Refer to the Tivoli Common Reporting documentation for additional requirements.

To import the report package:

1. Tivoli Common Reporting requires drivers to connect to remote data sources and to retrieve data into the reports. The drivers for certain databases are already implemented during the installation of Tivoli Common Reporting. You can copy additional drivers from other data sources to `/opt/IBM/tivoli/tip/products/tcr/lib/birt-runtime-2_2_1/ReportEngine/plugins/org.eclipse.birt.report.data.oda.jdbc_2.2.1.r22x_v20070919/drivers`.
2. After you have the reporting package in the Tivoli Integrated Portal machine, you can start importing the files into Tivoli Common Reporting. Go to the Tivoli Integrated Portal at **Reporting** → **Common Reporting**, right-click **Report Sets**, and select **Import Report Package**. Figure 5-49 on page 222 shows an example of importing an IBM Tivoli Monitoring 6.2.1 report package file.

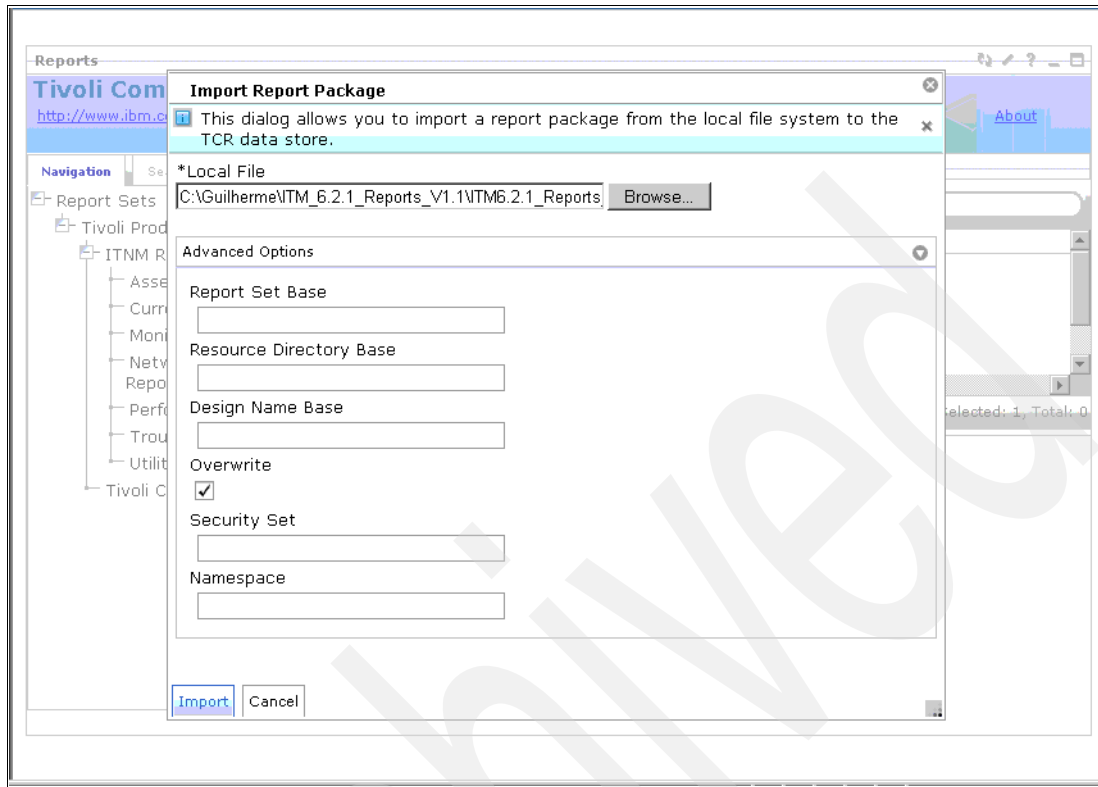


Figure 5-49 Importing a report package file

3. Create a data source to retrieve data for the reports. From the Tivoli Common Reporting page, expand the **Report Sets** → **Tivoli Products** → **ITM 6.2.1 Reports**, and select any report category. Right-click any report and select **Data Sources**. Figure 5-50 on page 223 shows an example.

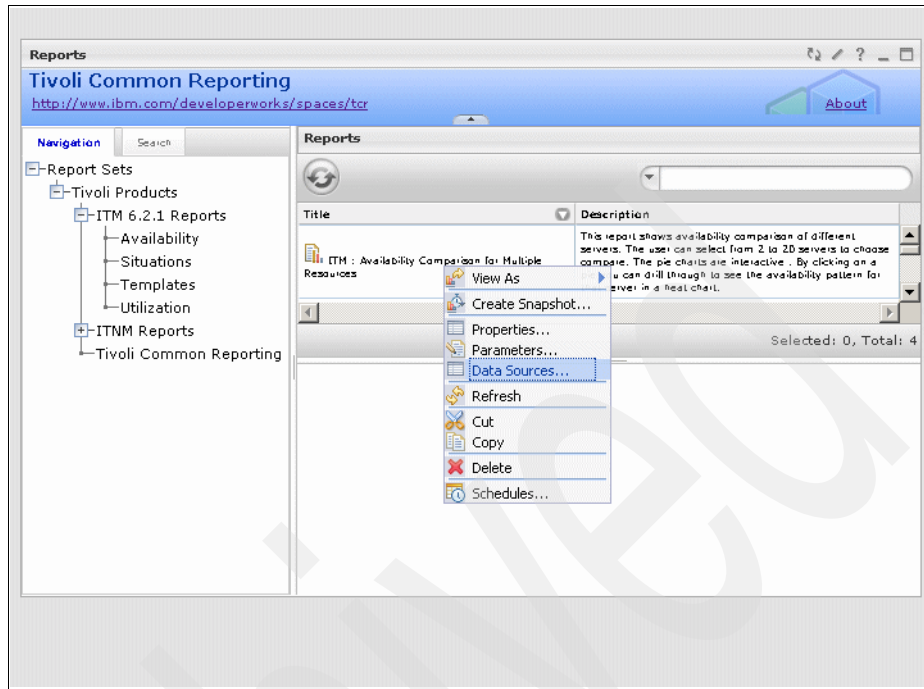


Figure 5-50 Setting the data source

4. Click **Edit** to modify the data source settings as shown in Figure 5-51 on page 224. Configure the data source based on Table 5-5 on page 224. Click **Save** when done.

Reports

Tivoli

http://

Report Data Sources

Edit the ITM : Availability Comparison for Multiple Resources report JDBC/JNDI data source.

Display Name: ITM_TDW

User ID: itmuser

Password:

JDBC Driver: com.ibm.db2.jcc.DB2Driver

JDBC URL: jdbc:db2://itm.itso.ral.ibm.com:50000/A

JNDI Name:

Additional comments to define data source.
none

Enabling a JDBC Driver

Save Cancel

Figure 5-51 Data source settings

After configuring the data source, click **Save**.

Table 5-5 Data source parameters

Field	Value	Comment
Display Name	ITM_TDW	A name for the data source
User ID	itmuser	The user to connect to the remote data source. We connect to IBM Tivoli Monitoring data warehouse.
Password	itso4you	The password of the user to connect to the remote data source
Java Database Connectivity (JDBC™) Driver	com.ibm.db2.jcc. DB2Driver	The driver related to the type of database used as the data source. We use a DB2 JDBC driver.
JDBC URL	jdbc:db2://itm:50000/WAREHOUS	The JDBC URL to the database used as a data source. We use the host name, port, and database name where the Tivoli Warehouse was created.

- After importing the report package file and configuring the data source, you are ready to run a report for data that exists in the data source. Selecting the **Report Sets** → **Tivoli Products** → **ITM 6.2.1 Reports** → **Utilization** report, right-click **CPU Utilization Detail for Single Resource**, and select **Create Snapshot**. See Figure 5-52.



Figure 5-52 Creating a report snapshot

- Fill the required fields, and click **Run**, as shown in Figure 5-53.

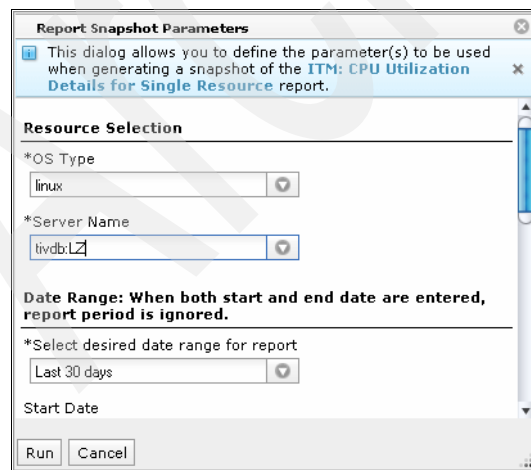


Figure 5-53 Report snapshot

7. The report runs in the background. When the status is completed, you can open the report, as shown in Figure 5-54. Select **View As** → **HTML**, or you can also choose to show the result as a PDF document, a Microsoft Excel® spreadsheet, or a PostScript® file.

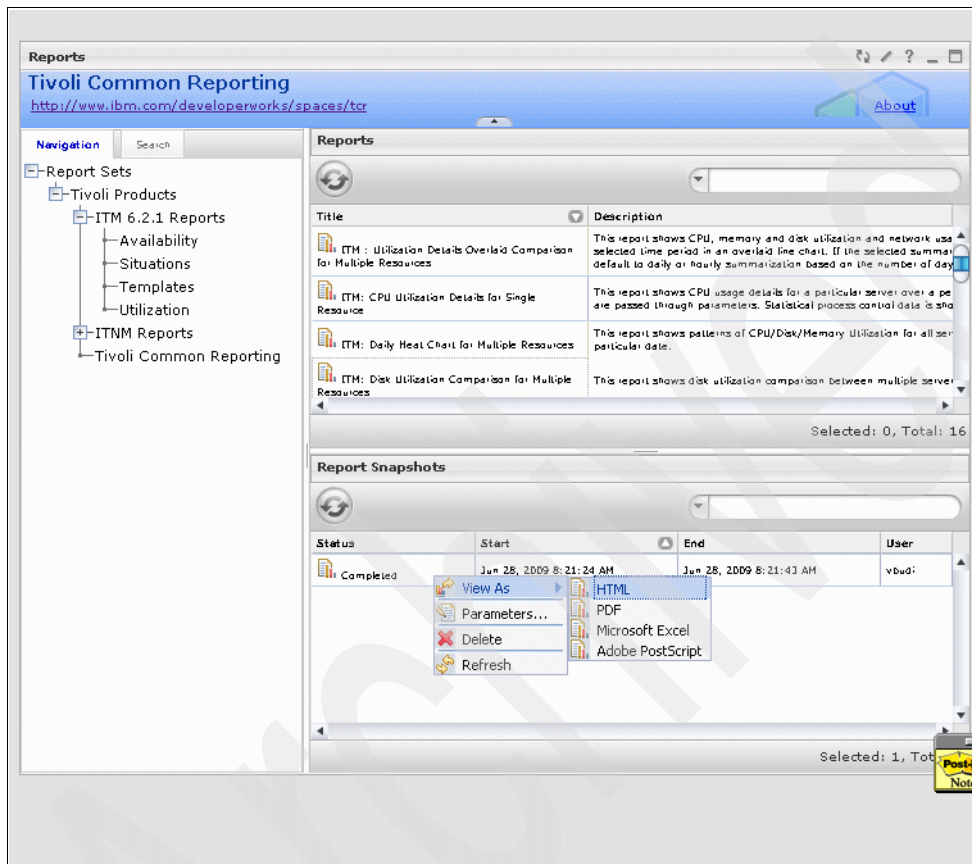


Figure 5-54 Opening a completed report

8. Figure 5-55 on page 227 shows the result.

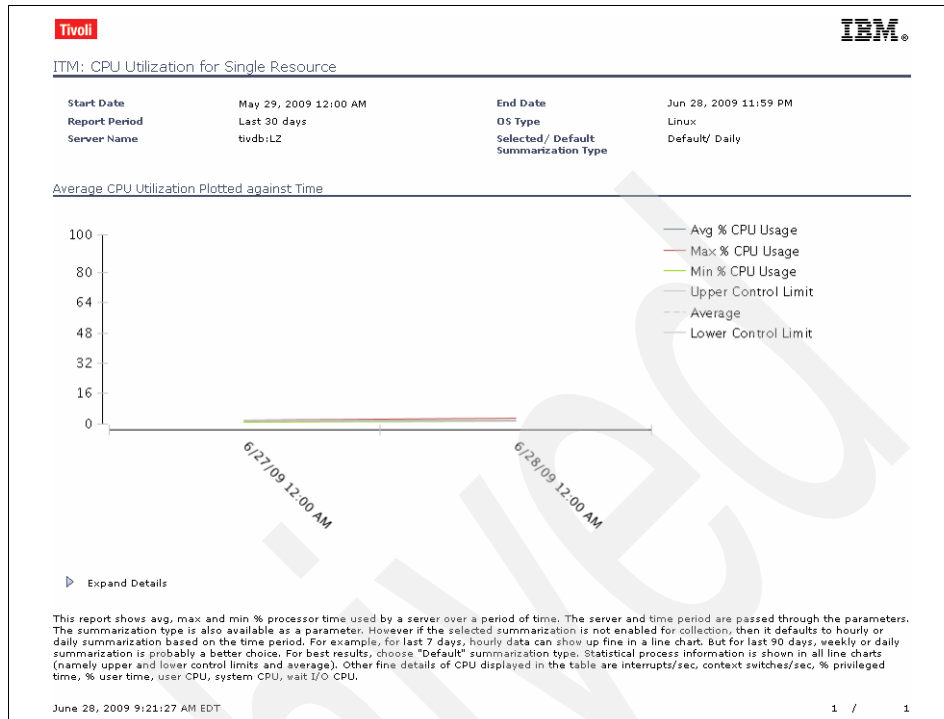


Figure 5-55 Report result

5.4.3 Available reports for Tivoli Common Reporting

The most up-to-date report list is available at the Tivoli Common Reporting report catalog Web page at <http://www.ibm.com/developerworks/spaces/tcr>. Go to the reporting table. That list is fairly current as of the writing of this book. Any product names with hyperlinks are provided on OPAL; the rest of the product names come with the products.

These products support reports in Tivoli Common Reporting:

- ▶ **Asset Management**

The reports for Asset Management and Service Management that are based on Tivoli Process Automation Engine cannot be run under Tivoli Integrated Portal. They must be run from Tivoli Process Automation Engine:

- IBM Maximo Enterprise Asset Management 7.1
- IBM Asset Manager for IT 7.1

- ▶ **Security Management:**

- Tivoli Access Manager for e-business 6.1

- Tivoli Federated Identity Manager 6.1
- Tivoli Identity Manager 5.0
- ▶ Service Management:
 - IBM Tivoli Change and Configuration Manager Database
 - IBM Tivoli Release Process Manager
 - IBM Tivoli Service Request Manager
- ▶ Business Application Management:
 - IBM Tivoli Business Service Manager 4.2
 - IBM Tivoli Business Service Manager 4.1.1
 - IBM Tivoli Performance Analyzer 6.1.1
- ▶ Server, Network, and Device Management:
 - IBM Tivoli Monitoring 6.2 OS Agent
 - IBM Tivoli Monitoring for Energy Management
 - IBM Tivoli Monitoring 6.2 VMWare VI Agent
 - IBM Tivoli Composite Application Manager for Response Time 6.2
 - IBM Tivoli Composite Application Manager for SOA 7.1
 - IBM Tivoli Composite Application Manager for WebSphere/J2EE
 - IBM Tivoli Composite Application Manager for Applications
 - IBM Tivoli Decision Support for z/OS
 - IBM Tivoli OMEGAMON® XE for IMS™ 4.1
 - IBM Tivoli OMEGAMON XE for CICS® 4.1
 - IBM Tivoli OMEGAMON XE on z/OS 4.1
 - IBM Tivoli OMEGAMON XE for z/VM® and Linux 4.1
 - IBM Tivoli OMEGAMON XE for Mainframe Networks 4.1
 - IBM Tivoli OMEGAMON XE for Storage 4.2
 - IBM Tivoli OMEGAMON XE for Messaging 7.0
 - IBM Tivoli Netcool/OMNibus 7.2.1
 - IBM Tivoli NetView® for z/OS 5.3
 - IBM Tivoli Network Manager for IP 3.7
 - IBM Tivoli System Automation 3.1

Several products directly use Tivoli Common Reporting, which means that they have reports (*.rptdesign) that are part of the product:

- ▶ Tivoli Storage Manager 6.1
- ▶ IBM Tivoli Application Dependency Discovery Manager 7.1.2
- ▶ Tivoli Security Policy Manager 7.0
- ▶ IBM Tivoli OMEGAMON XE for IMS 4.2
- ▶ IBM Tivoli OMEGAMON XE for z/OS 4.2
- ▶ IBM Tivoli Network Manager for IP 3.8
- ▶ IBM Tivoli Composite Application Manager for SOA 7.1.1
- ▶ IBM Tivoli Access Manager for e-Business 6.1

Additional products use an interface that is similar to Tivoli Common Reporting, but they do not include Tivoli Common Reporting directly. Those products are based on Tivoli Process Automation Engine and are all in the Asset Management and Service Management sections. The reports are part of the products for these areas:

- ▶ IBM Maximo for Life Sciences
- ▶ IBM Maximo Calibration
- ▶ IBM Maximo for Transportation
- ▶ IBM Maximo for Nuclear Power
- ▶ IBM Maximo for Utilities
- ▶ IBM Maximo Spatial Asset Management
- ▶ IBM Maximo Asset Management for Service Providers
- ▶ IBM Maximo for Oil and Gas
- ▶ IBM Maximo for Government
- ▶ IBM Maximo Asset Management Essentials
- ▶ IBM Tivoli Provisioning Manager

Additionally, certain products use an interface that is similar to Tivoli, but they use the Tivoli Process Automation Engine reporting UI instead of the Tivoli Common Reporting UI:

- ▶ IBM Maximo Enterprise Asset Management
- ▶ IBM Asset Management for IT
- ▶ IBM Tivoli Change and Configuration Manager Database
- ▶ IBM Tivoli Release Process Manager
- ▶ IBM Tivoli Server Request Manager

Tivoli Workload Scheduler 8.5 uses a similar interface to Tivoli, but it directly imbeds BIRT. The reports are part of the product.

5.5 Other data integration

IBM Tivoli Usage and Accounting Manager V7.1.1 defines a new collector that allows usage data to be retrieved from Tivoli Data Warehouse. This capability allows you to generate financial reports based on historical monitoring data that is collected by IBM Tivoli Monitoring. This function is documented in the collector documentation at this Web site:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.ituam.doc_7.1.2/admin_win_dc/c_tdw_collector.html

Navigation integration

In this chapter, we discuss navigation integration or “launch in context” configuration across various Tivoli products.

We divide the discussion into these areas:

- ▶ 6.1, “Navigation integration overview” on page 232
- ▶ 6.2, “Building a target URL” on page 232
- ▶ 6.3, “Launching out capabilities” on page 261

6.1 Navigation integration overview

Integration between various products will actually be realized when an operation can actually use multiple products seamlessly, such as getting the information from one product and directly linking to another product to get more information or to accomplish certain tasks.

This integration, which is commonly known as the “*launch in context*” feature, requires that security integration for single sign-on (SSO) is in place so that operators do not have to enter their user IDs and passwords repeatedly. Launch in context requires that the context exists and is established so that launching another application provides a relevant view to the operator instead of the initial page. The context is established by having data integration in place.

Currently, several Graphical User Interface (GUI) technologies are used by various Tivoli products:

- ▶ Tivoli Integrated Portal
- ▶ Tivoli Enterprise Portal
- ▶ Integrated Solution Console
- ▶ Maximo-based user interface
- ▶ Java Web Start-based user interface

Each of these interfaces has a separate flow and separate procedures for preparing launch in context. Individual products also have their own specific way of adding the menu launcher and providing links. In this chapter, we discuss the customizations for the products in our environment.

We divide this discussion of the navigation integration into two areas:

- ▶ Launch in context target (or sometimes also called *land in context*), which determines the required URL context and information for invoking a certain function for the product. See 6.2, “Building a target URL” on page 232.
- ▶ Configuration of launch in context functionality, which includes defining the target URL, performing variable substitutions, and associating the launch to an object context. See 6.3, “Launching out capabilities” on page 261.

6.2 Building a target URL

In this section, we describe the target launching of separate products that can be used for launch in context. The discussion is divided into these areas:

- ▶ 6.2.1, “IBM Tivoli Application Dependency Discovery Manager” on page 233
- ▶ 6.2.2, “Tivoli Process Automation Engine-based products” on page 235

- ▶ 6.2.3, “Tivoli Enterprise Portal” on page 245
- ▶ 6.2.4, “Tivoli Integrated Portal” on page 249
- ▶ 6.2.6, “IBM Tivoli Netcool/Impact operator view” on page 255
- ▶ 6.2.7, “IBM Tivoli Storage Productivity Center for Data” on page 255

6.2.1 IBM Tivoli Application Dependency Discovery Manager

You can launch IBM Tivoli Application Dependency Discovery Manager views from other Tivoli products to view more detailed information about your environment. You can configure your application to launch the IBM Tivoli Application Dependency Discovery Manager view by specifying a URL in the following syntax:

<protocol>://<TADDM hostname>:<port>/<ContextRoot>?<queryString>

Protocol	This protocol can be http or https, depending on whether Secure Sockets Layer (SSL) is enabled.
TADDM hostname	This name is the host name for the IBM Tivoli Application Dependency Discovery Manager server to which you are launching.
Port	The port number. The default value is 9430.
ContextRoot	The context root of the IBM Tivoli Application Dependency Discovery Manager launch Web Application as deployed on a TOMCAT server. The default is cdm/servlet/LICServlet.
queryString	A set of name-value pairs. Each pair is separated by an ampersand (&) symbol and is delimited by the equal (=) sign.

Table 6-1 on page 234 lists the valid and available name-value pairs for IBM Tivoli Application Dependency Discovery Manager.

Table 6-1 Valid launch in context parameters

Parameter	Description	Valid values
Guid	Specify the GUID for the configuration item (CI).	Valid string representation of a GUID
Target	Specify whether or not to launch a new product console instance. This parameter is ignored in the case of Web console.	These are the valid values: ► New ► Existing
Graph	Specify the type of topology graph to be launched.	These are the valid values for Web and Java console: ► physicalinfrastructure ► applicationinfrastructure ► businessapplications\ These are the valid values for Java console only: ► app_software ► app_physical ► bus_svc_software ► bus_svc_physical ► collation_relationship ► collation_physical
View	Only use when you want to display the change history.	The only valid value is changehistory.
Days_previous	Specify the number of days from the current date to go back for change history.	A valid number
Console	Specify whether to launch the Web or Java console. The default is Java console.	Valid values are Web or Java.

Example 6-1 shows an example of a valid URL to launch in context IBM Tivoli Application Dependency Discovery Manager from another application.

Example 6-1 Valid launch in context URL

<http://taddm.itso.ra1.ibm.com:9430/cdm/servlet/LICServlet?graph=businessapplications>

There are other URLs apart from the default main URL for performing launch in context for IBM Tivoli Application Dependency Discovery Manager:

- ▶ Launch a configuration item (CI) details panel directly:
`http://<host>:9430/cdm/detailsPanel.do?objectID=guidDetails<guid>&domainIP=localhost&pageID=domain&debug=9`
- ▶ Displaying CI change history:
`http://<host>:9430/cdm/changehistory.do?objectID=guidDetails<guid>&domainIP=localhost&pageID=domain&debug=9`
- ▶ Querying a CI attribute:
`http://<host>:9430/cdm/query.do`

6.2.2 Tivoli Process Automation Engine-based products

The discussion in this section applies to all products that use the Maximo interface (/maximo/ui):

- ▶ IBM Tivoli Change and Configuration Management Database
- ▶ IBM Tivoli Provisioning Manager
- ▶ IBM Tivoli Service Request Manager
- ▶ IBM Tivoli Asset Manager for IT
- ▶ IBM Process Management Products

To be able to launch into the IBM Service Management product interface, you must identify the application into which you want to launch. The list of applications is available under **Go To** → **System configuration** → **Platform configuration** → **Application Designer**. A partial list of applications is shown in Figure 6-1 on page 236.

For any fields that you want to use for filtering your query, you can select the field and choose **Properties**. Figure 6-3 shows the field properties.

Control ID	results_showlist_tablebody_6
Default Label	Classification
Label	
Label Attributes	
Label Source ID	
Title Attributes	
CSS Class Name	
Show Filter?	☒
Filterable?	☐
Sortable?	☐
Type	
Attribute	CLASSSTRUCTURE.CLASSIFICATIONID
Linked Controls ID	
Go To Applications	
Input Mode	
Menu Type	
Lookup	
Change Event	
Turn Smart Fill Off?	☐
Is LD Readonly?	☐
URL Attribute	
Event	
Event Description	
Event Icon	
Target ID	
Signature Option	
Sig Option Data Source ID	
Detail Image	

Configure Conditional Properties

Figure 6-3 Column property

The content of the Attribute field is the field ID that you can use in the launch in context query to simulate entering a value in that particular field. If you need a more complex query with a combination of various fields, you must gather the actual table data structure. The structure of the database object and also the underlying table definition can be inferred from **Go To → System Configuration → Platform Configuration → Database Configuration** and selecting the main object of the application. In our example, the CI object database configuration is shown in Figure 6-4 on page 238.

The screenshot shows the 'Database Configuration' interface with the 'Object' tab selected. The 'Object' field is set to 'CI' and the 'Table' field is set to 'Configuration Item Table'. The 'Status' field is empty. The 'Details' section includes fields for Service (CI), Description (Configuration Items), Entity (CI), Class (psdi.pmcfg.app.cilifecycle.PmCfgCISet), Extends Object, and Level (SYSTEM). The 'Table' section includes checkboxes for Main Object?, Persistent? (checked), User Defined?, Imported?, and Internal?, along with fields for Storage Partition (MAXDATA), Unique Column (CIID), Language Table, Language Column (LANGCODE), Alternate Index, and Trigger Root (CI). The 'View' section includes checkboxes for View? and Automatically Select?, and fields for View Where, Join to Object View Select, and View From. The 'Audit' section includes checkboxes for Audit Enabled? and Audit Table, and fields for E-audit Filter and E-signature Filter.

Figure 6-4 Database structure

In Figure 6-4, the table name is shown in the Entity field. The individual column definition can be seen on the Attributes tab, as shown in Figure 6-5.

The screenshot shows the 'Database Configuration' interface with the 'Attributes' tab selected. The 'Object' field is set to 'CI' and the 'Table' field is set to 'Configuration Item Table'. The 'Status' field is empty. The 'Attributes' section displays a list of columns with their respective types, lengths, scales, and required status.

Status	Attribute	Description	Type	Length	Scale	Required?
▶	ACTCINUM	Actual CI Identifier	UPPER	150	0	☐
▶	ASOFDATE	Displays the relationships that existed with the	DATETIME	10	0	☐
▶	ASSETLOCORGID	The Organization to which the CI belongs to if it	UPPER	8	0	☐
▶	ASSETLOCSITEID	The Site to which the CI belongs to if it is Asse	UPPER	8	0	☐
▶	ASSETNUM	Identifier of the Asset which is the CI	UPPER	12	0	☐
▶	CALNUM	Calendar that is associated with the CI	UPPER	8	0	☐
▶	CHANGEBY	Last Modified By	UPPER	30	0	☐
▶	CHANGEDATE	Change Date of CI	DATETIME	10	0	☐
▶	CIGUID	Actual CI GUID	ALN	192	0	☐
▶	CIID	Unique Id	INTEGER	12	0	☒

New Row

Figure 6-5 Database column structure

Now, we can define a query and display an application for the IBM Service Management application.

The base URL has this syntax:

`http://<hostname>/maximo/ui?<argument>`

where:

hostname	IBM Service Management server host name
argument	A set of name-value pairs, similar to the HTTP GET request

In the arguments, you can define the following information:

- ▶ To run an application, you can use the `event=loadapp&value=<appname>` argument.
- ▶ To open a record, you must find the unique ID for the record. Use `uniqueid=<uniqueidnumber>`. The unique ID is created using an identity column, and it contains an integer sequence with increasing value.
- ▶ To further qualify the application, you can use a query using `additionalevent=<useqbe|sqlwhere>&additionaleventvalue=<search argument>`

where:

- When an additional event contains `useqbe`, the additional event value contains a list of attribute value pairs in the format of “key1=value1|key2=value2 ...”.
- When an additional event contains `sqlwhere`, the additional event value has the where clause of a valid SQL statement.
- ▶ Using the argument `forcereoad=true` is recommended to ensure that the page is refreshed on subsequent launch of the target page.

Note: If product that is launching into Tivoli Process Automation Engine will be launching repeatedly from the same browser, then add the `uisessionid` to the url to prevent multiple sessions from being created.

IBM Tivoli Asset Manager for IT

Table 6-2 on page 240 lists the application IDs that relate to IBM Tivoli Asset Manager for IT.

Table 6-2 IBM Tivoli Asset Manager for IT objects

Application ID	Database object	Comments
8	ASSET	Asset application
71	DPAMADAPTER	Adapter Conversion application
72	DPAMMANUFACTURER	Manufacturer Conversion application
73	DPAMOS	OS Conversion application
74	DPAMPROCESSOR	Processor Conversion application
75	DPAMSOFTWARE	Software Conversion application
76	DPAMWSUITE	Software Suite application
77	DPAMSWUSAGE	Software Usage application
68	DPACOMPUTER	Computers application
69	NETDEVICE	Network Device application
70	NETPRINTE	Network Printer application
159	RECONASSETLINK	Asset Reconciliation Link application
157	RECONASSETRESULT	Asset Reconciliation Results application
132	RECONCOMPRULE	Reconciliation Comparison Rules application
95	RECONTASK	Reconciliation Task application
97	RECONLINKRULE	Reconciliation Link Rule application
96	RECONTASKFILTER	Reconciliation Task Filter application
160	SFWLICENSE	Software License Contract application
161	SFWVIEW	Software View application

IBM Tivoli Provisioning Manager

Tivoli Provisioning Manager provides support for other products to access a specific object in Data Model by specifying a specific URL, which we describe in detail in this section.

IBM Tivoli Provisioning Manager uses applications to access various types of objects. This application will be loaded through an HTTP connection by specifying it in the URL:

► tptask

Use this application to access objects managed by Provisioning Task Tracking. For example, if you want to know the status of the latest task on a hardware and software inventory scan, use this application with the parameter of that specific Provisioning Task ID.

► tpwfstat

Use this application to access objects that are managed by Provisioning Workflow Status. For example, when you scan hardware and software over Scalable Deployment Infrastructure (SDI), a workflow that is called Cit_SDI_OnDevice is executed with a request ID associated to it. If you want to know the status of that specific workflow, use this application with the request ID as the parameter.

► tpdcmfind

Use this application to search any objects whose attributes are persistent in the Maximo business object TPDCMOBJECT. For example, if you have discovered a machine using Deployment Engine and want to know what attributes are associated to it, you can use this application with the object ID as the parameter.

Table 6-3 on page 242 shows the mapping of several important object type IDs and object type names. For the complete list, you can access the table called MAXIMO.DCM_OBJECT_ID in the MAXDB71 database.

Table 6-3 Important object IDs

Object ID	Object ID name	Description
0	KANAHHA	Entire system
1	APPLICATION	Application
2	CLUSTER	Application tier
3	COMPUTER	Computer
4	NETWORK_INTERFACE	Network interface
5	NIC	Network interface card
6	SWITCH	Switch
17	SOFTWARE_PRODUCT	Software installable
18	FILE_REPOSITORY	File repository
36	SOFTWARE_PATCH	Software patch
88	SOFTWARE_RESOURCE	Software resource
89	SOFTWARE_RESOURCE	Software resource
90	SOFTWARE_INSTALLATION	Software installation
91	SOFTWARE_CONFIGURATION	Software configuration
92	SOFTWARE_APPLICATION_DATA	Software application data
93	FILE	File
106	TPM_DEPOT	IBM Tivoli Provisioning Manager Depot
107	TPM_REGION	IBM Tivoli Provisioning Manager Region
108	TPM_ZONE	IBM Tivoli Provisioning Manager Zone
136	COMPUTER_GROUP	Computer group

After you obtain the object ID, you can go directly to that object. For example, we can open one Provisioning Computer called tivapp1.itso.ral.ibm.com with the object ID of 5821 by entering the following URL:

<https://tpm.itso.ral.ibm.com:9045/maximo/ui/?event=loadapp&value=tpdcmfind&uniqueid=5821>

Figure 6-6 shows detailed information about one Provisioning Computer object with an object ID of 5821.

The screenshot displays the IBM Maximo Provisioning Computers interface. The top navigation bar includes links for Web Replay, Bulletins (0), Go To, Reports, Start Center, Profile, Sign Out, and Help. Below the navigation bar is a search and action area with a 'Find:' field and a 'Select Action' dropdown. The main content area is divided into several sections:

- Computer Information:** Includes fields for Computer (lvapp1.lso.ral.ibm.com), Globally Unique Identifier, and Configuration Item.
- Configuration:**
 - Management IP Address:** 9.42.171.36
 - Operating System:** Red Hat Enterprise Linux
 - Computer Template:** (empty)
 - Locale:** en_US.UTF-8
- Details:**
 - Model Type:** None
 - Architecture:** intel
 - Serial Number:** VMware-58 4d 98 49 c1 b
 - Manufacturer:** VMware, Inc.
- Operations:**
 - Framework:** Scalable Distribution Infrat
 - Agent:** TCA-1.4.1.0
 - Agent Status:** Running
- Compliance:**
 - Security:** Configuration required
 - Software:** Configuration required
 - Patches:** Configuration required
- Static Provisioning Groups:** A table showing groups like 'Computer and Inventory' and 'Discovery_TgtsGroup'.
- Dynamic Group Association:** A section for associating dynamic groups.
- Recent Discovery Scans:** A table listing discovery scans with their dates.

The bottom of the interface shows a 'Done' button and a 'Local intranet' link.

Figure 6-6 Detailed information about an object with the object ID of 5821

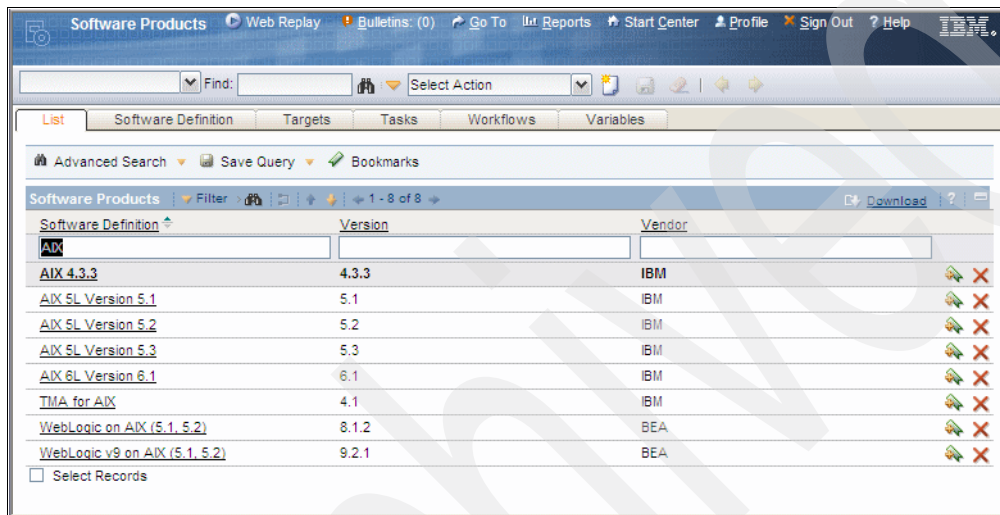
For other applications, to access specific object types, you have to use the specific application that has that object type as the primary Maximo business object. For example, to access the object type of Computer and Software Resource, specific applications' tpserver and tpswres have to be used. We will discuss the mapping of which application is used to access which object type later in more detail.

Other applications are used to access specific objects other than Provisioning Tasks and Workflow. Even though information can be retrieved using the tpdcmfind application, there are certain attributes that are not visible to tpdcmfind. Those specific attributes are not persistent in the Maximo business object TPDCMOBJECT.

For example, to retrieve information about software products in the IBM Tivoli Provisioning Manager database, you can use the tpsoftware application. Enter the following URL:

<https://tpm.itso.ral.ibm.com:9045/maximo/ui/?event=loadapp&value=tpsoftware&additionalEvent=useqbe&additionalEventvalue=Name=AIX>

This URL lists all software whose names consist of AIX word, as shown in Figure 6-7.



Software Definition	Version	Vendor
AIX		
AIX 4.3.3	4.3.3	IBM
AIX 5L Version 5.1	5.1	IBM
AIX 5L Version 5.2	5.2	IBM
AIX 5L Version 5.3	5.3	IBM
AIX 6L Version 6.1	6.1	IBM
TMA for AIX	4.1	IBM
WebLogic on AIX (5.1, 5.2)	8.1.2	BEA
WebLogic v9 on AIX (5.1, 5.2)	9.2.1	BEA

Figure 6-7 List of software products

If you want to retrieve only software products that are at version 4.3.3, for example, you can enter the following URL:

<https://tpm.itso.ral.ibm.com:9045/maximo/ui/?event=loadapp&value=tpsoftware&additionalEvent=useqbe&additionalEventvalue=Version=4.3.3>

And for a specific vendor, you can enter the following URL:

<https://tpm.itso.ral.ibm.com:9045/maximo/ui/?event=loadapp&value=tpsoftware&additionalEvent=useqbe&additionalEventvalue=Vendor=IBM>

The following list shows the mapping between the Data model object type and the application that can be used to access it:

TPACL	Access control list
TAPPL	Application
TPBLADE	Blade administration server
TPBOOTSRV	Boot server

TPCLUSTER	Cluster
TPCLSTDMN	Cluster domain
TPSERVERS	Computer
TPGROUP	Group of computers
TPCUST	Customer
TPDCD	DCD Management center
TPDEVMOD	Device model
TPDISC	Discovery activity
TPFILERE	File repository
TPGROUP	Group
TPIMAGE	Software image
TPLDBAL	Load balancer
TPPOWERU	Power unit
TPSAN	SAN
TPSTSUBST	SAN frame
TPCMPTMPL	Computer server template
TPSWRES	Software application data, installation, and instance
TPSOFTWARE	Software module
TPSWPATCH	Software patch
TPSWRES	Software resource
TPSTACK	Software stack
TPRESPOOL	Spare resource pool
TPSTPOOL	Storage allocation pool
TPSTGTMPL	Storage template
TPSUBNETS	Sub-net information
TPSWITCHS	Switch
TPTERMSRV	Terminal server
TPUNKRSRC	Unknown TCP/IP Device
TPVST	Virtual server template
TPVLANS	Virtual LAN
TPSTGMGR	Volume Manager

6.2.3 Tivoli Enterprise Portal

Tivoli Enterprise Portal can be invoked using the Web browser or as a Java Web start element. The Java Web start version of Tivoli Enterprise Portal is accessed by using the URL:

`http://<tep_host>:15200/LICServletWeb/LICServlet`

These Tivoli Enterprise Portal clients have the same functionalities. The differences mainly come from how they are invoked. The browser client allows an applet that runs Tivoli Enterprise Portal within a Web browser. The applet also runs in the Java Web start version of Tivoli Enterprise Portal.

The LICServlet allows you to connect directly to a context environment on the portal. This servlet supports similar parameters as the main Tivoli Enterprise Portal URL described next.

IBM Tivoli Monitoring can be accessed by other products from the Web browser by passing parameters in the URL. The list of possible parameters is not fully documented. The primary URL that is accessed by Tivoli Enterprise Portal is this URL:

```
http://<itm hostname>:1920///cnp/kdh/lib/cnp.html
```

The URL can accept standard HTML name-value pair parameters. These parameters are several of the useful parameters that can be used to access a workspace:

ip_address	Access a node with the specified IP address
hostname	Access the workspace of a node with the host name
managed_system_name	Access the workspace for the managed system name

You can use the asterisk (*) as a wildcard for specifying the arguments. When there are multiple matches for the search, you are presented with a dialog box showing the matches for you to select. These URLs are samples:

```
http://itm:1920///cnp/kdh/lib/cnp.html?ip_address=<ip address>
```

```
http://itm:1920///cnp/kdh/lib/cnp.html?hostname=<host name>*
```

```
http://itm:1920///cnp/kdh/lib/cnp.html?managed_system_name=<managed system name>
```

The managed system name is known by Tivoli applications that have access to the common data model that is generated by IBM Tivoli Monitoring Discovery Library Adapter, such as IBM Tivoli Change and Configuration Management Database, IBM Tivoli Application Dependency Discovery Manager, and IBM Tivoli Business Service Manager. The managed system name contains the node and the two character IBM Tivoli Monitoring product code. You can also obtain the list of valid managed system names from the Hub Tivoli Enterprise Monitoring Server. See Example 6-2.

Example 6-2 Getting managed system name

```
[root@itm tmsdla]# tacmd login -s localhost -u sysadmin
Password?
Validating user...
KUIC00007I: User sysadmin logged into server on https://localhost:3661.
[root@itm tmsdla]# tacmd listSystems
```

Managed System Name	Product Code	Version	Status
tivapp1	YN	06.20.04.00	N
tivapp2	YN	06.20.04.00	N
AppSrv01\$App1Serv:tivapp1:KYNS	YN	06.20.04.00	Y
taddm:ASFSDp:UAGENT00	UA	06.00.00.00	Y
taddm:ITSMERRORAPPL00	IT	06.00.00.00	Y
taddm:PERFITSMAPPL00	PE	06.00.00.00	Y
taddm:AVAILITSMAPPL00	AV	06.00.00.00	Y
taddm:PROCITSMAPPL00	PR	06.00.00.00	Y
taddm:CONFITSMAPPL00	CO	06.00.00.00	Y
itm:LZ	LZ	06.22.00.00	N
AppSrv01\$App1Serv:tivapp2:KYNS	YN	06.20.04.00	Y
app2:tivapp2:KYNA	YN	06.20.04.00	Y
tivdb:LZ	LZ	06.22.00.00	Y
taddm:UA	UM	06.22.00.00	Y
TPM:PE	PE	07.10.00.00	Y
tivapp1:LZ	LZ	06.22.00.00	Y
tivapp2:LZ	LZ	06.22.00.00	Y
tbsm:NO	NO	07.20.00.00	Y
tivapp1:KUL	UL	06.22.00.00	Y
tivdb:KUL	UL	06.22.00.00	Y
ncp_poller:tnmip:NP	NP	03.80.00.00	Y
taddm:LZ	LZ	06.22.00.00	Y
itso:tnmip:NP	NP	03.80.00.00	Y
db2inst1:tivdb:UD	UD	06.20.00.00	Y
app1:tivapp1:KYNA	YN	06.20.04.00	Y
itm:Warehouse	HD	06.22.00.00	Y
itm:SY	SY	06.22.00.00	Y
Primary:TPM:NT	NT	06.22.00.00	Y
itm:KUL	UL	06.22.00.00	Y
tivapp2:KUL	UL	06.22.00.00	Y
HUB_itm	EM	06.22.00.00	Y

You can also retrieve the managed system name from the IDML file that is generated by the tmsdla.sh script (see 5.2.4, “IBM Tivoli Monitoring integration” on page 150). In the IDML file, the launch information is stored in the following attributes for the discovered services:

► sourceContactInfo

This attribute contains the host name and the port number of Tivoli Enterprise Portal Server. It is used to build the URL by external applications:

```
<cdm:process.ManagementSoftwareSystem
sourceContactInfo="http://itm.itso.ra1.ibm.com:1920"
CDMSchemaVersion="2.3">
```

Note: The sourceContactInfo attribute must use a fully qualified host name for the single sign-on feature to work.

► sourceToken

This attribute contains information about managed systems in Tivoli Enterprise Portal Server. Example 6-3 shows sample content of the sourceToken attribute.

Example 6-3 The attribute of sourceToken

```
<cdm:sys.ComputerSystem id="9.42.171.34-ComputerSystem"
sourceToken="ip_address=9.42.171.34">
<cdm:sys.windows.WindowsOperatingSystem
id="9.42.171.34-WindowsOperatingSystem"
sourceToken="managed_system_name=Primary:TPM:NT&
object_id=p@Primary:TPM:NT">
<cdm:net.IpInterface id="9.42.171.34-IpInterface"
sourceToken="ip_address=9.42.171.34&mac_address=001641396452">
<cdm:net.IpV4Address id="9.42.171.34-IpV4Address"
sourceToken="ip_address=9.42.171.34">
<cdm:net.Fqdn id="tpm-9.42.171.34-Fqdn"
sourceToken="ip_address=9.42.171.34&mac_address=001641396452">
<cdm:app.TMSAgent id="Primary:TPM:NT-TMSAgent"
sourceToken="managed_system_name=Primary:TPM:NT&
object_id=p@Primary:TPM:NT&agent_only=false">
<cdm:sys.ComputerSystem id="9.42.171.37-ComputerSystem"
sourceToken="ip_address=9.42.171.37">
<cdm:sys.linux.Linux id="9.42.171.37-Linux"
sourceToken="managed_system_name=tivapp2:LZ&
object_id=p@tivapp2:LZ">
<cdm:sys.linux.Linux id="9.42.171.37-Linux"
sourceToken="managed_system_name=tivapp2:LZ&
object_id=p@tivapp2:LZ">
```

Based on the sourceContactInfo, the default URL path of `///cnp/kdh/lib/cnp.html?`, and the sourceToken, we can construct the complete URL to access a workspace in the Physical tree of IBM Tivoli Monitoring. For example, to access the agent in machine TPM that is highlighted in Example 6-3, you can use the URL:

```
http://itm.itso.ral.ibm.com:1920///cnp/kdh/lib/cnp.html?managed_system_name=Primary:TPM:NT&object_id=p@Primary:TPM:NT
```

There are additional parameters that can be used to access a workspace in Tivoli Enterprise Portal. Most of the parameters are in a negative numeric value.

6.2.4 Tivoli Integrated Portal

The URL for accessing products under Tivoli Integrated Portal is highly dependent on the products.

Launch to IBM Tivoli Network Manager for IP

You can configure other applications to launch to IBM Tivoli Business Service Manager using IBM Tivoli Business Service Manager-specific URLs and parameters.

Four view URLs are available to launch IBM Tivoli Network Manager for IP: Hop View, Network View, Find In View, and Structure View:

► Hop View

The Hop View provides a view of the level of connections from a particular seed device. The Hop View URL is:

```
https://TIPHOST:TIPPORT/ibm/console/ncp_topoviz/HopView.do?params
```

These Hop View parameters can be added to the end of the URL:

- seed: The name, IP address, or object ID of the seed system. This parameter is required.
- domain: The name of the IBM Tivoli Network Manager for IP domain. This parameter is required.
- hops: The number of the level of connections from the seed. The default is 1.
- layout: The layout of the view can be hierarchical with the seed at the top and the levels under it, symmetrical with the seed in the center and levels surrounding it in a circle, orthogonal with the seed in the center and levels surrounding it on a grid, or circular with the seed on the perimeter, as well as the other levels. The default is symmetrical.

- endNodes: Use the value true to display the end nodes. Use the value false to not display the end nodes. Anything that is not recognized as a network device is considered an end node. The default is true.
- connectivity: Use layer2 to display all switched connections. Use layer3 to display the IP layer. Or use ipsubnets to display the subnets. The default is ipsubnets.

Example 6-4 on page 250 illustrates URLs for the Hop View.

Example 6-4 Hop View URL

`https://tnmip.itso.ral.ibm.com:16316/ibm/console/ncp_topoviz/HopView.do?seed=tbsm.itso.ral.ibm.com&domain=ITS0`

`https://tnmip.itso.ral.ibm.com:16316/ibm/console/ncp_topoviz/HopView.do?seed=9.42.171.40&domain=ITS0&hops=3&layout=circular&endNodes=true&connectivity=layer2`

► **Network View**

The Network Views provide configurable views that can be saved and applied to certain users. Each network view has a unique ID when saved. You can find the ID by hovering over the network view name in the network view navigation tree. The Network View URL is:

`https://TIPHOST:TIPPORT/ibm/console/ncp_topoviz/NetworkView.do?id=<network view id>`

Example 6-5 illustrates a URL for a saved Network View.

Example 6-5 Network View URL

`https://tnmip.itso.ral.ibm.com:16316/ibm/console/ncp_topoviz/NetworkView.do?id=62`

► **Find in View**

The Find in View provides a list of all views for a user that contain a particular entity. Each entity has a unique entity ID. You can find this entity ID by selecting a particular device, then right-clicking, and selecting **Show Device Structure** as the attribute ENTITYID. The Find in View URL is:

`https://TIPHOST:TIPPORT/ibm/console/ncp_topoviz/FindInView.do?entityid=<entity id>`

Example 6-6 illustrates a Find in View URL for a particular entity.

Example 6-6 Find in View URL

```
https://tnmip.itso.ral.ibm.com:16316/ibm/console/ncp_topoviz/FindInView.do?entityId=11
```

► Structure View

The Structure View provides the device structure and attributes for a particular entity. You can find the entity ID in the attribute ENTITYID. The Structure View URL is:

```
https://TIPHOST:TIPPORT/ibm/console/ncp_structureview/Launch.do?entityid=<entity id>
```

Example 6-7 illustrates a Structure View URL for a particular entity.

Example 6-7 Structure View URL

```
https://tnmip.itso.ral.ibm.com:16316/ibm/console/ncp_structureview/Launch.do?entityid=11
```

Additionally, there are several Web-based tools that can be launched for IBM Tivoli Network Manager for IP:

► Simple Network Management Protocol (SNMP) Management Information Base (MIB) Browser

The SNMP MIB Browser provides the MIB values for a particular device. The SNMP MIB Browser generic URL, which will open the MIB browser without a particular device selected, is:

```
https://TIPHOST:TIPPORT/ibm/console/ncp_mibbrowser/Launch.do?
```

There are several SNMP MIB Browser optional parameters, which can be added to the generic URL:

domain	Name of IBM Tivoli Network Manager for IP domain
host	IP address of the target device
variable	Object identifier (OID) of the MIB object to query
resultsOnly	Use true to display the full MIB browser. Use false to display the MIB query results only. The default is false.

Example 6-8 illustrates a SNMP MIB Browser URL for a particular device.

Example 6-8 SNMP MIB Browser

```
https://tnmip.itso.ral.ibm.com:16316/ibm/console/ncp_mibbrowser/Launch.do?domain=ITS0&host=9.42.171.40&variable=1.3.6.1.2.1.1.1&resultsOnly=false
```

► Web Tools

Web Tools provide general, Cisco-specific, and Juniper-specific tools, which can be launched from a particular device. The Web Tools generic URL, which will open the WebTools menu, is:

`https://TIPHOST:TIPPORT/ibm/console/ncp_webtools`

Specific tools can be launched by other applications. Launch the tool in IBM Tivoli Network Manager for IP, and copy the exact URL launched.

Launch to IBM Tivoli Business Service Manager

You can configure other applications to launch to IBM Tivoli Business Service Manager using IBM Tivoli Business Service Manager-specific URLs and service context parameters.

For access to the ServiceAvailability page, the user must have the `tbsmReadOnly` role. For access to both `ServiceAvailability` and `ServiceAdministration`, the user must have the `tbsmAdminUser` role. In addition, to view the service, the user must have either the `tbsmViewService` role, the `tbsmCreateService` role, or the service level authorization. To edit the service, the user must also have the `tbsmEditService` role.

IBM Tivoli Business Service Manager URLs

IBM Tivoli Business Service Manager provides a Service Availability or Service Administration URL:

► Service Availability URL:

`https://TIPServer:TIPPort/ibm/action/launch?pageID=com.ibm.tbsm.navigationElement.desktop`

► Service Administration URL:

`https://TIPServer:TIPPort/ibm/action/launch?pageID=com.ibm.tbsm.navigationElement.serviceAdmin`

IBM Tivoli Business Service Manager Service context parameters

Optional service context parameters follow the `pageID` in the URL. Only one parameter is allowed. The first service context parameter listed in the URL is displayed, and any remaining service context parameters are ignored. You can use the following service context parameters:

► `ServiceInstanceID`

Shows the service tree starting from the specified `ServiceInstanceID`. This parameter is the root service instance for the service tree. In Tivoli Integrated Portal, this parameter is displayed under **Service Details** → **Events** tab as

the last number in the URL. Figure 6-8 on page 253 illustrates the ServiceInstanceID=71.

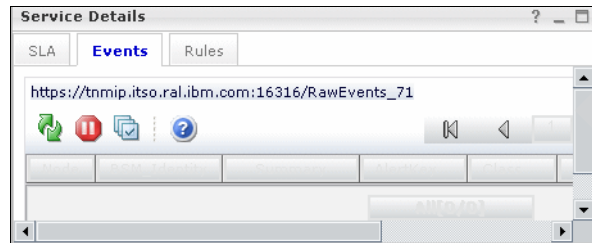


Figure 6-8 ServiceInstanceID

The URL result:

<https://tnmip.itso.ral.ibm.com:16316/ibm/action/launch?pageID=com.ibm.tbsm.navigationElement.desktop&ServiceInstanceID=71>

► ServiceInstanceName

Shows the service tree starting from the specified ServiceInstanceName. This parameter is the service name, not the display name. In Tivoli Integrated Portal, this parameter is displayed under **Service Editor** → **Edit Service** tab as the Service Name:

[https://tnmip.itso.ral.ibm.com:16316/ibm/action/launch?pageID=com.ibm.tbsm.navigationElement.desktop&ServiceInstanceName=tivdb.itso.ral.ibm.com\(3108B3FDA37132669A22838BF0441CE3\)-LinuxUnitaryComputerSystem](https://tnmip.itso.ral.ibm.com:16316/ibm/action/launch?pageID=com.ibm.tbsm.navigationElement.desktop&ServiceInstanceName=tivdb.itso.ral.ibm.com(3108B3FDA37132669A22838BF0441CE3)-LinuxUnitaryComputerSystem)

► ManagedSystemName

Shows the service tree starting from the specified IBM Tivoli Monitoring managed system. This parameter is only available for services discovered through the Discovery Library Toolkit from IBM Tivoli Monitoring. This parameter is the value of the additional attribute IBM_Tivoli_Monitoring_Services_sourceToken, which contains the managed_system_name for Tivoli Enterprise Portal. In Tivoli Integrated Portal, this parameter is displayed under **Service Editor** → **Edit Service** tab → **Additional** tab as the IBM_Tivoli_Monitoring_Service_sourceToken. If the IBM_Tivoli_Monitoring_Service_sourceToken is managed_system_name=tivapp2:LZ&object_id=p@tivapp2:LZ, the ManagedSystemName is tivapp2:LZ. The URL result:

<https://tnmip.itso.ral.ibm.com:16316/ibm/action/launch?pageID=com.ibm.tbsm.navigationElement.desktop&ManagedSystemName=tivapp2:LZ>

► GUID

Shows the service tree starting from the specified IBM Tivoli Application Dependency Discovery Manager object. This parameter is only available for services discovered through the Discovery Library Toolkit from IBM Tivoli Application Dependency Discovery Manager. In Tivoli Integrated Portal, this parameter is displayed under **Service Editor** → **Edit Service** tab → **Additional** tab as the TADDM_sourceToken:

```
https://tnmip.itso.ral.ibm.com:16316/ibm/action/1launch?pageID=com.ibm.tbsm.navigationElement.desktop&GUID=3108B3FDA37132669A22838BF0441CE3
```

► MSSName and SourceToken (both parameters are required)

Shows the service tree starting from a Discovery Library Toolkit imported IDML book. This parameter is only available for services imported through the Discovery Library Toolkit books. MSSName and SourceToken match the attributes for that object in the Discovery Library Toolkit book. If the product does not have a unique source token, the &CDMClass parameter from the Discovery Library Toolkit book is required, as well.

View parameter

Additionally, you can specify an optional View parameter. As an addition to the Default view, you can select BusinessImpact or BusinessImpactAll:

- The Default view is the relationship view. It shows the individual relationships between the services and the sub-services in a hierarchy.
- The BusinessImpact view shows only the top-level services of the model. When you select the Business Impact view for a service at any level in the service model, the service at the top level of the model is displayed in the Service Viewer.
- The BusinessImpactAll view shows a single branch of service relationships in a service model. When you select the Business Impact All view for a service at any level in the service model, a single branch of the model is displayed in the Service Viewer from bottom to top. The service that you selected is at the bottom level when you use the Business Impact All view.

6.2.5 IBM Tivoli Netcool/Webtop Active Event List

You can use transient entities to apply filters and views to an Active Event List launched from a Web page without having to formally define an entity in IBM Tivoli Netcool/Webtop. The URL can be defined this way:

```
Protocol://server:port/ibm/console/webtop/AELView?filter=filter&view=viewname&datasource=datasourcename
```

where:

filter	is an SQL filter string
viewname	is a predefined view
datasourcename	is an Object Server data source defined in the IBM Tivoli Netcool/Webtop data source configuration file. If you do not add a data source entry to the query-string, the default data source is used.

For this method to work, you must ensure that the `aelview.queries.enabled` property in `server.init` is set to true. You can apply an existing entity directly to an Active Event List through the browser address bar by entering a URL containing a query-string of the following format:

`Protocol://server:port/ibm/console/webtop/AELView?entity=entityname&datasource=datasourcename`

where:

Entityname	is a predefined entity created in IBM Tivoli Netcool/Webtop
datasourcename	is an Object Server data source defined in the IBM Tivoli Netcool/Webtop data source configuration file. If you do not add a datasource entry to the query-string, the default data source is used.

There is a set of characters that you cannot use when naming any IBM Tivoli Netcool/Webtop objects, such as filters, views, or entities. These illegal characters are defined in the `install_dir/profiles/TIPProfile/etc/webtop/illegalChar.prop` file.

6.2.6 IBM Tivoli Netcool/Impact operator view

The Impact operator view can be launched by the appropriate user using `http://hostname:9080/opview`.

6.2.7 IBM Tivoli Storage Productivity Center for Data

The Java Web Start for IBM Tivoli Storage Productivity Center can be launched from any Web-based application. The generic URL format is:

`https://device_server:device_port/ITSRM/jsp/jnlp/tpcgui.jsp?consoleClass=com.ibm.usmi.console.lic.tpc.LICConsole&-function=<functionname>&<arguments>`

The data server port value is 9549. The port value for the Device server is 9551 for https and 9550 for http.

Use these generic arguments:

-user	The user ID to log on to the Tivoli Storage Productivity Center server
-passwd	The valid password to log on to the Tivoli Storage Productivity Center server
-encpasswd	Encrypted password, which can be created using TPC-HOME/data/server/tools/tptool encrypt text password
-encmethod	The encryption method used to encrypt the password. The password can be encrypted using Tivoli Storage Productivity Center or Tivoli Enterprise Portal. The preferred method is to use Tivoli Storage Productivity Center to encrypt the password. If encpasswd is selected, you must also select encmethod with a value.
-function	The task to perform on the remote data server or the window name to open on the target system
-server	The Tivoli Storage Productivity Center server to log on to. This argument specifies a single server in the format of <i>server:port</i> .
-servers	The Tivoli Storage Productivity Center server or servers to log on to, in the format <i>server:port</i> . You can specify multiple server ports separated by commas.

The function argument can be either a task or a window name. The following list includes the possible task names:

- Create a volume (disk.create_volume); use these arguments:

-subsystem_id	The ID of the system on which the volume will be created. The subsystem_id is the name of the system as seen in the IBM® Tivoli® Storage Productivity Center graphical user interface (GUI). This parameter is required.
-name	The name of the volume. The name is not valid for IBM Tivoli Storage Enterprise Storage Server® and is ignored. If more than one volume is to be created, this value is treated as a prefix.
-size	The size of the volume.

- unit** The units to be used while creating the volume. Permissible values are KB, MB, and GB. The default unit is MB.
- number** The number of volumes to be created.
- ▶ SAN Planner (disk.san_planner) arguments:
 - computer_id** The ID of the computer for which the plan is to be created. The computer_id is the name of the computer as seen in the Tivoli Storage Productivity Center GUI.
 - subsystem_id** The ID of the system for which the plan is to be created. The subsystem_id is the name of the system as seen in the Tivoli Storage Productivity Center GUI Disk Manager.
 - plan_id** The ID of the previously saved plan. When the computer_id and subsystem_id are provided, plan_id is ignored. When you open a plan in the SAN Planner using launch in context, you must enter the exact name of the plan. The value for -plan_id is case-sensitive and must match the name of the plan when it was created. For example, if you created a plan named UserPlan1, you must use -plan_id UserPlan1 in the launch in context command.
- ▶ Capacity report (data.filesystem_capacity_by_computer):
 - computer_id** The ID of the computer for which the report is to be created. If this value is not specified, the Tivoli Storage Productivity Center GUI opens the default report panel for **Data Manager → Reporting → Capacity → FileSystem Capacity → By Computer**. You must enter a value for -computer_id that matches the ID for that computer that is stored in the Tivoli Storage Productivity Center database repository. You will receive a warning message if the ID for the computer you enter in the command line does not match the ID stored in the data repository. To ensure that this report is displayed properly when using launch in context, make sure to enter an ID for the computer that matches the ID that is displayed for it through the Topology Viewer or appropriate report in the product's user interface.
- ▶ Wasted space report (TPC.reports_data.wasted_space). This report has no parameters. The Tivoli Storage Productivity Center GUI opens the default

report panel for **Data Manager** → **Reporting** → **System Reports** → **Data** → **Wasted Space**.

Table 6-4 lists the window names.

Table 6-4 Window name list

Tree node	Window name
Administrative Services → Data Sources → CIMOM Agents	datasource.cimom
Administrative Services → Data Sources → Data Agents	datasource.data
Administrative Services → Data Sources → Inband fabric Agents	datasource.inband_fabric
Administrative Services → Data Sources → Out of band fabric Agents	datasource.out_of_band_fabric
Administrative Services → Data Sources → IBM Tivoli Storage Productivity Center Servers	datasource.tpc_servers
Administrative Services → Data Sources → VMWare VI data source	datasource.vmware
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → All	alert.all
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → Alerts Directed to user	alert.alerts_directed_to_user
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → Storage Subsystem	alert.storage_subsystem
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → Computer	alert.computer
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → Disk	alert.disk
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → Filesystem	alert.filesystem
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → Directory	alert.directory
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → User	alert.user
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → OS User Group	alert.os_user_group
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → Fabric	alert.fabric

Tree node	Window name
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → Switch	alert.switch
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → Endpoint Device	alert.endpoint_device
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → External	alert.external
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → Tape Library	alert.tape_library
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → Configuration Analysis	alert.configuration_analysis
IBM Tivoli Storage Productivity Center → Alerting → Alert Log → Hypervisor	alert.hypervisor
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Data → Disk Space Summary	data_report.disk_capacity
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Data → Storage Access Times	data_report.storage_access_times
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Data → Most Obsolete Files	data_report.most_obsolete_files
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Data → Storage Availability	data_report.storage_availability
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Data → Storage Capacity	data_report.storage_capacity
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Data → Total Freespace	data_report.total_freespace
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Data → User Space Usage	data_report.user_space_usage
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Disk → Port Performance	disk_report.port_performance
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Disk → Subsystem Performance	disk_report.subsystem_performance
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Disk → Top Volumes Data Rate Performance	disk_report.top_volumes_data_rate_performance
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Disk → Top Volumes I/O Rate Perf	disk_report.top_volumes_io_rate_performance

Tree node	Window name
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Fabric → San Assets (ALL)	fabric_report.san_assets
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Fabric → Port Connections	fabric_report.port_connections
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Fabric → Switch Performance	fabric_report.switch_performance
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Fabric → Top Switch Ports Data Rate Performance	fabric_report.top_switch_ports_data_rate_performance
IBM Tivoli Storage Productivity Center → Topology → Computers	topology.computers
IBM Tivoli Storage Productivity Center → Topology → Fabrics	topology.fabrics
IBM Tivoli Storage Productivity Center → Topology → Switches	topology.switches
IBM Tivoli Storage Productivity Center → Topology → Storage	topology.storage
IBM Tivoli Storage Productivity Center → Topology → Storage resource groups	topology.storage_resource_groups
IBM Tivoli Storage Productivity Center → Topology → Other	topology.other
IBM Tivoli Storage Productivity Center → Configuration Utility	config
Disk Manager → Storage Subsystems	disk.storage_subsystems
Fabric Manager → Fabrics	fabric.fabrics
Tape Manager → Tape Libraries	tape.tape_libraries
Fabric Manager → SAN Planner	disk.san_planner
Disk Manager → Storage Subsystems	disk.create_volume
Data Manager → Reporting → Capacity → Filesystem Capacity → By Computer	data.filesystem_capacity_by_computer
IBM Tivoli Storage Productivity Center → My Reports → System Reports → Data → Wasted Space	TPC.reports.data.wasted_space

6.3 Launching out capabilities

The launching out a URL can either be set from a predefined integration point of the product or it can also be customizable. In this section, we discuss several of these mechanisms in Tivoli tools for launching out URL-based applications. We discuss these topics:

- ▶ 6.3.1, “IBM Tivoli Application Dependency Discovery Manager” on page 261
- ▶ 6.3.2, “IBM Service Management products” on page 266
- ▶ 6.3.3, “IBM Tivoli Monitoring” on page 277
- ▶ 6.3.4, “IBM Tivoli Network Manager for IP” on page 286
- ▶ 6.3.5, “IBM Tivoli Business Service Manager” on page 296
- ▶ 6.3.6, “IBM Tivoli Netcool/Webtop Active Event List” on page 309
- ▶ 6.2.6, “IBM Tivoli Netcool/Impact operator view” on page 255

6.3.1 IBM Tivoli Application Dependency Discovery Manager

To launch in context from IBM Tivoli Application Dependency Discovery Manager to other Tivoli products, you have to import the configuration items from these products into IBM Tivoli Application Dependency Discovery Manager first using bulk loader. In this section, we focus on how to launch in context from IBM Tivoli Application Dependency Discovery Manager to IBM Tivoli Monitoring and IBM Tivoli Network Manager for IP after you import the CIs. For details about data integration, refer to 5.2.6, “IBM Tivoli Network Manager for IP integration” on page 163.

Launch IBM Tivoli Monitoring

In large environments, it becomes difficult to keep track of targets that are not running Tivoli Enterprise Management Agents. IBM Tivoli Monitoring V6.1 or later can use the integration between IBM Tivoli Application Dependency Discovery Manager and IBM Tivoli Monitoring to discover targets that do not have IBM Tivoli Monitoring Tivoli Enterprise Management Agents installed.

IBM Tivoli Application Dependency Discovery Manager is able to launch in context to IBM Tivoli Monitoring by using IBM Tivoli Monitoring IDML data that was previously loaded in the Discovery Library File Store. The IDML contains IBM Tivoli Monitoring information in Common Data Model (CDM) format.

After loading the IDML, IBM Tivoli Application Dependency Discovery Manager presents information about IBM Tivoli Monitoring in its Coverage console. This information contains html links to IBM Tivoli Monitoring managed systems.

The IBM Tivoli Application Dependency Discovery Manager monitoring coverage report highlights unmonitored resources by contrasting data that is extracted

from IBM Tivoli Monitoring with other discovered data. Also, you can launch IBM Tivoli Monitoring from the coverage report, as shown in Figure 6-9.

From the navigation menu, go to **Analytics** → **Monitoring Coverage Report**, and then, click the links in the Management Software System table to launch IBM Tivoli Monitoring.

Monitoring Coverage Report

Monitoring Coverage Summary

MSS: ibm-cdm:///CDMMSS/Hostname=itm.itso.ral.ibm.com+ManufacturerName=IBM+ProductName=IBM Tivoli Monitoring Services

Management Software System

productName	IBM Tivoli Monitoring Services			
MSSName	ibm-cdm:///CDMMSS/Hostname=itm.itso.ral.ibm.com+ManufacturerName=IBM+ProductName=IBM Tivoli Monitoring Services			
guid	2F21AD7B03393A03B2D652CC0C650E55			
productVersion	6.2.0			
Access Info	http://itm.itso.ral.ibm.com:1920			
	managedSystemName	softwareVersion	affinity	productCode
	HUB_itm	06.22.00	0000000080000000000000000000000040000468f0	EM
	app1:tivapp1:KYNA	06.20.04	000000000000000000000000000000000100400004Q0a7	YN
	AppSrv01\$ApplServ:tivapp1:KYNS	06.20.04	000000000000000000000000000000000200400004Q043	YN
	tbsm:NO	07.20.00	0000000000000000000000000000000004000000100000w0a7	NO
	tivdb:KUL	06.22.00	%IBM.STATIC070 000000000300000yw0a7	LUL
	db2inst1:tivdb:UD	06.20.00	000000000000000000000000W0000000000000080000yw0a7	UD
	tivapp2:KUL	06.22.00	%IBM.STATIC070 000000000300000yw0a7	LUL
	TPM:PE	07.10.00	%IBM.TPM_SERVER 000000000200000yw0a7	PE
	app2:tivapp2:KYNA	06.20.04	000000000000000000000000000000000100400004Q0a7	YN
	tivapp1:KUL	06.22.00	%IBM.STATIC070 000000000300000yw0a7	LUL
	AppSrv01\$ApplServ:tivapp2:KYNS	06.20.04	000000000000000000000000000000000200400004Q043	YN

Monitoring Coverage Report *Unix, WindowsOperatingSystem, ZOS, Linux, HpUx, Solaris, Aix, Db2Instance, OracleInstance, SqlServer, SybaseServer*

Name	Monitoring agent
tpm.itso.ral.ibm.com : WindowsOperatingSystem	Unmonitored
tivapp1.itso.ral.ibm.com : Linux	tivdb:L2
tivdb.itso.ral.ibm.com:db2inst1 : Db2Instance	Unmonitored

Report generated on 5/25/09 11:13 PM

Figure 6-9 Launch IBM Tivoli Monitoring from the Monitoring Coverage Report

After you click the IBM Tivoli Monitoring links in the table, you will see the Tivoli Enterprise Portal applet loading. Then, you enter the user name and password to access the Tivoli Enterprise Portal.

Note: IBM Tivoli Application Dependency Discovery Manager does not support generating Lightweight Third Party Authentication (LTPA) tokens, so when you launch other applications from IBM Tivoli Application Dependency Discovery Manager, you must sign in.

Launch IBM Tivoli Network Manager for IP

To enable the Launch in Context function from IBM Tivoli Application Dependency Discovery Manager to IBM Tivoli Network Manager for IP, you must perform the data integration. For details about data Integration, refer to 5.2, “Resource data integration” on page 135.

In this section, we show how to add the IBM Tivoli Network Manager for IP inventory report to IBM Tivoli Application Dependency Discovery Manager and how to launch the report:

1. Add the IBM Tivoli Network Manager for IP inventory report to IBM Tivoli Application Dependency Discovery Manager:
 - a. Ensure that the `$COLLATION_HOME` environment variable is set appropriately:

```
export COLLATION_HOME=/opt/IBM/cmdm/dist
```
 - b. Copy the `itnm_inventory.jsp` file from the IBM Tivoli Network Manager for IP server to the IBM Tivoli Application Dependency Discovery Manager machine. Copy the file from:
`$NCHOME/precision/adapters/nsp_dla/inventory` and copy the file to:
`$COLLATION_HOME/deploy-tomcat/reports/WEB-INF/view`.
 - c. Copy the `itnm_images` directory and its contents from IBM Tivoli Network Manager for IP server to IBM Tivoli Application Dependency Discovery Manager machine. Copy from the
`$NCHOME/precision/adaptors/nsp_dla/inventory/itnm_images` directory to the `$COLLATION_HOME/deploy-tomcat/images` directory.
 - d. Edit the `$COLLATION_HOME/etc/cdm/xml/reports.xml` file to add the section in Example 6-9 before the closing `</beans>` tag.

Example 6-9 IBM Tivoli Network Manager for IP Inventory report

```
<bean class="com.collation.cdm.reports.viewer.JspReportViewer"
      id="ITNMInventoryReport">
  <property name="reportGroup">
    <value>Inventory Reports</value>
  </property>
  <property name="reportName">
    <value>ITNM IP Inventory Report</value>
  </property>
  <property name="jsp">
    <value>/WEB-INF/view/itnm_inventory.jsp</value>
  </property>
</bean>
```

- e. Restart the IBM Tivoli Application Dependency Discovery Manager server.
2. View the IBM Tivoli Network Manager for IP inventory report from the domain manager.

On a Web browser, log in to the IBM Tivoli Application Dependency Discovery Manager domain manager. From the top menu, select **Analytics** → **ITNM Inventory Reports**. See Figure 6-10.

Discovery
Topology
Analytics

Tivoli
IBM®

ITNM IP Edition Inventory Report

Software Server Inventory

ITNM IP Edition Server Summary

Interactive Wizard

Hyperlink [\[View Server No.1 Resources\]](#)
Version 3.8.27
Host 9.42.171.29
Access URL https://tnmip.itso.ral.ibm.com:16316/ibm/console/ncp_topoviz/FindInView.do?entityid=

ITNM IP Inventory Report

ITNM IP Edition Resource Summary

BIRT Reports

Monitoring Coverage Report

Administration

Server No. 1 Resources (9.42.171.29) [\[Return to ITNM Server Summary\]](#)

Hyperlink	Name	Manufacturer	Type	Model	Source Token (EntityId)
[ITNM]	tpm.itso.ral.ibm.com	net-snmp	ComputerSystem	Linux running netSnmp	11&domain=ITSO
[ITNM]	admin.itso.ral.ibm.com	Unknown	IpDevice	Unknown	15&domain=ITSO
[ITNM]	ccmdb.itso.ral.ibm.com	Unknown	IpDevice	Unknown	9&domain=ITSO
[ITNM]	itm.itso.ral.ibm.com	net-snmp	ComputerSystem	Linux running netSnmp	8&domain=ITSO
[ITNM]	localhost	Unknown	IpDevice	Unknown	5&domain=ITSO
[ITNM]	security1.itso.ral.ibm.com	net-snmp	ComputerSystem	Linux running netSnmp	12&domain=ITSO
[ITNM]	taddm.itso.ral.ibm.com	net-snmp	ComputerSystem	Linux running netSnmp	7&domain=ITSO
[ITNM]	tbsm.itso.ral.ibm.com	net-snmp	ComputerSystem	Linux running netSnmp	17&domain=ITSO
[ITNM]	tivapp1.itso.ral.ibm.com	Unknown	IpDevice	Unknown	13&domain=ITSO
[ITNM]	tivapp2.itso.ral.ibm.com	Unknown	IpDevice	Unknown	18&domain=ITSO
[ITNM]	tivapp3.itso.ral.ibm.com	Unknown	IpDevice	Unknown	14&domain=ITSO
[ITNM]	tivbk1.itso.ral.ibm.com	Unknown	IpDevice	Unknown	10&domain=ITSO
[ITNM]	tivdb.itso.ral.ibm.com	Unknown	IpDevice	Unknown	16&domain=ITSO
[ITNM]	tnmip.itso.ral.ibm.com	net-snmp	ComputerSystem	Linux running netSnmp	6&domain=ITSO

This report provides a brief summary of the resources ITNM has provided to TADDM. The Hyperlink column provides a launch capability into the ITNM Topoviz GUI for the selected resource.

2009/05/25 23:17:22 1 /

Figure 6-10 IBM Tivoli Network Manager for IP inventory report

3. Launch IBM Tivoli Network Manager for IP from the inventory report.

From the IBM Tivoli Network Manager for IP inventory report, you can click the [ITNM] hyperlink to launch to IBM Tivoli Network Manager for IP:

- New browser window for each launch
- You are required to log in to IBM Tivoli Network Manager for IP using a valid user and password. Subsequent launches will use the same user ID and password.

6.3.2 IBM Service Management products

The Launch in Context application comes with the base services of IBM Tivoli Change and Configuration Management Database V7.1. It gives you the ability to launch from the IBM Tivoli Change and Configuration Management Database Web User interface into separate IBM Tivoli Application Dependency Discovery Manager views (physical infrastructure, application infrastructure, and business application) or to other Operational Management Products, such as Tivoli Provisioning Manager or IBM Tivoli Monitoring.

By assigning special launch points to users or groups, you can design the Launch in Context application according to the organization's needs. In this section, we discuss these topics:

- ▶ “IBM Tivoli Application Dependency Discovery Manager” on page 266
- ▶ “Launching Operational Management applications” on page 269

IBM Tivoli Application Dependency Discovery Manager

IBM Tivoli Change and Configuration Management Database V7.1 comes with a predefined set of default launch entries. The Launch in Context user interface can launch in context from the Actual configuration item (CI) and Authorized CI applications in the IBM Tivoli Change and Configuration Management Database Web interface directly into IBM Tivoli Application Dependency Discovery Manager.

Note: Before using the Launch in Context to IBM Tivoli Application Dependency Discovery Manager, you must perform these tasks:

- ▶ Import IBM Tivoli Application Dependency Discovery Manager data into the IBM Tivoli Change and Configuration Management Database using IBM Tivoli Integration Composer. See 5.2.1, “IBM Tivoli Integration Composer” on page 136.
- ▶ Activate and configure the Single Sign-On function. For more details, refer to 4.1.4, “Single sign-on” on page 51.

You can start the Launch in Context application from the Start Center by selecting **Go To → System Configuration → Platform Configuration → Launch in Context**. See Figure 6-11.

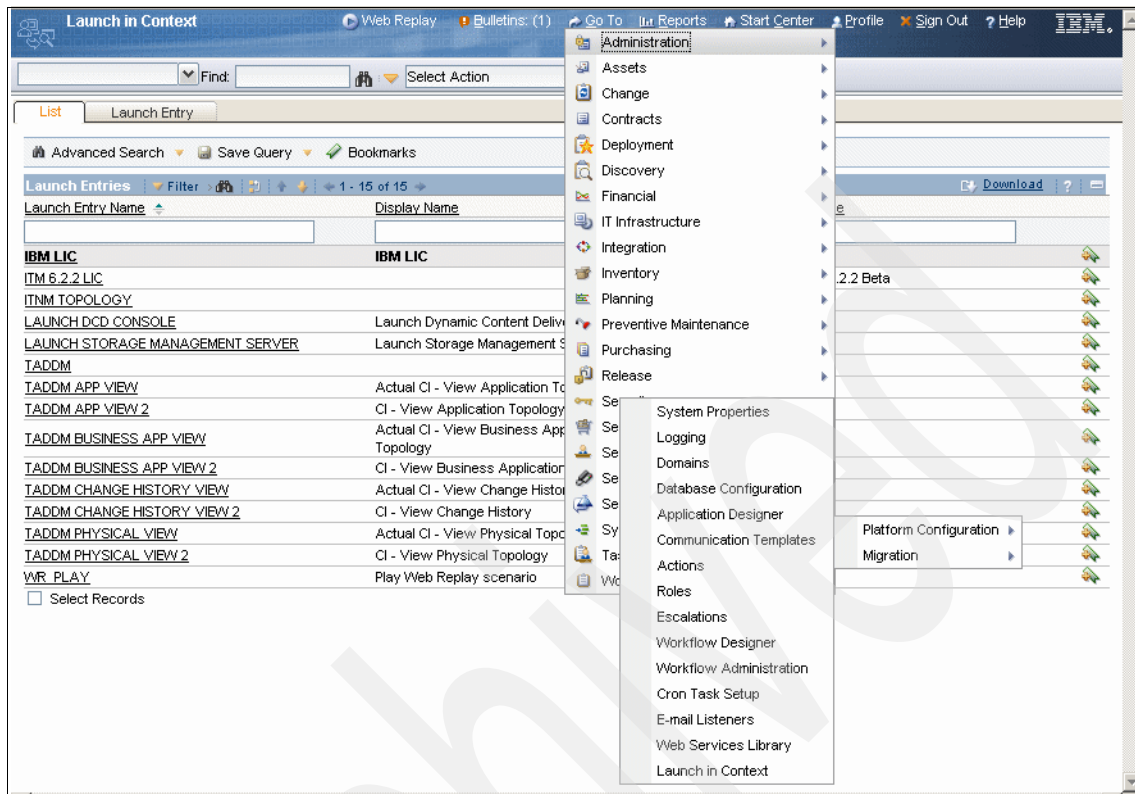


Figure 6-11 IBM Tivoli Change and Configuration Management Database Launch in Context Application

The existing launch points show on the List tab that is displayed in Figure 6-12 on page 268. These predefined launch points come with the IBM Tivoli Change and Configuration Management Database V7.1 installation.

Launch Entry Name	Display Name
IBM LIC	IBM LIC
TADDM APP VIEW	Actual CI - View Application Topology
TADDM APP VIEW 2	CI - View Application Topology
TADDM BUSINESS APP VIEW	Actual CI - View Business Application To
TADDM BUSINESS APP VIEW 2	CI - View Business Application Topology
TADDM CHANGE HISTORY VIEW	Actual CI - View Change History
TADDM CHANGE HISTORY VIEW 2	CI - View Change History
TADDM PHYSICAL VIEW	Actual CI - View Physical Topology
TADDM PHYSICAL VIEW 2	CI - View Physical Topology

Figure 6-12 List of predefined Tivoli Application Dependency Discovery Manager launch points

You can change the URL specification and parameter setting from the Launch Entry tab, as shown in Figure 6-13. For more details about IBM Tivoli Application Dependency Discovery Manager URL specifications, refer to 6.2.1, “IBM Tivoli Application Dependency Discovery Manager” on page 233.

Launch Entry Name: TADDM APP VIEW

Console URL: <http://taddm.itso.ra1.ibm.com:9430/cdm/servlet/>

Target Browser Window: TADDM

OMP Product Name:

OMP Version:

Resource Object Name	Resource Classification
ACTCI	

Figure 6-13 Editing Tivoli Application Dependency Discovery Manager launch in context URL parameters

After IBM Tivoli Application Dependency Discovery Manager data is loaded into the IBM Tivoli Change and Configuration Management Database database using the IBM Tivoli Integration Composer adapter, you can launch in context into the IBM Tivoli Application Dependency Discovery Manager application. All URLs provide the GUID parameter to show the details window.

Start the Launch in Context application from the Start Center by selecting **Go To → IT Infrastructure → Configuration Items or Actual → Select the designated Configuration Items → Select the CI from which to launch from.** And then, click **Select Action → View Actual CI → Topology Application**, as shown in Figure 6-14.

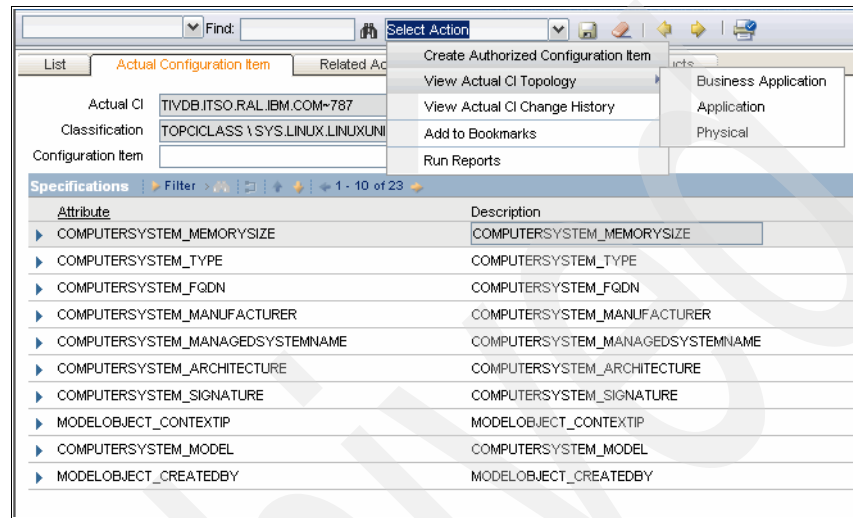


Figure 6-14 Launch Tivoli Application Dependency Discovery Manager from the Actual CI application

Launching Operational Management applications

If, in the context of a Service Management process, you must automatically link a user to the user interface of an external system to analyze specifics within the external system, consider the Launch in Context facility.

The Launch in Context application lets you define launch points to external system consoles. You can use the launch points inside specific tasks of your process flow definitions.

In this section, we describe in a step-by-step manner how to create a new launch entry for an external application, in this case, Tivoli Enterprise Portal of IBM Tivoli Monitoring V6.2.2. We implement the new launch entry as a Select Action in the Actual Configuration Items application.

To create a new launch in context entry for users, follow these required steps:

1. Define a launch entry point.

The first step is to create a new launch entry point and to define the appropriate URL specifications and parameters for the external system. Select **Go To** → **System Configuration** → **Platform Configuration** → **Launch in Context** to create a new launch in context entry. Provide the information that is shown in Figure 6-15 on page 270.

Launch in Context

Find: Select Action

List Launch Entry

Launch Entry Name: ITM 6.2.2 LIC

Console URL: http://itm.itso.ral.ibm.com:1920//cnp/kdh/lb/cnp

Target Browser Window: _use_new

OMP Product Name: Tivoli Monitoring 6.2.2 Beta

OMP Version: 6.2.2

Launch Contexts for

Resource Object Name	Resource Classification
...No rows to display...	

Figure 6-15 Create a new Launch in Context entry point

The URL can contain a variable substitution from any accessible field for the application that you will associate later. The field name must be enclosed between curly brackets {}. You retrieve the list of fields in a similar manner to the discussion in 6.2.2, “Tivoli Process Automation Engine-based products” on page 235. Special substitution exists when there is an Operational Management Product associated with the CI. The Operational Management Product (OMP) association allows you to substitute the OMP entries based on the selection from the OMP Product Name and OMP Version fields:

{reportinghostname} This is the hostname of the OMP.
{sourcetoken} This is a token that the reporting OMP identifies as the managed object identifier within the OMP.

2. Associate the Launch Entry with a Signature option.

After you set up the URL specification and parameter setting, you must associate the launch entry point to an application. You create this association with a Signature:

- a. Click **Go To** → **System Configuration** → **Platform Configuration** → **Application Designer**. In the List tab, select the application where the Launch Entry will be implemented, as shown in Figure 6-16.

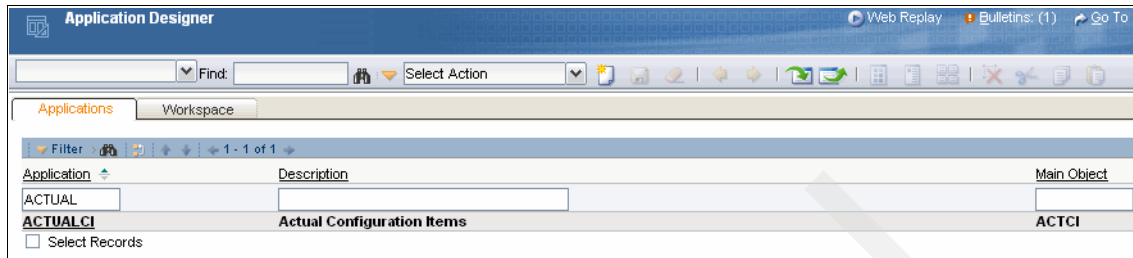


Figure 6-16 Select the application from the list

- b. Choose **Add/Modify Signature Options** from the **Select Action** list, as shown in Figure 6-17 on page 271.

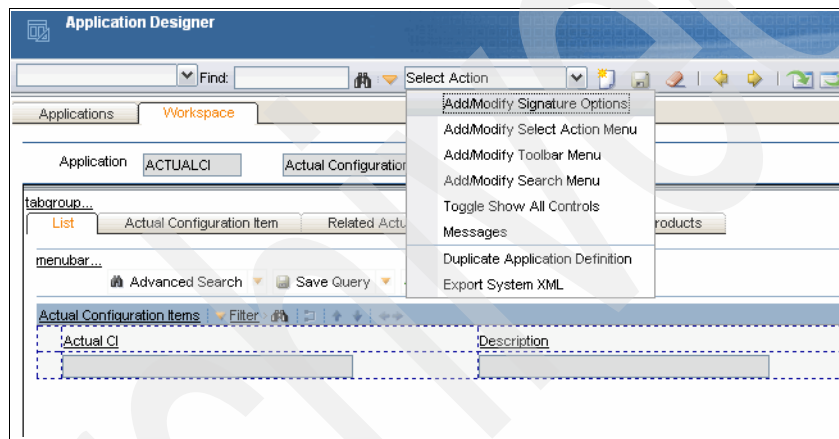


Figure 6-17 Add/Modify Signature Options

- c. Click **New Row** in the Add/Modify Signature Options dialog, and provide an option and a meaningful description, as shown in Figure 6-18 on page 272. This description will be the entry that shows in the Select Action menu.

Add/Modify Signature Options

Filter 1 - 5 of 5 Download

Option	Description
TE	
CREATECI	Create Authorized Configuration Item
CREATECIS	Create Authorized Configuration Items
CREATEREPT	Create Report Button
DELETEREPT	Delete Report Button
TEPL	Tivoli Monitoring 6.2.2

Option: TEPL
 Description: Tivoli Monitoring 6.2.2
 Option Also Grants:
 Option Also Revokes:
 Prerequisite:
 E-signature Enabled? ☐
 Visible? ☒

Advanced Signature Options

☐ None
☐ Warning appears when this action is selected from List page where multiple records are shown and no particular record is selected
☐ This is an action that must be invoked by user in the UI
☒ Associate to launch entry to enable launch in context

Launch Entry Name: ITM 6.2.2 LIC

Select Value

Filter 1 - 1 of 1 Download

Launch Entry Name	Display Name	Console URL
itm		http://itm.itso.ralibm.com:1920//cnp/kdh/lib/cnp.html?managed_system_name=TPM:PE&object_id=p@TPM:PE&agent_only=false

Cancel

Figure 6-18 Specifying the new options

Note: You must expand the **Advanced Signature Options** section to insert the appropriate launch entry point.

3. Modify the Select Action menu.

You must add the newly created Signature Option as a menu option to the application. We add the entry to the Select Action menu:

- a. Select **Go To** → **System Configuration** → **Platform Configuration** → **Application Designer**, and then, choose **Add/Modify Select Action Menu**, as shown in Figure 6-19.

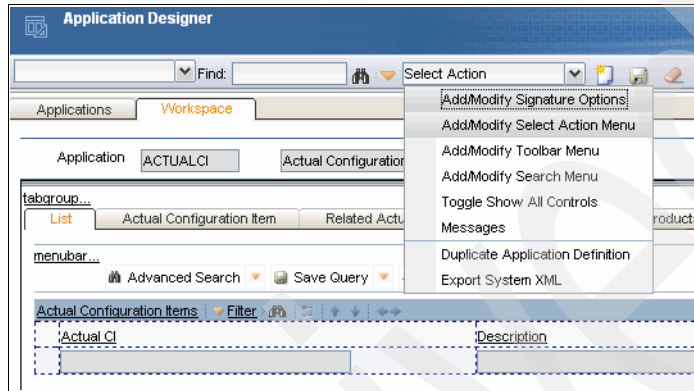


Figure 6-19 Add/Modify Select Action Menu

- b. Click **New Row**, and provide the entries that are shown in Figure 6-20 on page 274. The Key Value is the name in the Signature Option entry. The Position number is the relative item position of the Select Action menu.

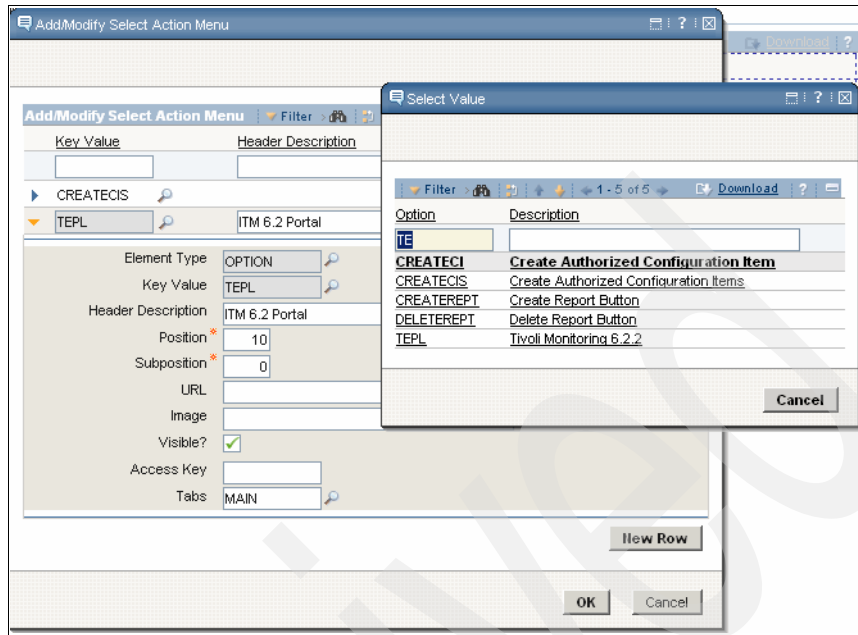


Figure 6-20 Modifying the Select Action menu

4. Allow access for users or groups by defining security.

You must adjust the security to allow users to use this launch entry point. In this example, we give access to this menu to the EVERYONE security group:

- a. Use **Go To → Security → Security Groups** to select the EVERYONE group (Figure 6-21).

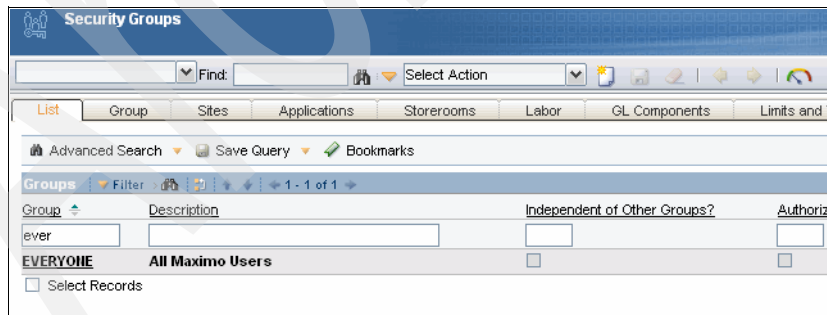


Figure 6-21 Select the group

- b. Then, from the applications list, select the application to which to add the launch in context item. In our case, it is the Actual Configuration Items application (Figure 6-22 on page 275).

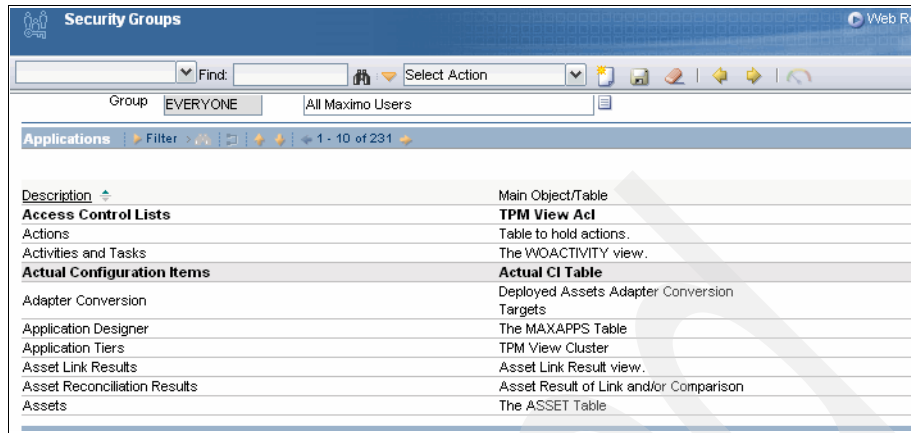


Figure 6-22 Select the application from the list

- c. From the list of available options, put a check on the newly added option. You can use the Filter row to find your launch in context option. See Figure 6-23 on page 276. Clicking **Grant Listed Options for This Application** gives access to all options.

Security Groups Web Replay Bulletins: (1) Go To Reports Start Center Profile Sign Out Help IBM

Find: Select Action

List Group Sites Applications Storerooms Labor GL Components Limits and Tolerances Data Restrictions Users

Group: **EVERYONE** All Maximo Users

Applications Filter 1 - 10 of 231

Description	Main Object/Table	Original Application (if copied)
Access Control Lists	TPM View Acl	
Actions	Table to hold actions.	
Activities and Tasks	The WVOACTIVITY view.	
Actual Configuration Items	Actual CI Table	
Adapter Conversion	Deployed Assets Adapter Conversion Targets	
Application Designer	The MAXAPPS Table	
Application Tiers	TPM View Cluster	
Asset Link Results	Asset Link Result view.	
Asset Reconciliation Results	Asset Result of Link and/or Comparison	
Assets	The ASSET Table	

Options for Actual Configuration Items Filter 1 - 2 of 2

Description	Grant Access?	Condition
Read Access to Actual CIs	☒	
Save Actual CI	☒	

Filter 1 - 10 of 14

Description	Grant Access?	Condition
Bookmarks	☒	
Clear Changes	☐	
Create Authorized Configuration Item	☐	
Create Authorized Configuration Items	☐	
Create Report Button	☐	
Delete Report Button	☐	
More Search Fields	☐	
Run Reports	☒	
Save Current Query	☐	
Tivoli Monitoring 6.2.2	☒	

Details

Description: Tivoli Monitoring 6.2.2

Grant Access? ☒

Condition:

Type:

Expression:

Condition Class:

Figure 6-23 Select option name from list

5. Verify the new launch entry.

Note: Before you begin, ensure that you can launch into the application by using a URL in a Web browser. If SSO is configured between the Change and Configuration Management Database and the external application, you are not required to enter the user and password again.

The new Launch in Context entry is visible after a new login. Sign out and sign in to the application. Go to your application and find the Launch in Context menu item in the **Select Action** menu. See Figure 6-24.

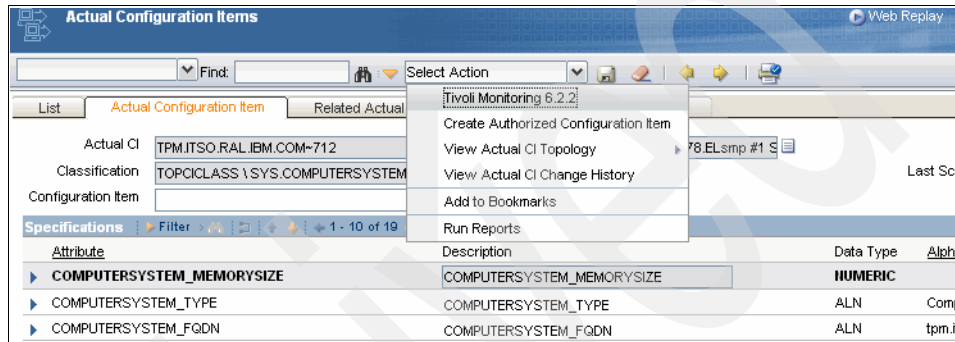


Figure 6-24 Open the Select Action menu to see the new item

6.3.3 IBM Tivoli Monitoring

In this section, we describe setting up the Launch in Context facility between IBM Tivoli Monitoring and other Tivoli products. There are various methods to launch other applications from IBM Tivoli Monitoring:

- ▶ “Application launcher” on page 277
- ▶ “Workspace browser view” on page 283
- ▶ “IBM Tivoli Provisioning Manager monitoring agent” on page 283

Application launcher

IBM Tivoli Monitoring V6.2 or later provides a generic launch facility that allows Tivoli Enterprise Portal users to launch additional applications from a Tivoli Enterprise Portal workspace. It enables you to start programs and open Web pages based on custom definitions.

This component accepts variables, making it flexible and useful to launch a target application using the context where it is launched. These variables can be passed as arguments to the target application when it is started. The variables exist in a context. The launch of applications must be originated from one of these contexts:

- ▶ Navigator item
- ▶ Row in a table view
- ▶ Data entry of a chart
- ▶ Entry in a situation event console view
- ▶ Object from a topology view

You create this launch information manually from Tivoli Enterprise Portal or by using an SQL statement to the Tivoli Enterprise Portal Server database. A predefined launcher might already be defined by the installation program. For example, IBM Tivoli Business Service Manager provides the script in Tivoli Integrated Portal server:

```
/opt/IBM/tivoli/tip/systemApps/isclite.ear/sla.war/download/launchtotbsm.sql
```

These launcher programs reside in the table called ITMUSER.KFWLAUNCH. The launch process includes the following information:

Target	Leaving this field blank uses the Tivoli Enterprise Portal browser client. Otherwise, to support a launch using Internet Explorer®, use C:\WINNT\system32\CMD.EXE. To launch using Firefox, specify the location of firefox.exe
Arguments	Click Browse to select variable substitution items to retrieve variable values at launch time. For a URL target, for Internet Explorer, select /c start C:\launchtotbsm\launchtotbsm.bat <URL>. For Firefox, select <URL>
StartIn	Path on which the launched application starts

For information and details about the Tivoli Enterprise Portal Server launch application, refer to:

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.2/launch_intro_c.htm

The following steps show how to create a launch application for IBM Tivoli Business Service Manager in Tivoli Enterprise Portal Server:

1. Log in to Tivoli Enterprise Portal Server as a user with administrative rights.
2. Click the **Administer Users** icon, select your userID, and click **Workspace Administration** in the Authorities section. See Figure 6-25 on page 279.

Make sure that **Workspace Administration Mode** option is selected. This permission allows the new Launch Application instance to be available to any Tivoli Enterprise Portal Server user.

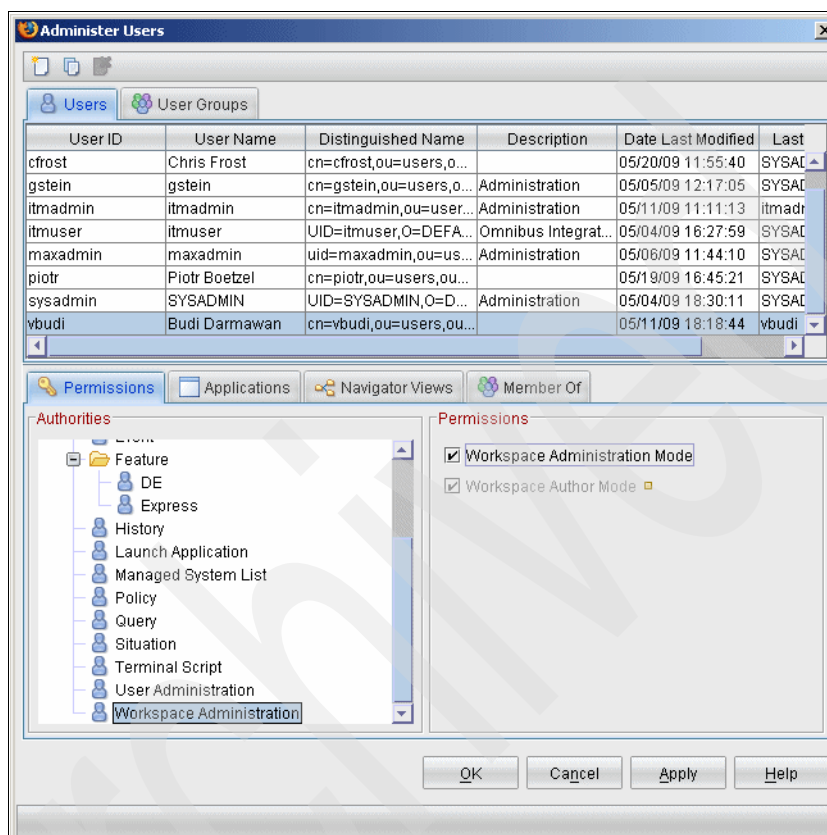


Figure 6-25 Managing user authority

3. In the Physical navigator view, select an agent, for instance, select **Enterprise** → **Linux Systems** → **tivdb.itso.ral.ibm.com** → **Linux OS**.
4. Right-click the selected agent, and choose **Launch**.

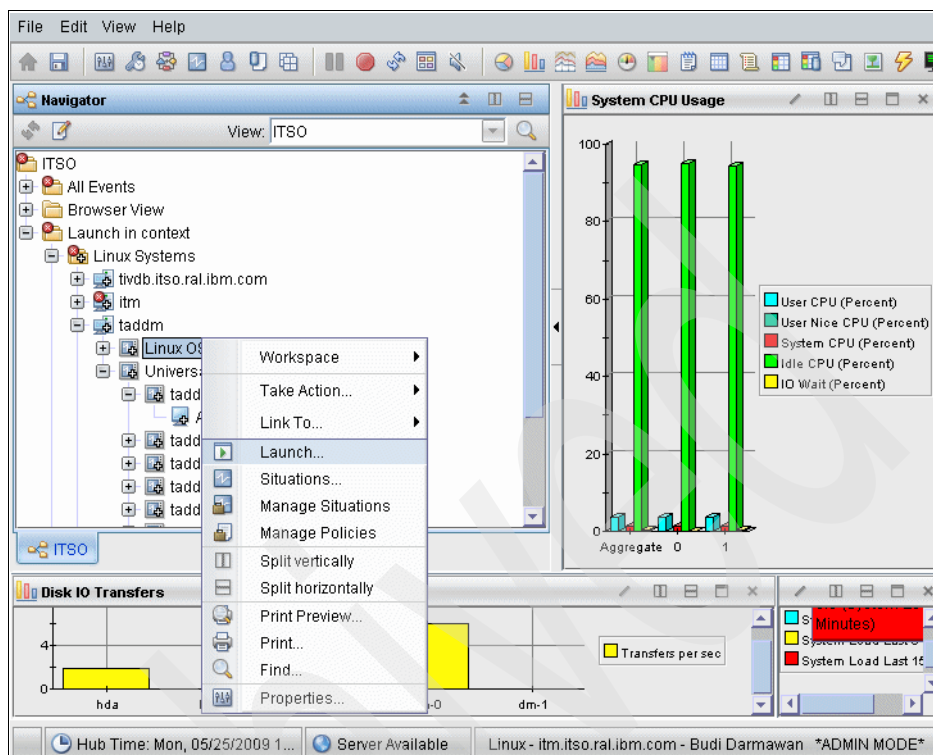


Figure 6-26 Launch in IBM Tivoli Monitoring

5. Click **Create New**. Enter a name for this new Launch definition, for example, LIC_to_TBsm. See Figure 6-27 on page 281.

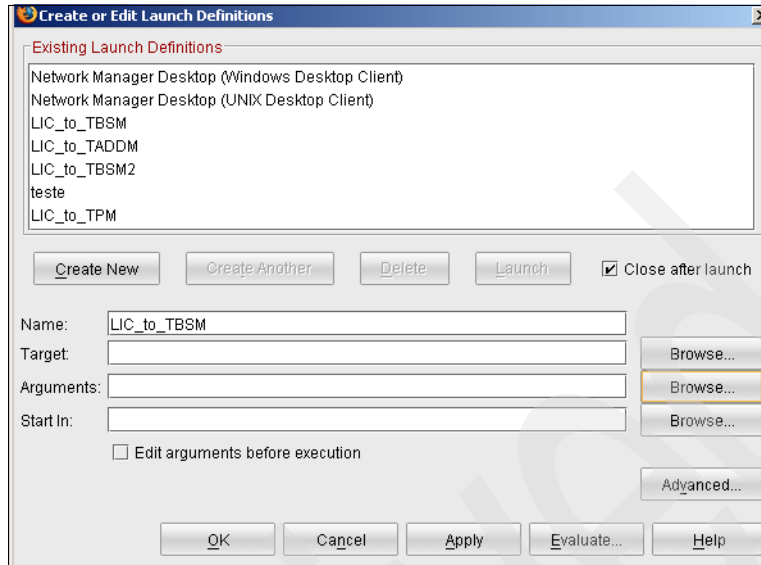


Figure 6-27 New launch entry

6. In the Arguments row, click **Browse**. See Figure 6-28. Select **Substitutable Items**, and click **OK**.

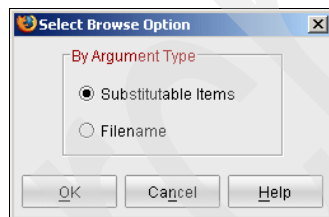


Figure 6-28 Argument type

7. Select a variable, such as **Node**, in the Selected Context tree, as shown in Figure 6-29 on page 282. Click **OK**.

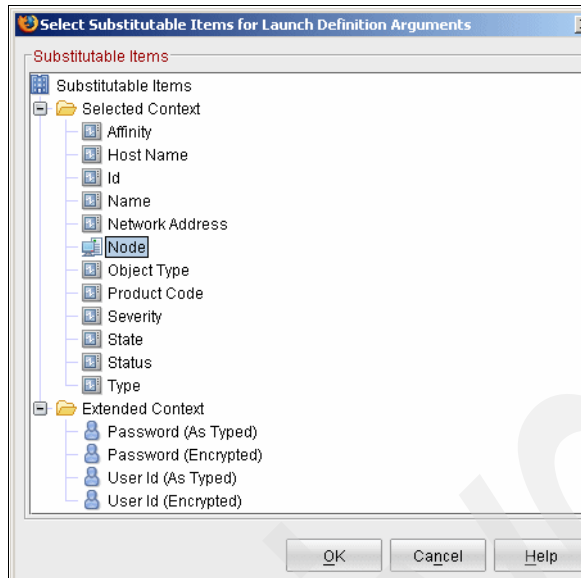


Figure 6-29 Substitution

8. The **&NODE** variable appears in the Arguments field. Add the URL body before this variable. As an example, the Arguments field has the URL:
`https://tnmip.itso.ral.ibm.com:16316/ibm/console/xLaunch.do?pageID=com.ibm.tb
sm.navigationElement.desktop&ManagedSystemName=&NODE`
9. Click **Evaluate** (see Figure 6-27 on page 281) to make sure that IBM Tivoli Monitoring is able to resolve the URL and its variables correctly. See Figure 6-30. A message might appear asking if you want to save the new URL.

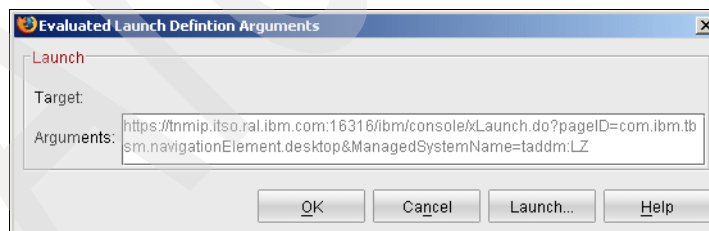


Figure 6-30 Launch definition argument

10. Click **OK** if the URL resolved correctly, and click **OK** again to finish the creation of the Launch Application definition.

For more details about the parameters that are accepted for the URL argument, refer to 6.2.4, “Tivoli Integrated Portal” on page 249.

Workspace browser view

If you know the URL address of a target Web application's URL that you want to access from IBM Tivoli Monitoring, you can load it from the browser view. It allows you to specify context variables in the address.

For example, using a browser view inside of the tivdb Linux agent workspace and specifying this URL:

`http://itm.itso.ra1.ibm.com:1920///cnp/kdh/lib/classes/candle/fw/resources/help/info.htm?$NODE$`

results in the following URL as the substitution for \$NODE\$:

`http://itm.itso.ra1.ibm.com:1920///cnp/kdh/lib/classes/candle/fw/resources/help/info.htm?tivdb:LZ`

IBM Tivoli Provisioning Manager monitoring agent

After IBM Tivoli Monitoring Agent for IBM Tivoli Provisioning Manager has successfully connected to Tivoli Enterprise Monitoring Server, we can see the Tivoli Enterprise Portal Server Workspace, as shown in Figure 6-31.

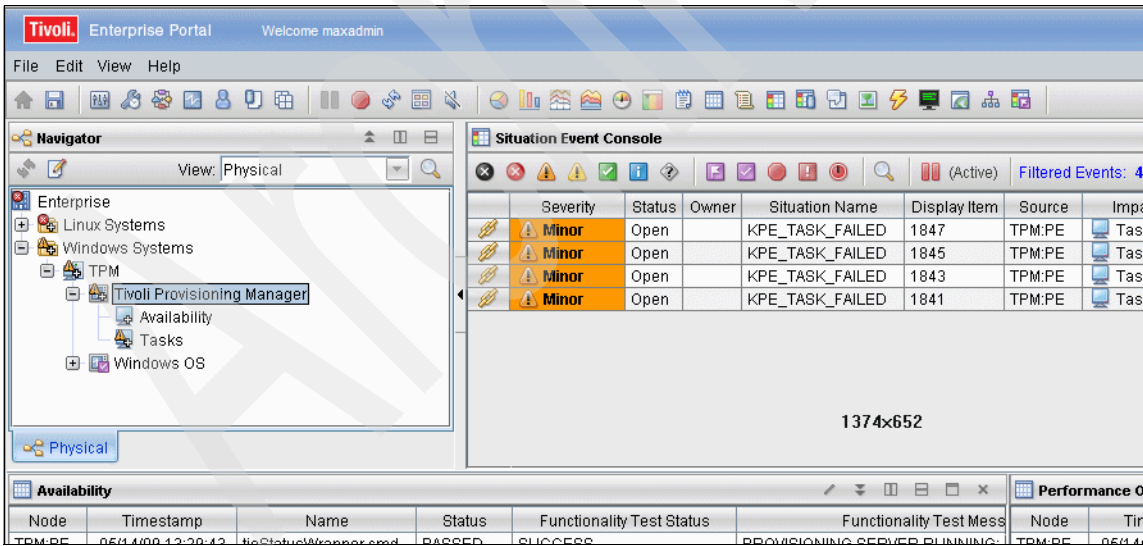


Figure 6-31 IBM Tivoli Monitoring Agent for IBM Tivoli Provisioning Manager workspace

To successfully launch in context, we assume that SSO between IBM Tivoli Monitoring and IBM Tivoli Provisioning Manager has been established, as discussed in 4.7, “IBM Tivoli Monitoring” on page 120 and in 4.4, “Tivoli Process Automation Engine security setup” on page 73. First, we configure the workspace:

1. Click **Tasks** in the Navigator tree to open the Tasks workspace. Figure 6-32 shows the Tasks workspace.

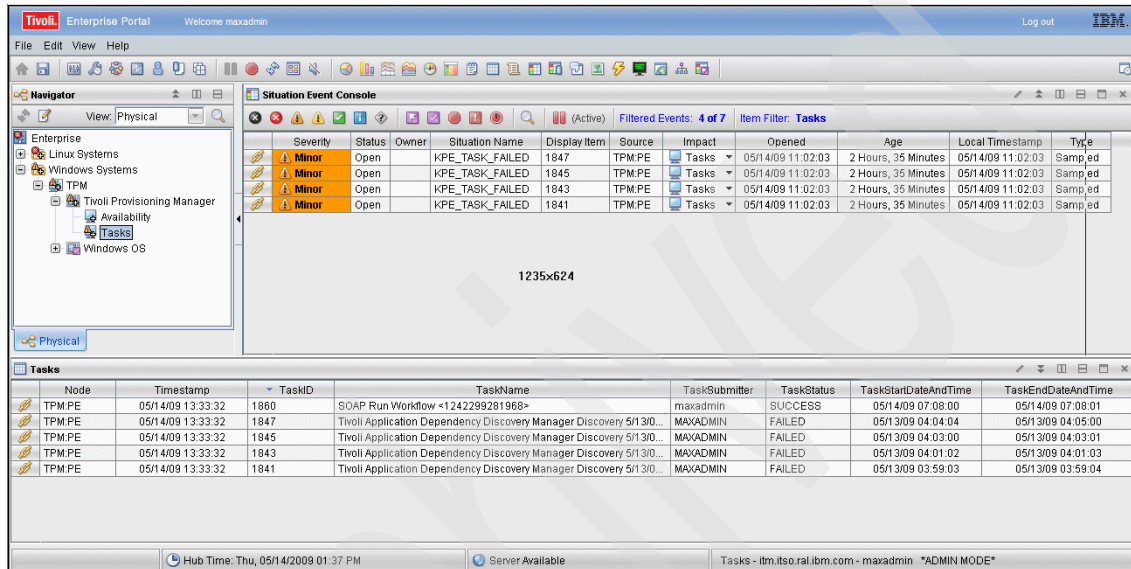


Figure 6-32 Tasks workspace

2. Right-click **Tasks** → **Workspace** → **Task Details**.
3. On the toolbar menu, click the third icon from the left. It is the button for **Properties**.
4. On the Task Details Properties panel, click **Provisioning Task Tracking** under the Browser Views from the navigation tree on the left.
5. In the Options section, change the Home variable to:

```
https://<fqdn_tpm_hostname>:9045/maximo/ui/?event=loadapp&value=tptask&uniqueid=$TPMTaskID$
```

In our environment, the value becomes:

```
https://tpm.itso.ral.ibm.com:9045/maximo/ui/?event=loadapp&value=tptask&uniqueid=$TPMTaskID$
```

Figure 6-33 on page 285 shows the final Provisioning Task Tracking configuration.

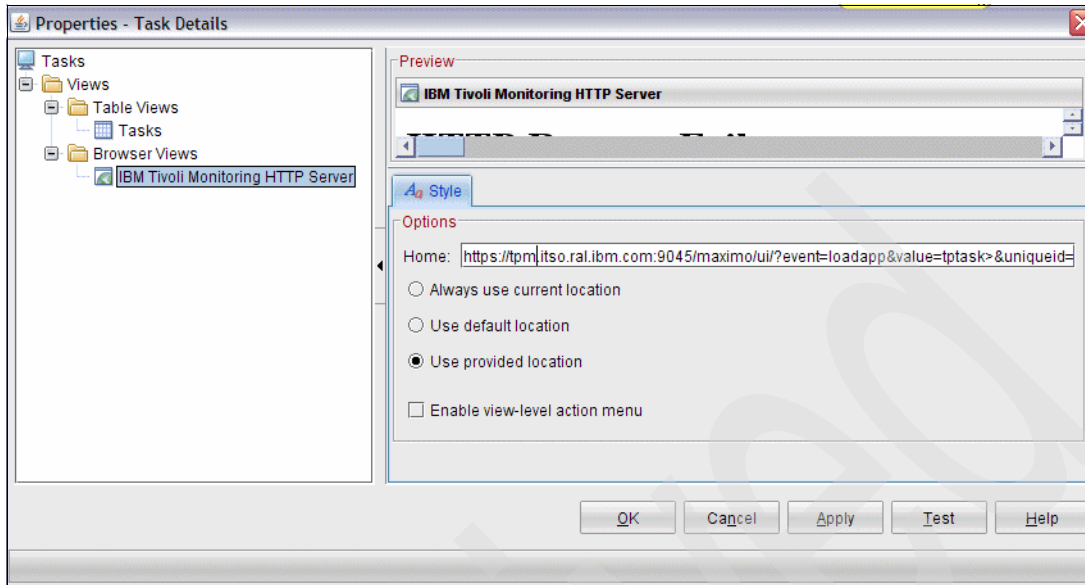


Figure 6-33 Provisioning Task Tracking configuration

Note: At this moment, we cannot use the \$TPServer\$ variable in the URL, because at IBM Tivoli Provisioning Manager V7.1, the variable is not populated from the Task Details workspace. It is only populated from the Task table view.

6. Click **OK** to save the changes.
7. On the Task Details workspace, click the chain link of one task in the Tasks table. It will load the detail of that particular Task and show it in the Provisioning Task Tracking browser view, as shown in Figure 6-34 on page 286.

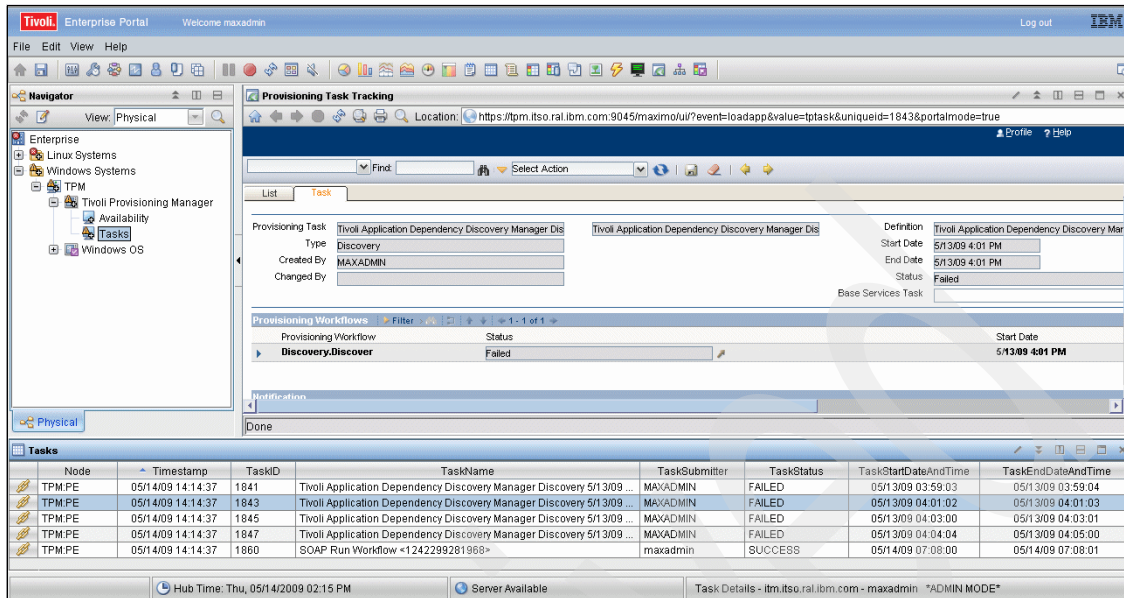


Figure 6-34 Task detail view

6.3.4 IBM Tivoli Network Manager for IP

IBM Tivoli Network Manager for IP includes a launch in context integration for IBM Tivoli Application Dependency Discovery Manager (IBM Tivoli Change and Configuration Management Database), which is enabled when configured. IBM Tivoli Network Manager for IP can launch a generic URL to external Web sites or other applications. To demonstrate the generic URL launch to applications, we configure contextualized launch to Tivoli Enterprise Portal and IBM Tivoli Provisioning Manager.

You can configure IBM Tivoli Network Manager for IP to launch to a generic URL with specific parameters, which can be used to launch to other applications.

IBM Tivoli Network Manager for IP menu structure

The menu structure in the context of IBM Tivoli Network Manager for IP is dynamic. You can add an additional menu item in the object context menu by editing the menu's XML file.

The menu file resides under the \$NCHOME/tip/profiles/TIPProfile/etc/tnm/menus directory. You can either define a submenu or a single menu level for the menu. Each menu entry refers to another XML file. The URL menu entry allows you to pass a set of name-value

pair parameters. As an example, we demonstrate the implementation of launching IBM Tivoli Monitoring and IBM Tivoli Change and Configuration Management Database.

You can define a more complex interaction with customized URLs if, for example, the target URL has a complex argument that cannot be fulfilled by a simple name-value pair from the available data. The approach that we choose is to define a custom Common Gateway Interface (CGI) application that evaluates and reformats the URL and redirects the browser to open the final page.

Use the CGI implementation to assign a parameter value that is a composite for several values. The standard URL only allows a value that is retrieved from a single field. IBM Tivoli Provisioning Manager integration requires a composite parameter value as discussed in “Launching IBM Tivoli Provisioning Manager” on page 293.

See the *IBM Tivoli Network Manager for IP Administration Guide* for more details:

http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?topic=/com.ibm.networkmanagerip.doc_3.8/itnm/ip/3.8/admin/reference/nmip_admin_toolleg.html

Launching a URL application

Prior to being able to launch a menu, data integration is needed to acquire IBM Tivoli Application Dependency Discovery Manager GUID information. See 5.2.6, “IBM Tivoli Network Manager for IP integration” on page 163. For IBM Tivoli Monitoring, we use the IP address as the context information.

To launching a URL application, you perform the following tasks:

1. Define the launch menu item in an existing context menu. We decide to add a submenu in the IBM Tivoli Network Manager for IP topology visualization context menu for external launching:
 - a. The context menu resides in the `$NCHOME/tip/profiles/TIPProfile/etc/tnm/menus/` directory.
 - b. Edit the `ncp_topoviz_device_menu.xml` file, which defines the device's context menus.
 - c. Add a menu entry at the end, such as `<menu id="ncp_wt_launch"/>`. This menu entry must be within the definition block. You can also add a separator using the `<separator/>` definition.
2. Create a second-level menu (a submenu):
 - a. Because this submenu is another menu, the submenu definition file also resides in the `$NCHOME/tip/profiles/TIPProfile/etc/tnm/menus/` directory.

- b. Create a file called `ncp_wt_launch.xml`, which is based on the menu name that was created in step c on page 287.
- c. The content of the submenu xml file is shown in Example 6-10. The entity type represents the entities that will be active on this submenu. We define a launch to IBM Tivoli Application Dependency Discovery Manager, IBM Tivoli Monitoring, and IBM Tivoli Provisioning Manager.

Note: We define the IBM Tivoli Provisioning Manager menu item later in “Launching IBM Tivoli Provisioning Manager” on page 293.

Example 6-10 Submenu items ncp_wt_launch.xml

```
<ncp_menu id="ncp_wt_launch" label="Launch Tools...">
  <context>
    <attribute id="entityType">
      <equals value="1"/>
      <equals value="8"/>
    </attribute>
  </context>
  <definition>
    <tool id="ncp_wt_taddm_details"/>
    <tool id="ncp_wt_itm_tep"/>
    <tool id="ncp_wt_tpm"/>
  </definition>
</ncp_menu>
```

3. Define the menu item definition for IBM Tivoli Application Dependency Discovery Manager:
 - a. Create a tool definition under the `$NCHOME/tip/profiles/TIPProfile/etc/tnm/tools/` directory.
 - b. The filename from Example 6-10 is `ncp_wt_taddm_details.xml`. The content is shown in Example 6-11.

Example 6-11 ncp_wt_taddm_details.xml

```
<ncp_tool id="ncp_wt_taddm_details" key="ncp_wt_taddm_details"
  label="View Details" type="url" runOnList="true" runForEach="false">
  <url
    value="http://taddm.itso.ral.ibm.com:9430/cdm/servlet/LICServlet"
    target="_blank" windowFeatures="ScrollBars=yes,Resizable=yes"
    method="GET">
    <parameter name="guid" valueType="ncim"
      table="entityGUIDCache" column="guid" runOnMainNode="true"/>
    <parameter name="console" valueType="text" text="web"/>
```

```

        <parameter name="username" valueType="text"
            text="administrator"/>
        <parameter name="password" valueType="text" text="collation"/>
    </url>
</ncp_tool>

```

- c. Specify the IBM Tivoli Application Dependency Discovery Manager server name, port, user, and password in the XML file.
- d. Define the name-value pair of the arguments as discussed in 6.2.1, “IBM Tivoli Application Dependency Discovery Manager” on page 233. The parameters are defined in the following ways:

name: This parameter is the variable name appended to the URL.

valueType: This parameter shows from where the variable value will be retrieved. This parameter can have one of the following values:

ncim	Column from the IBM Tivoli Netcool/Impact (NCIM) database. If this parameter is specified, the table name, column name, and RunOnMainNode parameters must be specified. For example, valueType="ncim" table="chassis" column="accessIPAddress" runOnMainNode="true"
domainName	IBM Tivoli Network Manager for IP domain that contains the device
webtopDataSource	Data source name that contains the device
cookie	This parameter obtains the value of the cookie. If this parameter is specified, the cookieName parameter must also be specified. For example, valueType="cookie" cookieName="userId"
text	Value is in plain text. If this parameter is specified, the text parameter must also be specified. For example, valueType="text" text="3"

4. Define the menu item definition for IBM Tivoli Monitoring. This definition is similar to the IBM Tivoli Application Dependency Discovery Manager definition. The menu item (tool) definition is shown in Example 6-12.

Example 6-12 ncp_wt_itm_tep.xml

```
<ncp_tool id="ncp_wt_itm_tep" key="ncp_wt_itm_tep"
  label="Launch TEP (ITM)" type="url" runOnList="true"
  runForEach="false">
  <url

value="http://itm.itso.ral.ibm.com:1920///cnp/kdh/lib/cnp.html?"
  target="_blank" windowFeatures="ScrollBars=yes,Resizable=yes"
  method="GET">
    <parameter name="ip_address" valueType="ncim" table="chassis"
      column="accessIPaddress" runOnMainNode="true"/>
  </url>
</ncp_tool>
```

5. Log in to Tivoli Integrated Portal. Select **Availability** → **Network Availability** → **Hop View** or **Network View**.
6. Select a resource to view in IBM Tivoli Application Dependency Discovery Manager, and right-click to select **Launch Tools** → **View Details**. Figure 6-35 illustrates the IBM Tivoli Application Dependency Discovery Manager tools.

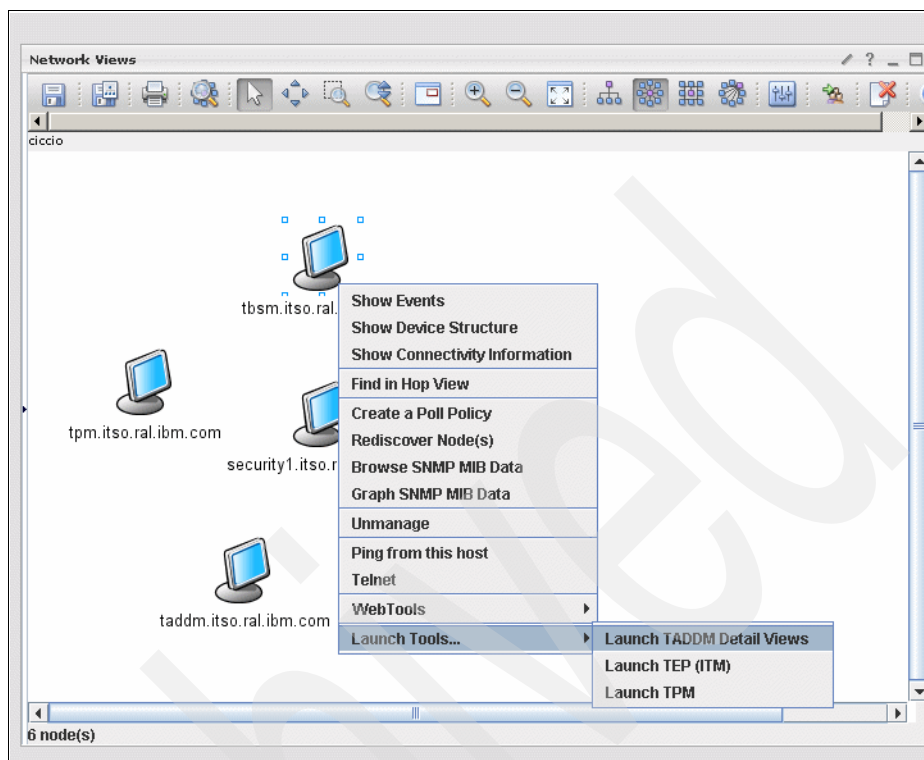


Figure 6-35 IBM Tivoli Application Dependency Discovery Manager tools

Figure 6-36 on page 292 illustrates the IBM Tivoli Application Dependency Discovery Manager details for the selected system.

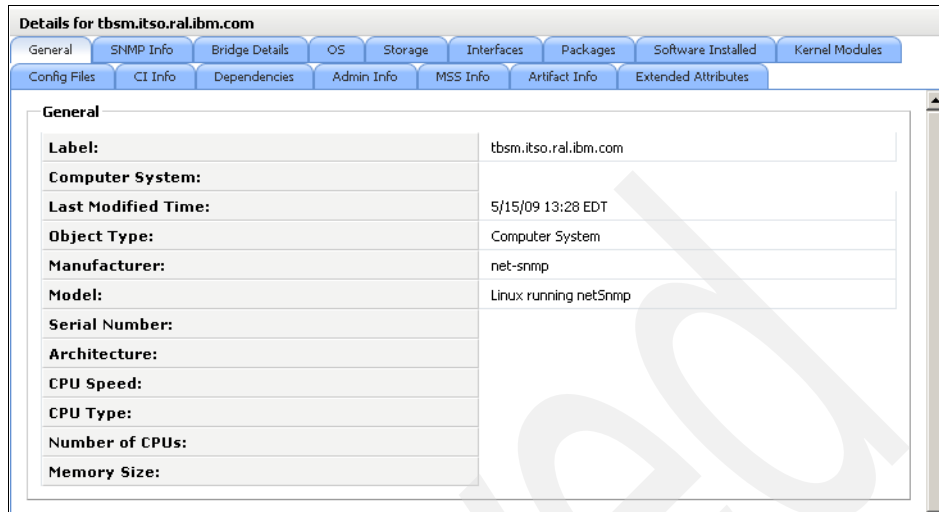


Figure 6-36 IBM Tivoli Application Dependency Discovery Manager details

- Right-click a resource to view in IBM Tivoli Network Manager for IP, and select **Launch Tools** → **Launch TEP (ITM)**. Figure 6-37 illustrates the Tivoli Enterprise Portal.

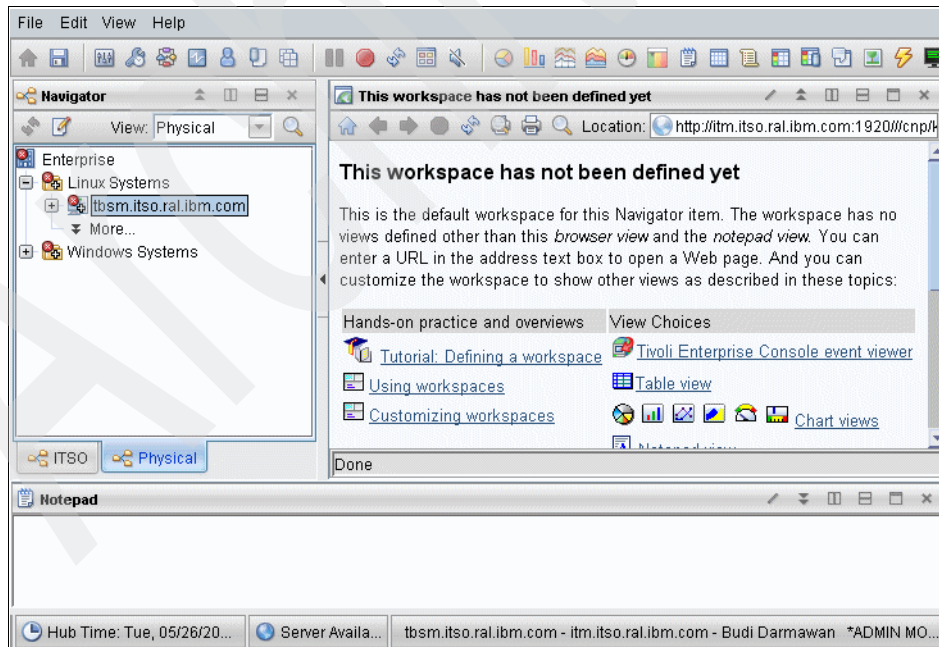


Figure 6-37 Tivoli Enterprise Portal

Launching IBM Tivoli Provisioning Manager

Because the URL format of IBM Tivoli Provisioning Manager is more complex than launching to Tivoli Enterprise Portal, the IBM Tivoli Network Manager for IP launch in context to IBM Tivoli Provisioning Manager must use a CGI script that interprets and opens the IBM Tivoli Provisioning Manager URL.

The URL format that we use is:

```
http://<tpmhost>/maximo/ui/?event=loadapp&value=tpservers&additionalevent=useqbe&additionaleventvalue=Name=<systemname>
```

The system name is retrieved from IBM Tivoli Network Manager for IP, but the `additionaleventvalue` parameter requires the whole value of `Name=<systemname>`. However, IBM Tivoli Network Manager for IP does not allow a concatenation of arguments.

The Launch in Context facility launches a URL tool, which passes the system name parameter to a CGI script. The CGI script receives the system name parameter and passes it to IBM Tivoli Provisioning Manager.

The following steps illustrate creating CGI tools in addition to IBM Tivoli Network Manager for IP menus and URL tools:

1. Assume that the additional menu options have been built from the steps in “Launching a URL application” on page 287.
2. We develop a CGI script that receives the system name, and we construct the URL for launching IBM Tivoli Provisioning Manager. The CGI script resides in the `$TIP_HOME/profiles/TIPProfile/etc/webtop/cgi-bin` directory. The script is called `launch_tpm.cgi`. Ensure that the file permission is 755. Example 6-13 lists the `launch_tpm.cgi` script.

Example 6-13 The launch_tpm.cgi script

```
#!/usr/bin/perl
$buffer = $ENV{'QUERY_STRING'};
@pairs = split(/&/, $buffer);

foreach $pair (@pairs)
{
    ($name, $value) = split(/=/, $pair);
    $value =~ tr/+/ /;
    $value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
    $name =~ tr/+/ /;
    $name =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
    $FORM{$name} = $value;
}
$sysName = "sysName";
```

```

print STDOUT "Content-type: text/html\n\n";
print STDOUT "<html>\n";

$serviceName = $FORM{"\"$sysName"};

# Define universal parts of the URL
$URL = 'https://tpm.itso.ral.ibm.com:9045/maximo/ui/?'.
    'event=loadapp&value=tpservers&additionalevent=useqbe&'.
    'additionaleventvalue=Name%3d'.'$value';
print STDOUT "<head>\n";
print STDOUT "<meta http-equiv=\"Refresh\" \n\n";
print STDOUT "content=\"1;url=$URL\">\n";
print STDOUT "<title>Connecting to TPM ...</title>\n";
print STDOUT "</head>\n";
print STDOUT "<body style=\"background-color: $Background_color; ";
print STDOUT "color: $Text_color\">\n";
print STDOUT "<h2>Connecting to TPM ...</h2>\n";
print STDOUT "<p>\".$sysName.\"+\".$serviceName.\"+\".$value;";
print STDOUT "</body></html>\n";

```

3. Register the newly created CGI script with IBM Tivoli Netcool/Webtop:
 - a. Log in to Tivoli Integrated Portal as the IBM Tivoli Netcool/Webtop administrative user.
 - b. Select the IBM Tivoli Netcool/Webtop view. Select **Administration** → **Event Management Tools** → **CGI Registry**.
 - c. Click **Register**. Enter a name, and enter the file name of launch_tpm.cgi from the \$TIP_HOME/profiles/TIPProfile/etc/webtop/cgi-bin directory. See Figure 6-38 on page 295.

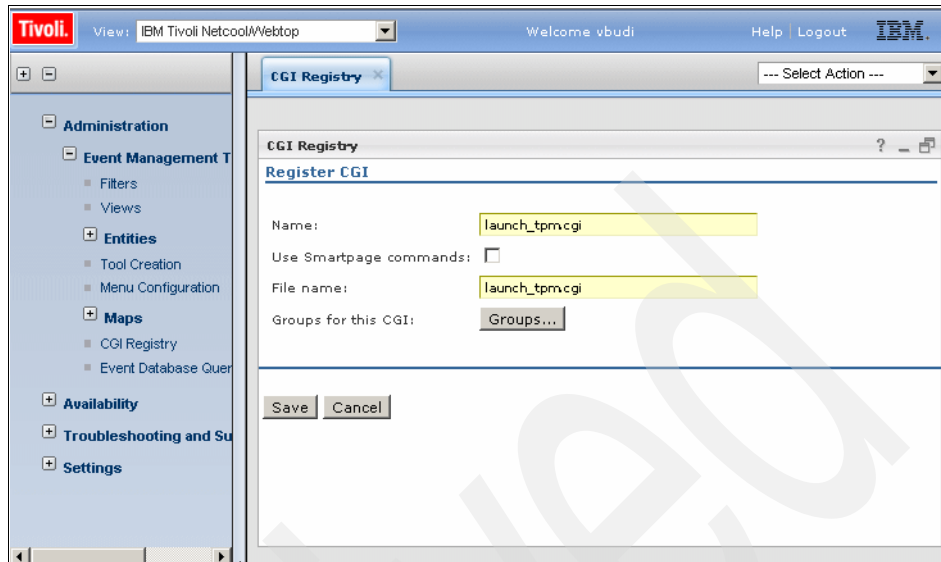


Figure 6-38 CGI Registry

- d. Click **Groups**, and select group access if required.
 - e. Select **Save**.
4. Create a URL tool to execute the CGI script in the \$TIP_HOME/profiles/TIPProfile/etc/tnm/tools/ directory. The resulting ncp_wt_tpm.xml is shown in Example 6-14.

Example 6-14 ncp_wt_tpm.xml

```
<ncp_tool id="ncp_wt_tpm" key="ncp_wt_tpm" label="Launch TPM"
  type="url" runOnList="true" runForEach="false">
  <url
    value="https://tnmip.itso.ral.ibm.com:16316/ibm/console/
webtop/cgi-bin/launch_tpm.cgi?" target="_blank"
windowFeatures="ScrollBars=yes,Resizable=yes" method="GET">
    <parameter name="sysName" valueType="ncim" table="chassis"
      column="sysName" runOnMainNode="true"/>
  </url>
</ncp_tool>
```

5. In Tivoli Integrated Portal, select a system in IBM Tivoli Provisioning Manager, right-click it, and select **Launch Tools** → **Launch TPM**.
6. If single sign-on is enabled between IBM Tivoli Network Manager for IP and IBM Tivoli Provisioning Manager, it will not prompt for a user ID and password; otherwise, enter an IBM Tivoli Provisioning Manager user ID and

password. IBM Tivoli Provisioning Manager is displayed with the system information.

Figure 6-39 shows the IBM Tivoli Provisioning Manager window for the specified device.

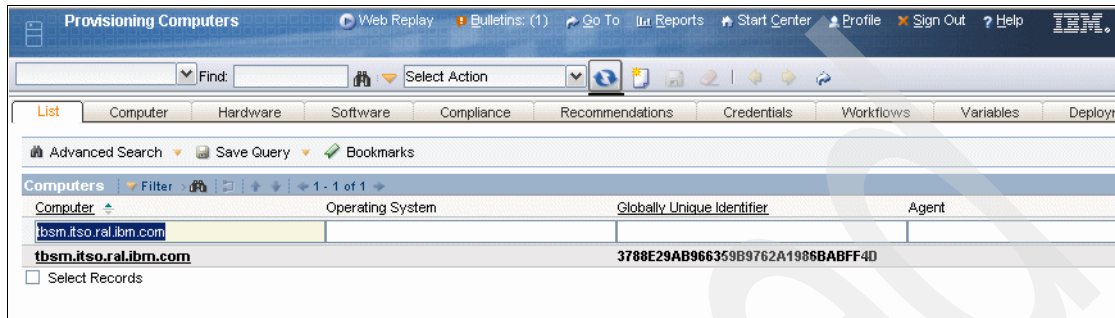


Figure 6-39 IBM Tivoli Provisioning Manager

6.3.5 IBM Tivoli Business Service Manager

IBM Tivoli Business Service Manager can launch from its console several graphical user interfaces of other IBM Tivoli products. By default, it provides launch capability to the following consoles:

- ▶ IBM Tivoli Application Dependency Discovery Manager Version 7.1
- ▶ IBM Tivoli Monitoring Version 6.2.1
- ▶ IBM Tivoli Network Manager for IP Version 3.8
- ▶ IBM Tivoli Change and Configuration Management Database Versions 7.1 and 7.1.1
- ▶ IBM Tivoli Netcool/Webtop Version 2.2

Figure 6-40 on page 297 shows the launch submenu. These launch definitions are stored in a custom canvas definition XML file within the data server:
\$TIP_HOME/profiles/TBSMProfile/installedApps/TBSMCell/tbsm.ear/sla.war/
av/xmlconfig/canvasOpenURLActions.xml

You can obtain additional integration information at this Web site:

<https://www.ibm.com/developerworks/wikis/display/tivolibsm/Integration+Scenarios>

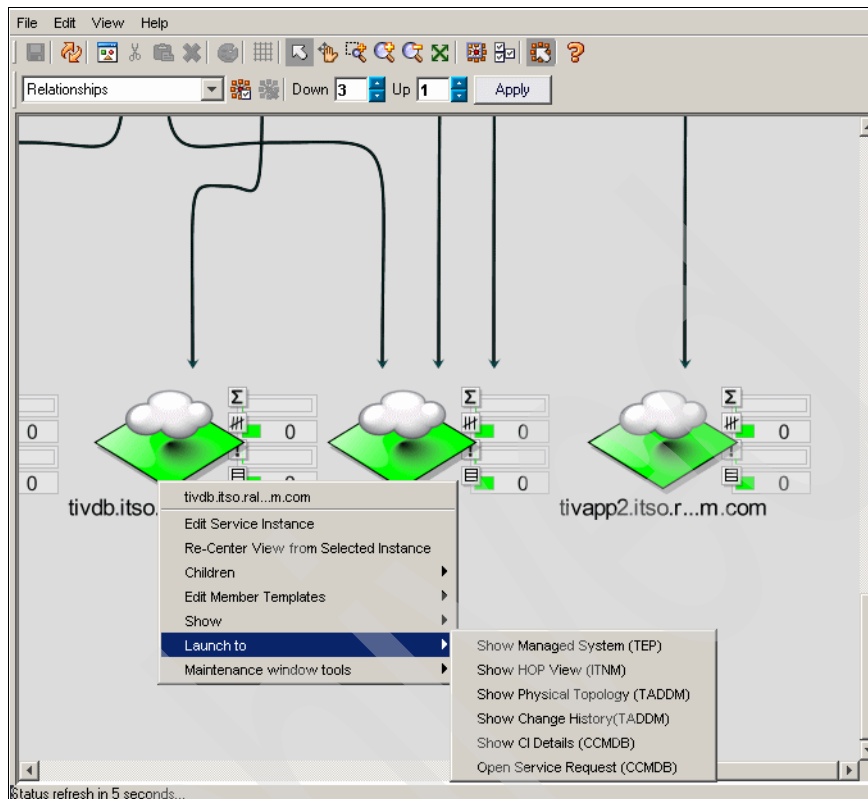


Figure 6-40 Launching application from IBM Tivoli Business Service Manager

In addition to the default Launch in Context capability integrations for IBM Tivoli Business Service Manager, you can add a URL with variable parameters to launch any Web site.

In this example, we configure a Launch in Context function to open the IBM Tivoli support page for the IBM Tivoli Business Service Manager product:

1. Log in to Tivoli Integrated Portal as a IBM Tivoli Business Service Manager administrator.
2. Go to **Administration** → **Service Administration**. Ensure that you have a minimum service tree created.
3. Add a new parameter to be used in the generic URL. Under Service Navigation, select **Templates**. Select the template to add Launch in Context. Under Service Editor, select the **Edit Template** tab. Select the **Additional** tab. Select the **New Parameter** icon. Enter the Parameter name with a variable name and the Default value with the variable value.

Figure 6-41 illustrates adding a new parameter for the generic URL to the template. In this example, we add Parameter TivoliProduct with Default value IBMTivoliBusinessServiceManager.

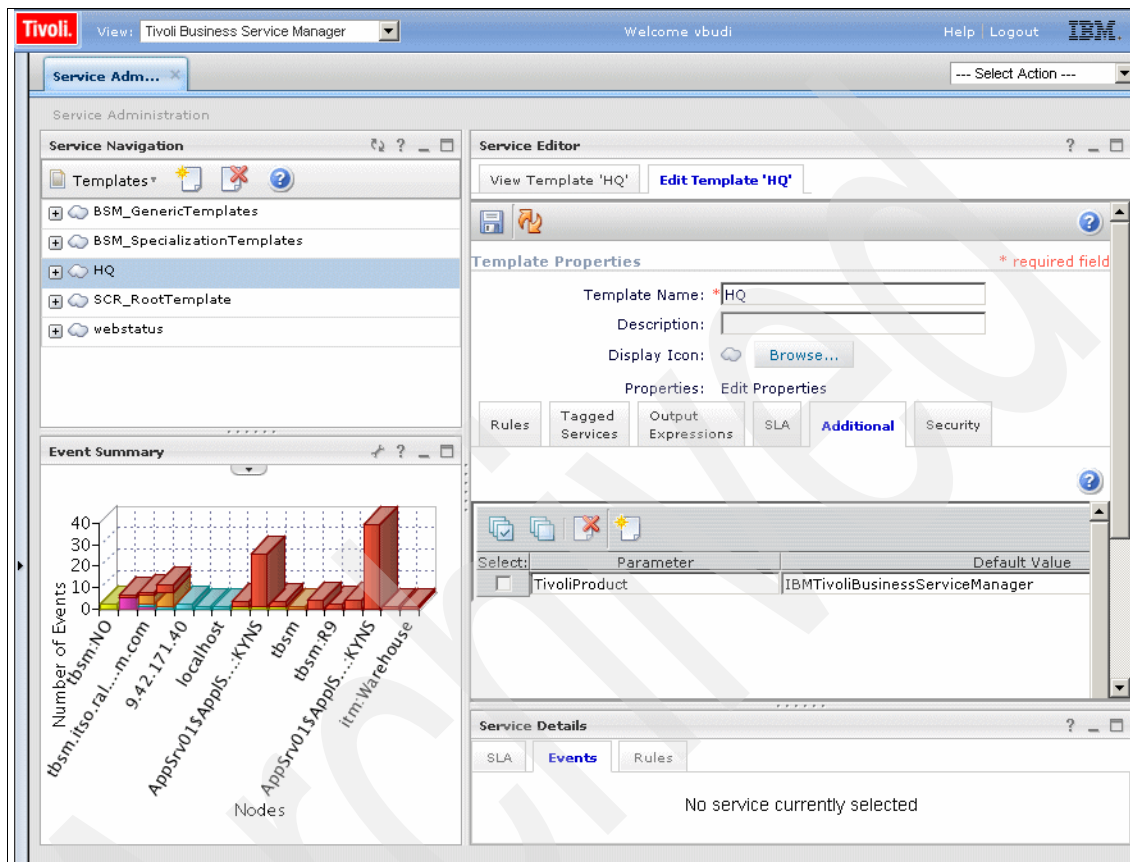


Figure 6-41 Adding a new parameter

4. Under Service Navigation, select **Services**. Select a service with the template that was edited in the previous step. Under Service Editor, on the **View Service** tab, click the **Edit View Definition** icon ().
5. On the Edit View Definition window, click **Save As New**.
6. Enter the name of the Launch in Context menu item. Click **OK**. In Figure 6-42 on page 299, we add a new Launch in Context view definition called LIC_Support_Page.

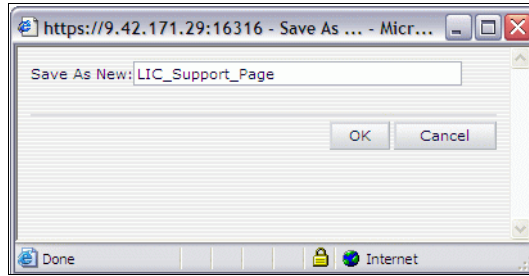


Figure 6-42 New view definition

7. Click the  icon for LIC_Support_Page. Select the **Action** tab. Click the **Add Action** icon. Complete the following parameters, and click **OK**:

Action Name	Name of action created in the previous step
Action Display Name	Menu display name for action
Action Description	Optional description
Action URL	Full URL to the Web site. Substitute any parameters with the format __parameter__
Action Frame	Use the default _blank

Figure 6-43 illustrates creating an action for the URL http://www.ibm.com/software/sysmgmt/products/support/__TivoliProduct__.html. TivoliProduct is a previously defined parameter.

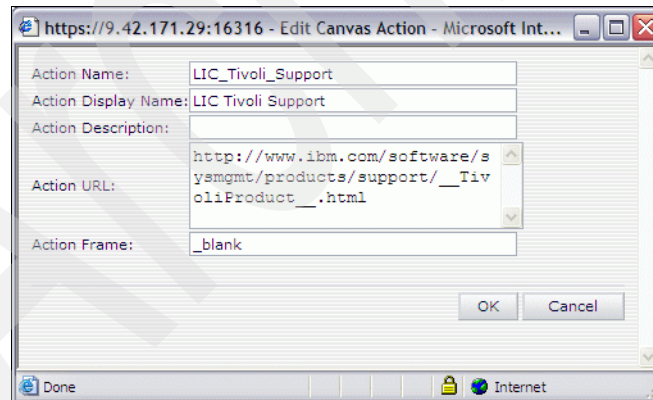


Figure 6-43 Add Action

8. Under Right Click Menu Options, ensure the new Action was added. If not, click the **New** icon under Right Click Menu Options. In the new drop-down option, select the Action Name that was created in the previous step.

Actions are defined for each view definition and each template. If the action needs to be defined for multiple views and templates, repeat the previous steps for each view and template.

Figure 6-44 illustrates editing the new action Launch in Context Tivoli Support for the selected Service Template. A new Right Click menu option is defined for the LIC Tivoli Support action.

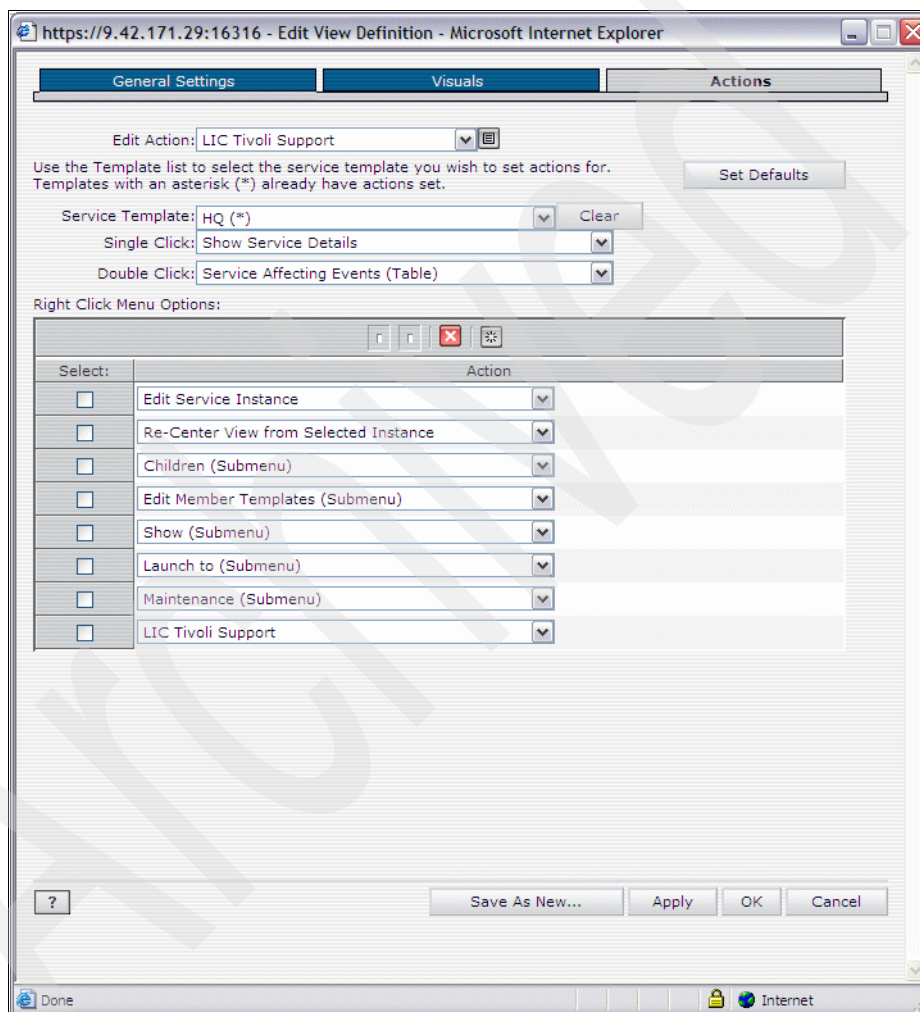


Figure 6-44 Right Click Menu Options

9. Before launching the new action, you must add the parameter to the custom view definition XML file. On the IBM Tivoli Business Service Manager Data Server, make a backup copy and edit the file:

```
$TIP_HOME/profiles/TBSMProfile/installedApps/TBSMCell/tbsm.ear/sla.war/av/xmlconfig/ViewDefinition_LIC_Support_Page.xml
```

At the end of the `<fieldToPassToModelExpr modelField=...>` entries, add the new parameter in the format `<fieldToPassToModelExpr modelField="parameter name">parameter name</fieldToPassToModelExpr>`

Example 6-15 illustrates the `TivoliProduct` parameter added at the end of the `fieldToPassToModelExpr` entries.

Example 6-15 ViewDefinition_LIC_Support_Page.xml

```
<fieldToPassToModelExpr modelField="cumulSLAStatusMax">
    replace(BadCumulThresholdSecs, ""+NULL, "100")
</fieldToPassToModelExpr>
<fieldToPassToModelExpr modelField="TivoliProduct">
    TivoliProduct
</fieldToPassToModelExpr>
</dataTypeMapping>
<templateMapping primaryTemplateName="DefaultTag">
```

10. Stop and start the IBM Tivoli Business Service Manager Data Server and Dashboard server for the view definition change to take effect. You can use the **tbsm_suite.sh stop** command to stop the processes and the **tbsm_suite.sh start** command to start them. If the dashboard server resides on a separate machine, use the **stopServer** and **startServer** commands to restart WebSphere Application Server.
11. Under the Service Editor, select the `LIC_Support_Page` view definition. Because the action was created for the `LIC_Support_Page` view definition, it will not be available in other views. Figure 6-45 on page 302 illustrates selecting the created view definition.

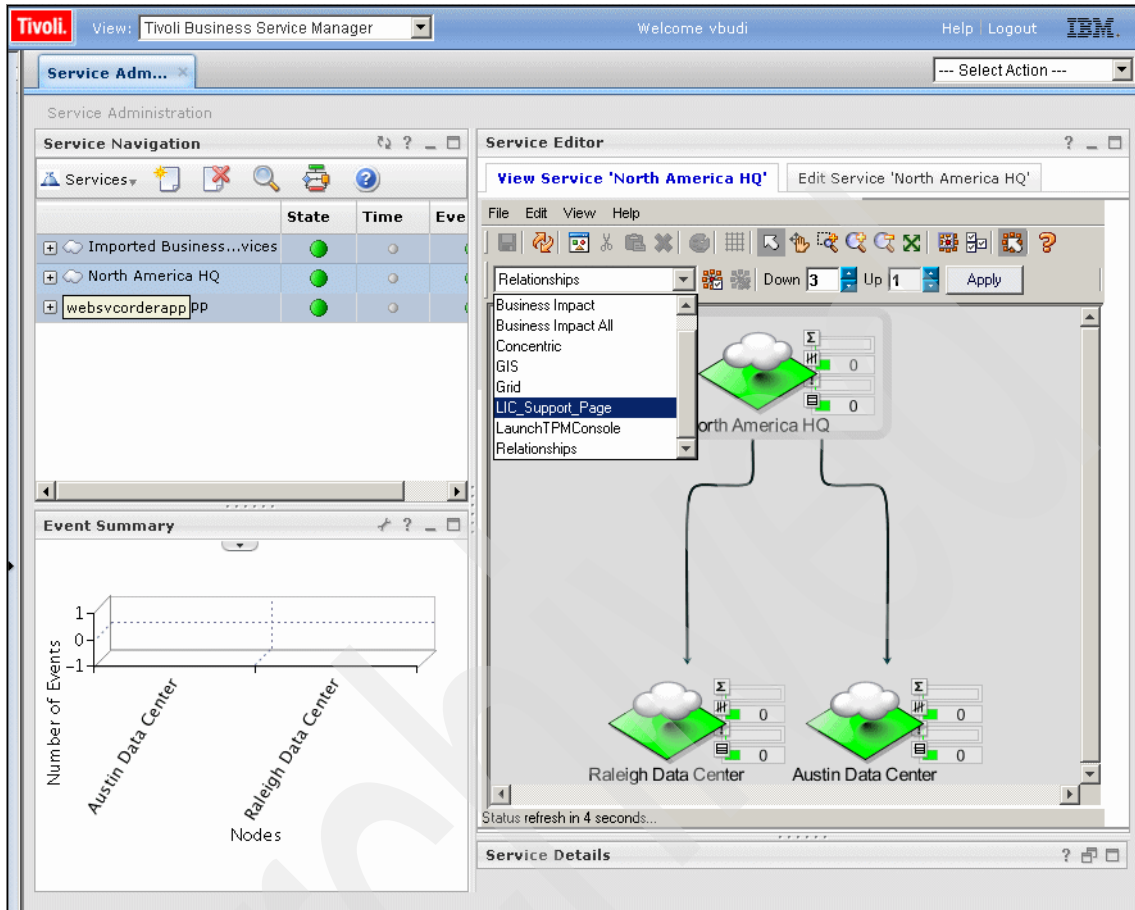


Figure 6-45 Select LIC_Support_Page view definition

12. Right-click a service, and select the new Launch in Context Support Page menu action. Figure 6-46 on page 303 illustrates the right-click menu options.

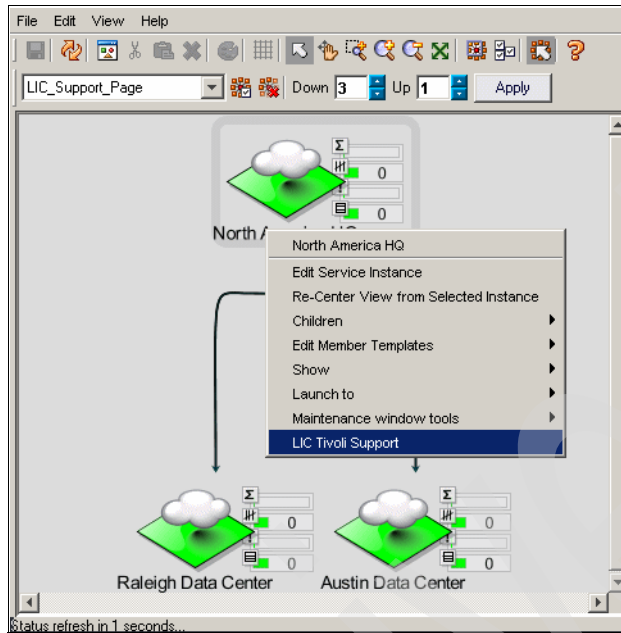


Figure 6-46 Select LIC_Support_Page menu action

13. The Launch in Context Support Page Launch in Context action is launched, opening the defined TivoliProduct parameter IBM support page. Figure 6-47 on page 304 illustrates the launched generic URL.

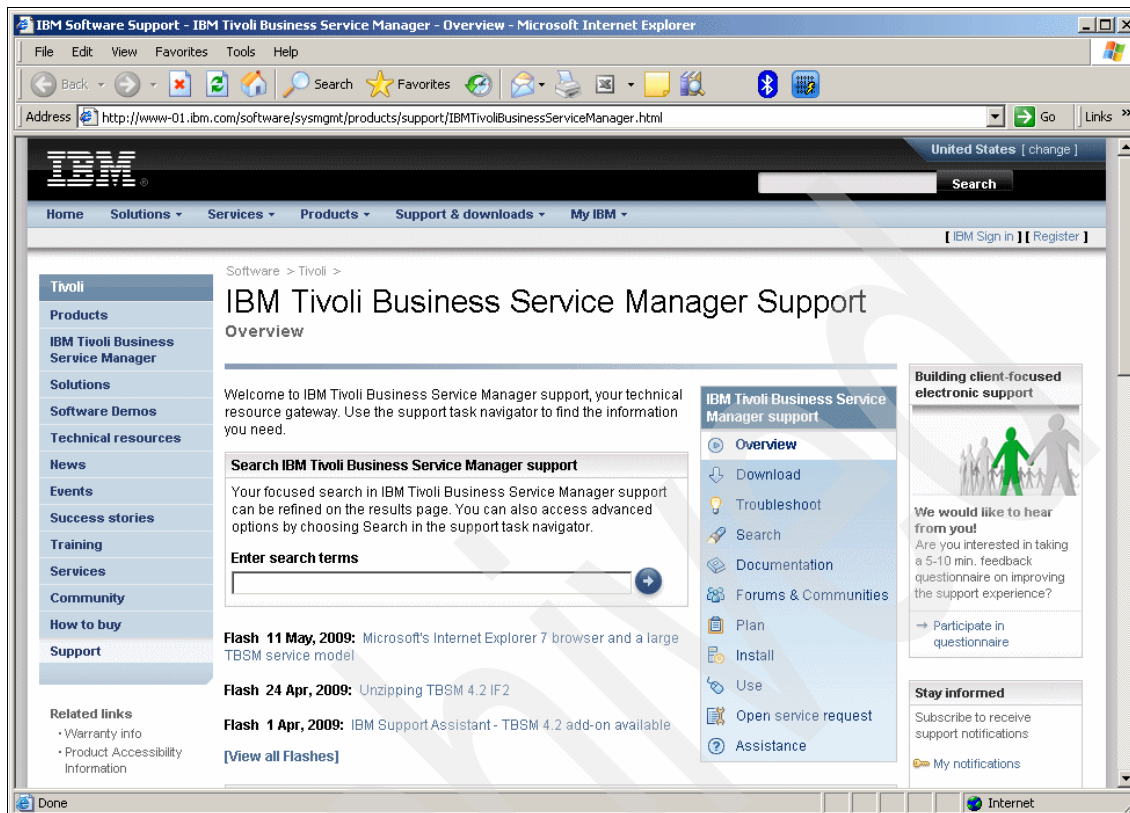


Figure 6-47 Launched generic URL

Note: You can use a generic URL to launch to other applications. There are three common service attributes that are defined by default: ServiceInstanceID, ServiceInstanceName, and PrimaryTagName:

- ▶ ServiceInstanceID is the unique ID associated with the service, for example, ServiceInstanceID=77
- ▶ ServiceInstanceName is the Service Name, for example, ServiceInstanceName=tivapp2.itso.ra1.ibm.com(898C0D7F1972370DBDCEACCE2146F643)-LinuxUnitaryComputerSystem
- ▶ PrimaryTagName is the primary template associated with the service, for example, PrimaryTagName=BSM_Node

Each service instance has additional parameters that can be used for more detailed launching parameters. Figure 6-48 shows these additional parameters. If any of the attributes do not exist, the option is inactive.

Additional Parameters	
EventMatchingFieldsSet	"Created by parser"=(%__%__0
ForceValidationWhenInvalidated	WHEN_PERSISTED
IBM_Tivoli_Monitoring_Services_sourceContactInfo	http://itm.itso.ral.ibm.com:1920
IBM_Tivoli_Monitoring_Services_sourceToken	ip_address=9.42.171.36
IBM_Tivoli_Network_Manager_IP_Edition_sourceContactInfo	https://tnmip.itso.ral.ibm.com:1631
IBM_Tivoli_Network_Manager_IP_Edition_sourceToken	5&domain=ITSO
ImportAllDescendants	true
InvalidationPeriodSecs	86400
LifeTimeSecs	96400
Order	
PersistifyAutopopInstances	false
PersistifyInstanceIfParentIsPersistent	true
SyncTagsWithPolicy	true
TADDMM_sourceContactInfo	http://taddm.itso.ral.ibm.com:9430
TADDMM_sourceToken	2AE592EFA9D13B7892F2A736062E1
_PrimaryTag	BSM_Node
_TagList	BSM_Node:::SCR_RetrieveDepende
cdm:Fqdn	tivapp1.itso.ral.ibm.com
cdm:Label	tivapp1.itso.ral.ibm.com
cdm:ManagedSystemName	tivapp1:LZ
cdm:Name	tivapp1.itso.ral.ibm.com
class	cdm:sys.linux.LinuxUnitaryCompute
dependency_cnt	4
id	130
label	tivapp1.itso.ral.ibm.com
othertemplates	BSM_Node:::SCR_RetrieveDepende
primarytemplate	BSM_Node
radinstanceid	tivapp1.itso.ral.ibm.com(2AE592EF#

Figure 6-48 Additional Parameters

You can use the appropriate sourceToken and sourceContactInfo to launch the appropriate application. SourceContactInfo contains the target URL. The sourceToken contains the unique object name that the service object represents. Figure 6-48 on page 305 contains the sourceContactInfo and sourceToken for these applications:

- ▶ IBM Tivoli Application Dependency Discovery Manager: TADDM_*
- ▶ IBM Tivoli Monitoring: IBM_Tivoli_Monitoring_Services_*
- ▶ IBM Tivoli Network Manager for IP:
IBM_Tivoli_Network_Manager_IP_Edition*

Use these fields in the following manner:

1. On the IBM Tivoli Business Service Manager Dashboard Server, edit the \$TIP_HOME/systemApps/isclite.ear/sla.war/etc/rad/RAD_sla.props file.
2. Uncomment the line impact.sla.ccmbd.sourceContactInfo, and edit the IBM Tivoli Change and Configuration Management Database host name and console port. For example:
impact.sla.ccmbd.sourceContactInfo=http://ccmbd.itso.ra1.ibm.com:9080
3. On the IBM Tivoli Business Service Manager console, you can also right-click a service object and select **Launch to** → **Open Service Request (CCMDB)**. Figure 6-49 on page 307 illustrates the open service request launch to IBM Tivoli Change and Configuration Management Database.

Create Service Request

Use this form to fill out a new request for service. When you are done, select the Submit button below to create the record. For additional details about a field description, place the cursor in the field and press Alt located next to a field may be used to assist in choosing/selecting an appropriate value for a field.

Reported By: VBUDI
 Phone: 123456789
 E-mail: vbudi@us.ibm.com
 Affected User: VBUDI

Affected Asset:
 Affected Asset Location:
 Affected Configuration Item:
 Reported Priority:
 Reported Date: 2009-05-08 14:06:09

[Download Remote Diagnostics Agent](#)

Request Description
 Please enter a summary and a more detailed description of your Service Request.
 Summary:
 Details:

Classify
 Use this section to choose the best classification or grouping for this request. You can do it filling or the Class Description fields.
 Classification:
 Class Description:

Attachments
 Use this section to add any additional documents that would help in processing your Service Request. Attach multiple files and URL links using the buttons below.
 Document:
 Description:
 ...No rows to display...
 Attach File Attach Web Ad

Attributes
 Filter:

Attribute	Description	Data Type	Alphanumeric Value	Numeric Value	Unit of Measure	Table Value
...No rows to display...						

 Submit

Figure 6-49 Open Service Request

Launching IBM Tivoli Netcool/Webtop

IBM Tivoli Business Service Manager includes default IBM Tivoli Netcool/Webtop views, which are available in Tivoli Integrated Portal. To access these views:

1. Log in to the Tivoli Integrated Portal console. Go to **Administration** → **Service Administration**. Ensure that you have a minimum service tree created.
2. Right-click the object for which you want to see data.
3. The event table is automatically refreshed, or select **Show** → **Service Affecting Events (Table)**. Figure 6-50 on page 308 illustrates the event table in the lower-right corner.

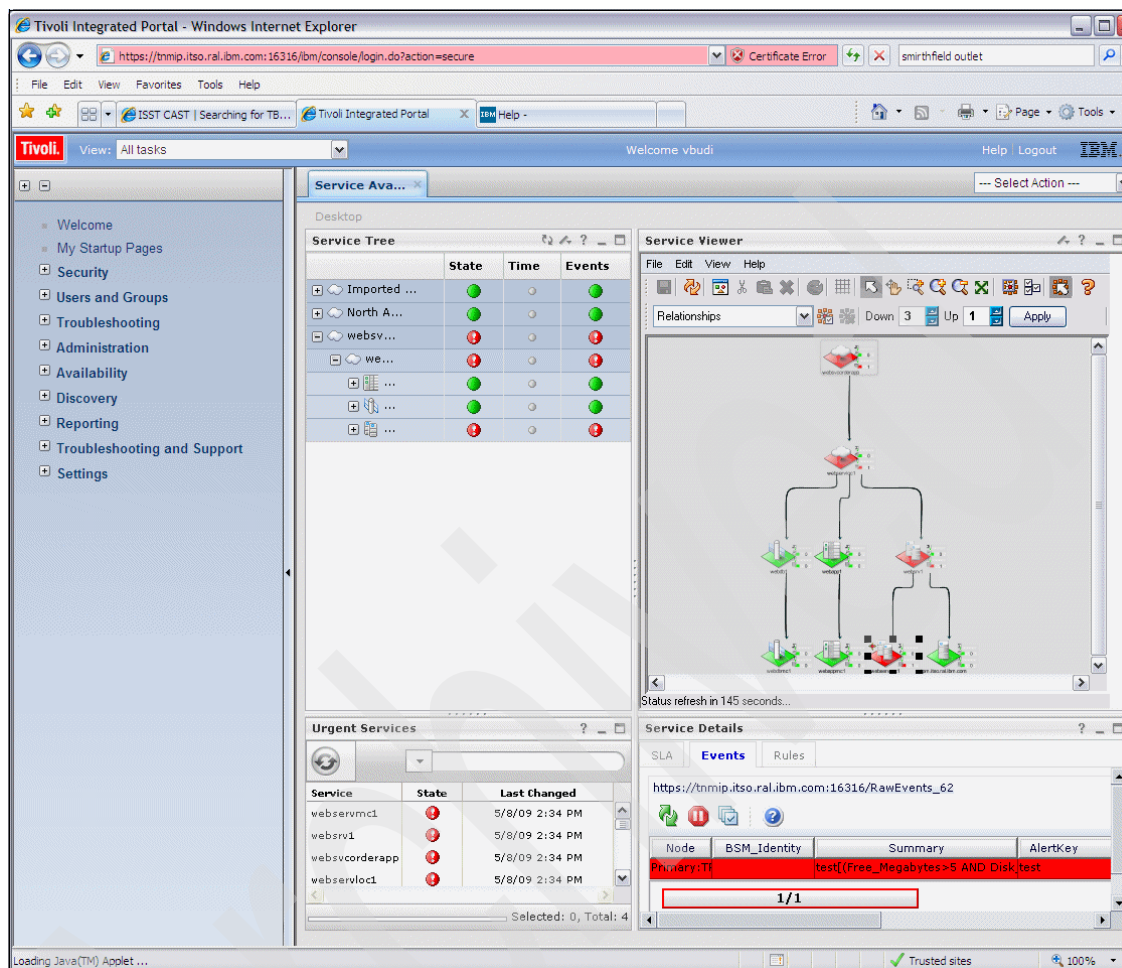


Figure 6-50 Event table

- On the IBM Tivoli Business Service Manager console, you can also right-click the object and select **Show** → **Service Affecting Events (AEL)**. If prompted, select **Run** to run the application. The Active Event List is opened on a new Tivoli Integrated Portal tab.

Figure 6-51 on page 309 illustrates the IBM Tivoli Netcool/Webtop Active Event List.

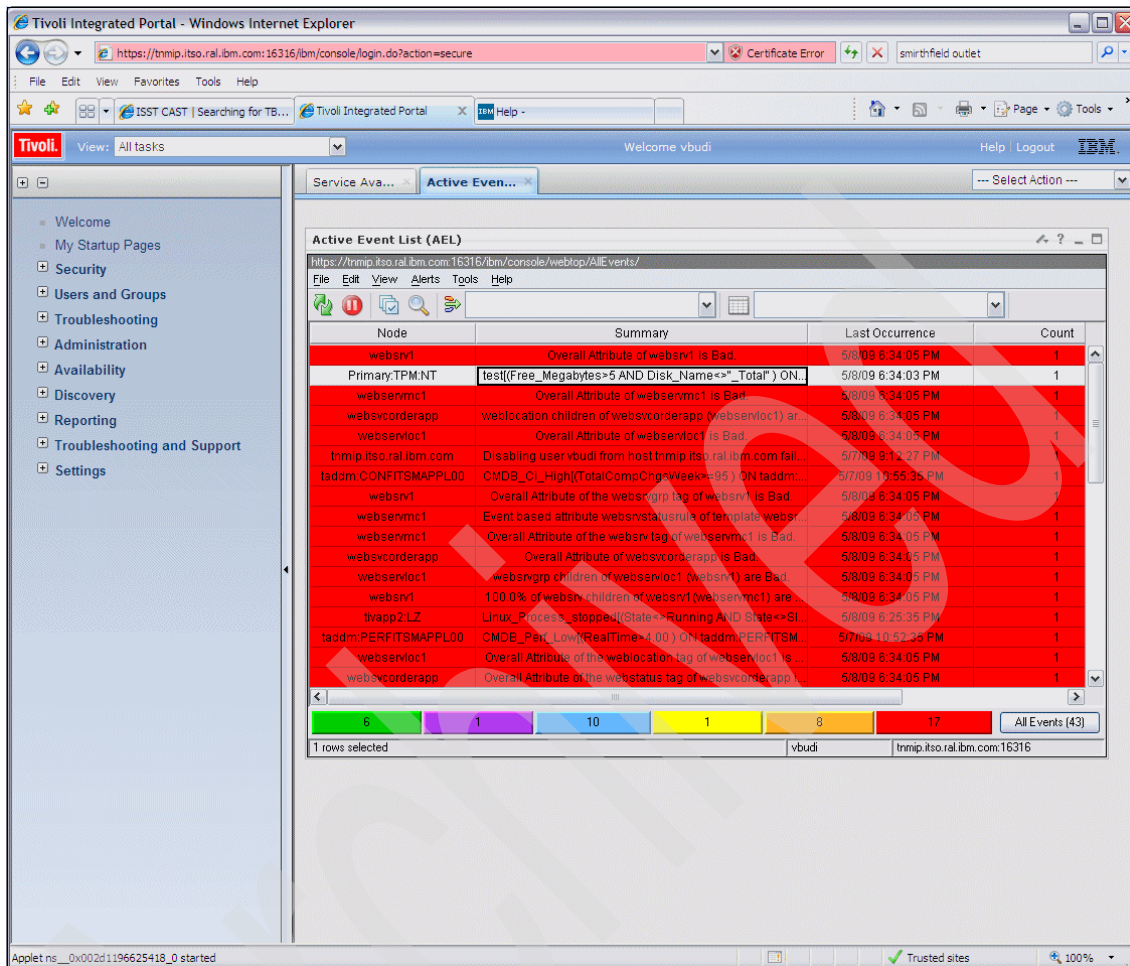


Figure 6-51 Active Event List

6.3.6 IBM Tivoli Netcool/Webtop Active Event List

The list of events from IBM Tivoli Netcool/OMNIBus is displayed under IBM Tivoli Netcool/Webtop's Active Event List. The interface that allows launch in context is provided by IBM Tivoli Netcool/Webtop. You can manipulate the Active Event List to provide "launch out" capabilities.

The Alerts menu contains a number of SQL tools that you can use to interact with alert data and manipulate the data. By default, the Tools menu contains CGI tools and local (command-line) tools. You can access the Alerts menu either from the Active Event List toolbar or by right-clicking an event in the Active Event List. You can access the Tools menu from the Active Event List toolbar.

You can configure tools in the Tools menu with access criteria that apply to users and events. Tools are visible only if the access criteria applied to them are met, or when no criteria are set because no groups or classes have been defined. If multiple events are selected in the Active Event List, all access criteria must be satisfied for all selected events for a tool to be displayed.

You can categorize the creation of launch out tools this way:

1. Use the Tools Editor to create the tool that you want. See “Tools editor” on page 310.
2. Possibly use a CGI script to extend the functionality of the tool. See “CGI program definition” on page 313.
3. Add the tools to the menus by using the Menus Editor. See “Menu item definition” on page 315.

Tools editor

Use the tool editor:

1. Open the **Administration** → **Event Management Tools** → **Entities** → **Tools creator** from Tivoli Integrated Portal. See Figure 6-52 on page 311.

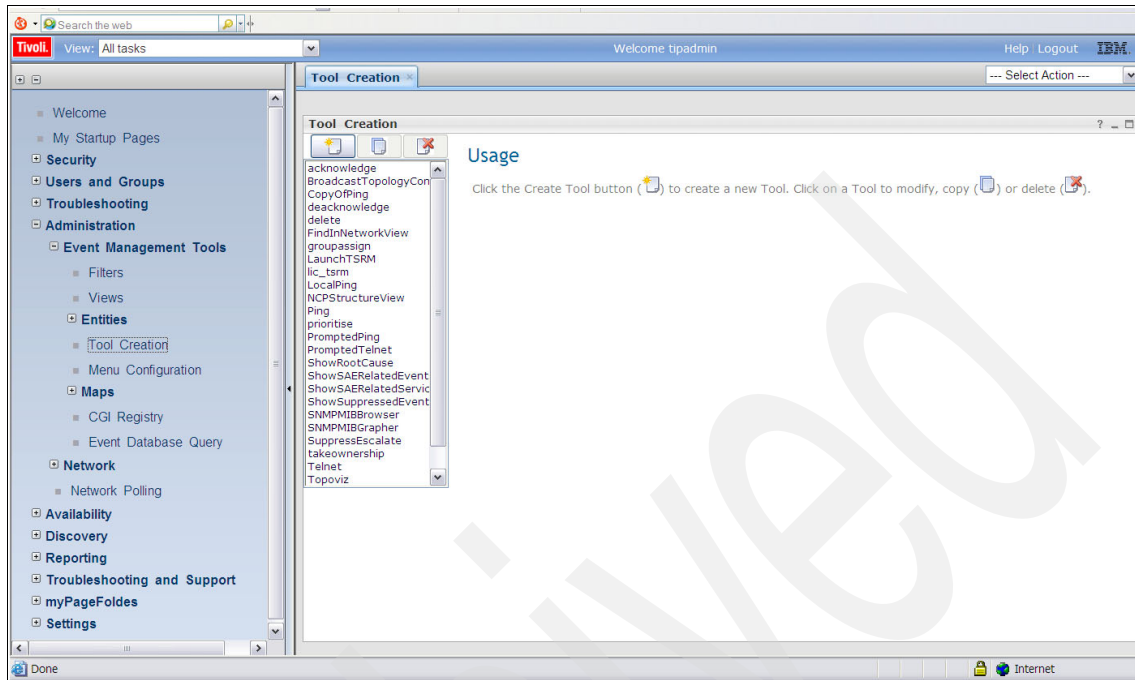


Figure 6-52 Tools browser

2. Click **Create Tool**, specify the tool name, and select the type of tool that you want to create. In Figure 6-53 on page 312, we create a new CGI/URL tool.

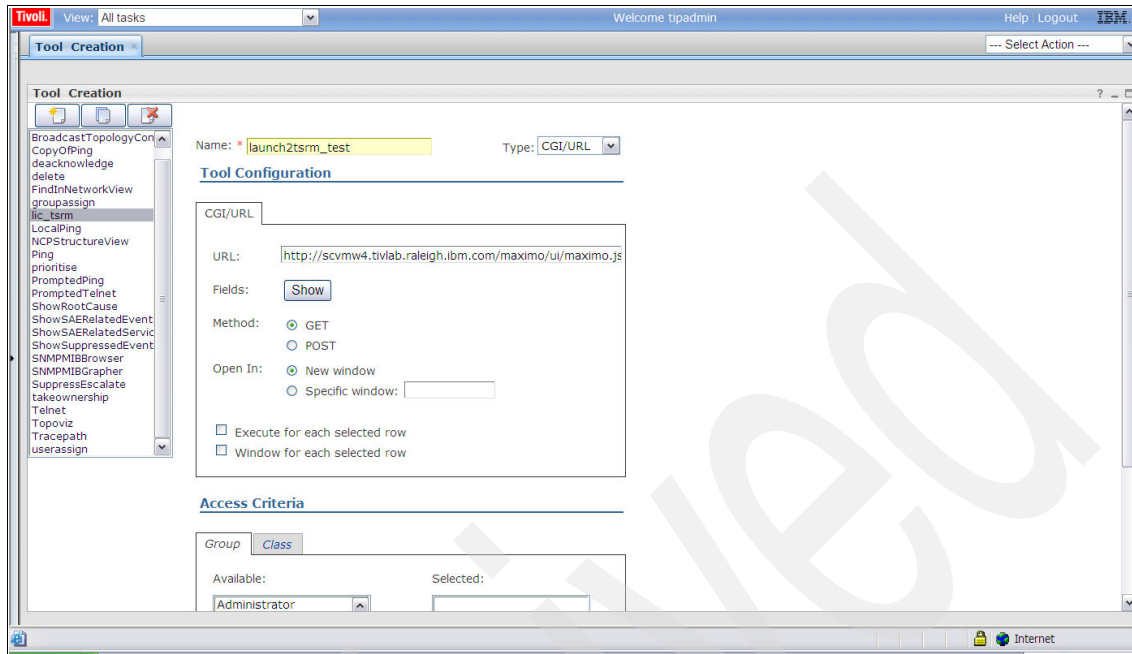


Figure 6-53 Webtop admin CGI tool menu

3. You can also include the alert fields to pass to the CGI tools. Click the Fields: **Show** button, and select the fields. See Figure 6-54.

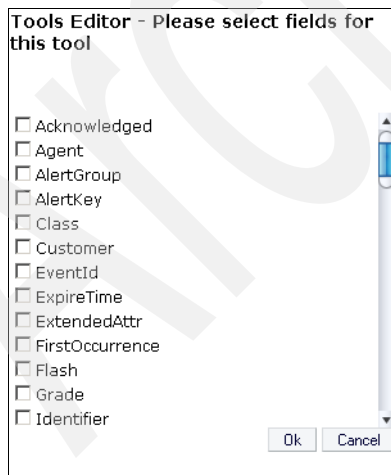


Figure 6-54 Field definitions

4. Set the group authorization and assign it to the appropriate group.

5. The tools require that you define a text file according to the tool's name. Create the `launch2tsrm_test.nova` file in the `<TIP_Install_dir>\profiles\TIPProfile\etc\webtop\configstore\ncwTools\` directory. Example 6-16 shows sample content.

Example 6-16 Launching IBM Tivoli Service Request Manager

```
tool(name="launch2tsrm_test")
{
    access
    {
        osfield
        security
    }

    cgiurl(windowforeach="false",method="GET",target="_blank",foreach="false")
    {
        url
        {

            text(data="http://scvmw4.tivlab.raleigh.ibm.com/maximo/ui/maximo.jsp")
            {
                query
                {

                    text(data="event=loadapp&value=incident&additional=event=sqlwhere&additional=eventvalue=ticketid%3D%271050%27")
                    {
                    }
                }
            }
        }
    }
}
```

CGI program definition

Create and register a new CGI program by performing the following steps:

1. Create a CGI script in the `$NCHOME/etc/webtop/cgi-bin` directory. We create an example CGI script called `example-cgi.cgi` for finding all listening server ports. Example 6-17 shows the content.

Example 6-17 Example CGI script

```
#!/usr/local/bin/perl
print "Content-type: text/html\n\n";
print <<__HTML__>>;
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML><HEAD><TITLE>My sample WebTop CGI</TITLE></HEAD>
```



```

<BODY>
<h1>WebTop CGI tools</h1>
<CENTER><HR HEIGHT="15" ></CENTER>
<PRE>
__HTML__
system("/bin/netstat -an | grep LISTEN | grep -v STREAM");
print <<__HTML__;
</PRE>
<CENTER><HR HEIGHT="15" ></CENTER>
<form action="">
<div align="center"><input type="button" value="Close Window"
onClick="javascript:window.close();"></div>
</form></BODY></HTML>
__HTML__

```

2. Open Tivoli Integrated Portal, and use the CGI registry option. See Figure 6-55.

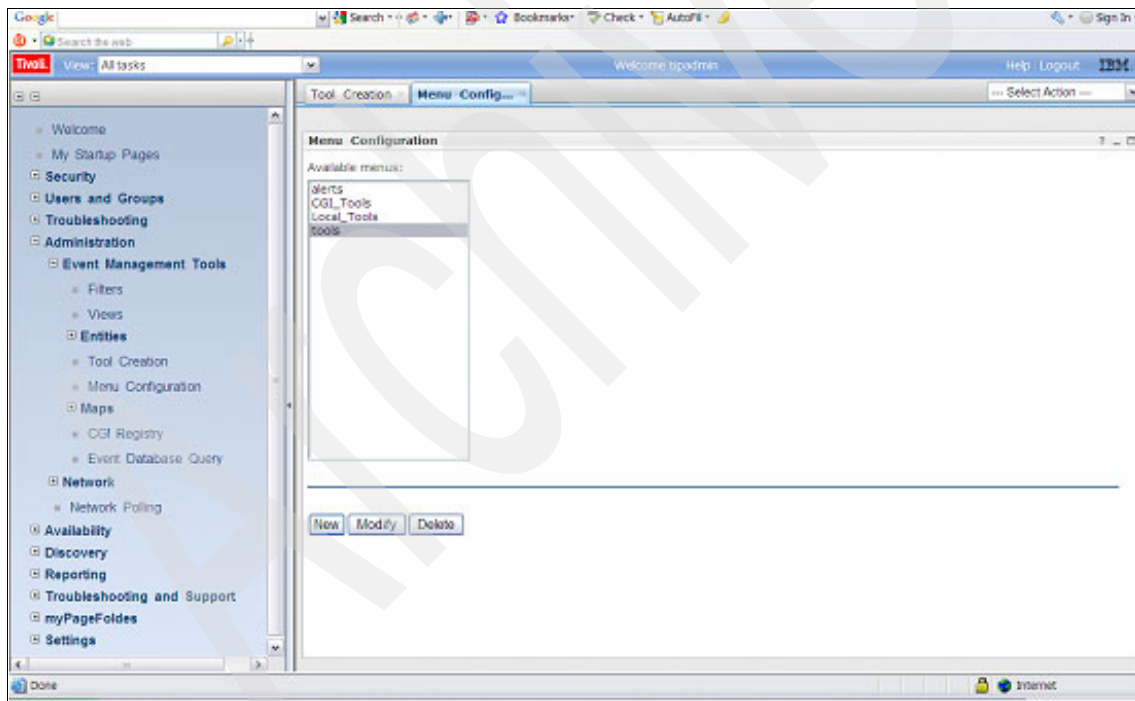


Figure 6-55 CGI registration

3. In the Register CGI page, click **Register**. Enter the CGI name and File name, and select the groups that have access to the CGI.

4. Click **Register** to save the cgi in the registry.
5. You can invoke the cgi by using the URL:
`http://<webtop>:8080/webtop/cgi-bin/<cgi-name>`

So, in our case, the URL is:

`http://webtop:8080/webtop/cgi-bin/example_cgi.`

Figure 6-56 shows the resulting page.

WebTop CGI tools				
tcp	0	0 127.0.0.1:4100	0.0.0.0:*	LISTEN
tcp	0	0 0.0.0.0:32841	0.0.0.0:*	LISTEN
tcp	0	0 0.0.0.0:111	0.0.0.0:*	LISTEN
tcp	0	0 127.0.0.1:631	0.0.0.0:*	LISTEN
tcp	0	0 0.0.0.0:27000	0.0.0.0:*	LISTEN
tcp	0	0 0.0.0.0:8089	0.0.0.0:*	LISTEN
tcp	0	0 127.0.0.1:25	0.0.0.0:*	LISTEN
tcp	0	0 0.0.0.0:891	0.0.0.0:*	LISTEN
tcp	0	0 ::ffff:127.0.0.1:5600	:::*	LISTEN
tcp	0	0 :::32803	:::*	LISTEN
tcp	0	0 :::8009	:::*	LISTEN
tcp	0	0 :::8077	:::*	LISTEN
tcp	0	0 :::8080	:::*	LISTEN
tcp	0	0 ::ffff:127.0.0.1:8085	:::*	LISTEN
tcp	0	0 :::22	:::*	LISTEN
tcp	0	0 :::1275	:::*	LISTEN

Figure 6-56 Result from the CGI tool

Menu item definition

Use menu items to actually invoke a tool (which can be a CGI program, SQL, or command tool). To define the menu, you must define the tool definition in advance. After Webtop is installed and running, create a new menu item, and make it available for use by performing the following steps:

1. Select the Menu Configuration pane to open the Menus Editor from the Webtop admin page, as shown in Figure 6-57 on page 316.

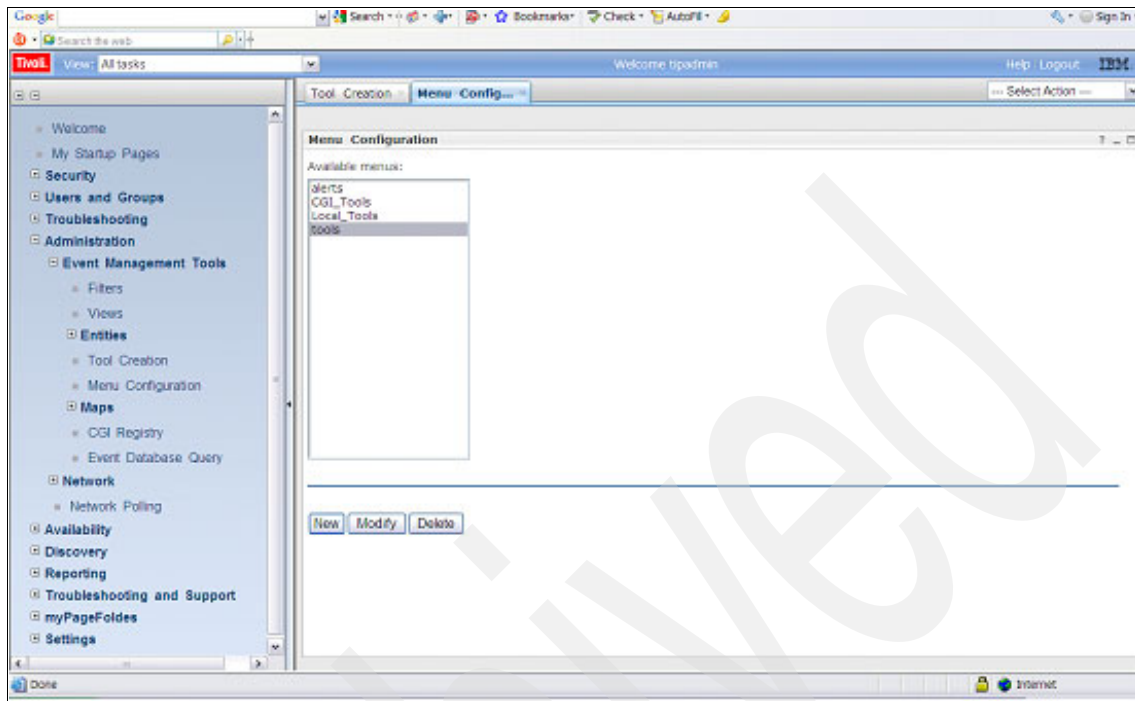


Figure 6-57 Webtop admin menu pane

2. We create a tool called Listentool similar to “Tools editor” on page 310 that defines the CGI example from “CGI program definition” on page 313.
3. Now, we modify the CGI_Tools (or we can create a new menu item), highlight the CGI_Tools menu, and click **Modify**. The dialog is shown in Figure 6-58 on page 317. You can add a tool or submenu to the selected menu by highlighting the object and clicking Add (>).

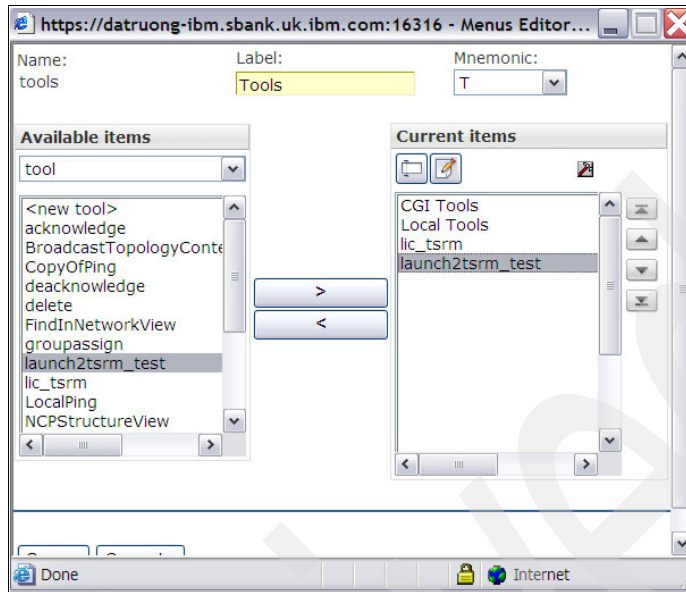


Figure 6-58 Webtop admin alerts menu Menus Editor

4. Open the Webtop desktop **Event Lists** tab and open **Active Event List** (if you have an open Active Event List, you must close it first before the new menu is available). From the menu bar, use **Tools** → **CGI Tools** → **launch2tsrm_test** to verify that the new tool has been added in the menu as shown in Figure 6-59 on page 318.

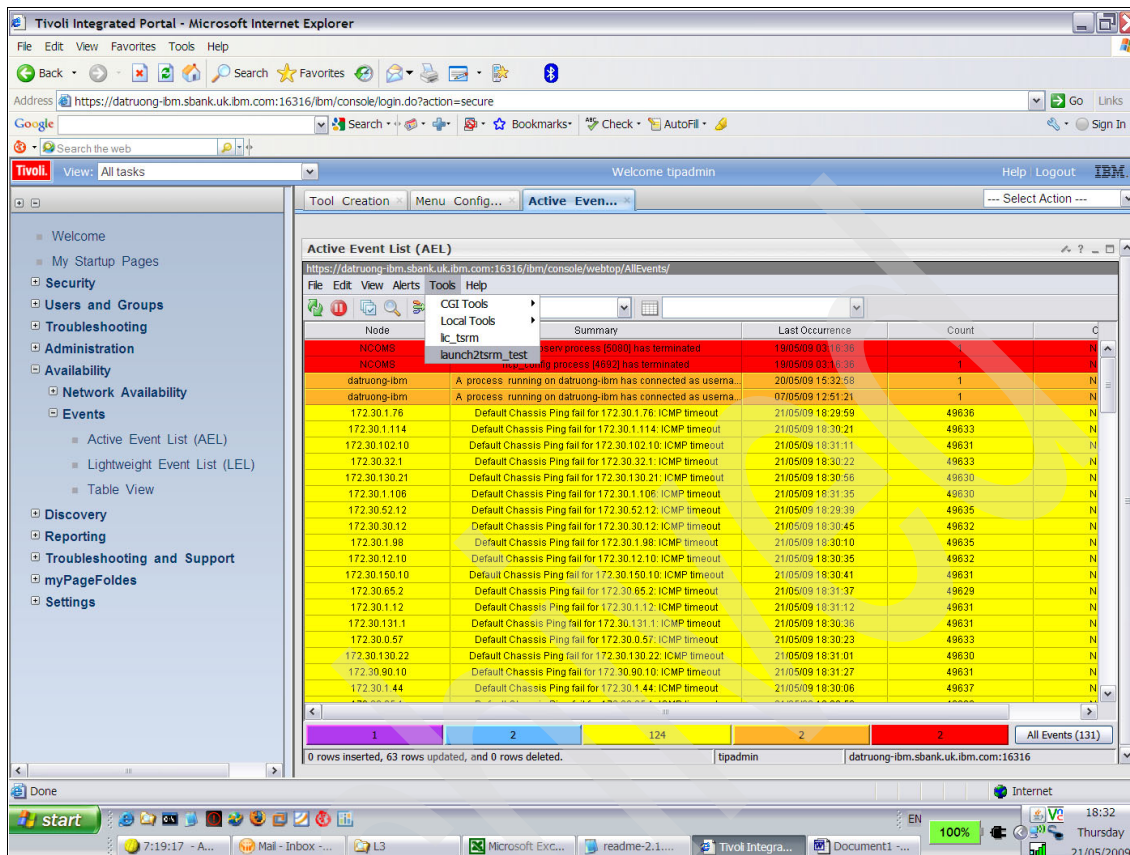


Figure 6-59 Webtop Active Events List (AEL) alerts menu with new tool

6.3.7 IBM Tivoli Storage Productivity Center for Data

The context menu in Tivoli Storage Productivity Center can be modified. To add an external tool in Tivoli Storage Productivity Center, follow these steps:

1. Expand the **IBM Tivoli Storage Productivity Center** → **Configuration Utility** → **Element Manager**.
2. In the External Tools section, select **Add Tool** from the **Select Action** list. The Add External Tool dialog box is displayed.
3. The Add External Tool window opens. Enter definitions in the following fields:

URL/Command

Enter the URL or shell command representing the external tool. You can enter an IPv4 or IPv6 address, depending on what is supported on the machine where the external tool is located. You must include

brackets [] around IPv6 addresses to separate those addresses from their port numbers, for example:
[2001:DB8::1234:0000:0000:5678:ABCD]:9550.

Label

Enter the name of the tool. The label name is shown in the Name field in the external tools table. If this field is blank, the Name field is blank.

Description

Enter a short description for the external tool.

4. Click **Save**.

The tool can then be launched from **IBM Tivoli Storage Productivity Center** → **Configuration Utility** → **Element Manager**. Select the tool from the External Tools section, and select **Launch Tool** from the **Select Action** list. The URL or command that is defined for the tool is run immediately.

Self monitoring and management

In this section, we discuss several self-monitoring agents that integrate with the IBM Tivoli Monitoring environment. IBM Tivoli Monitoring can monitor the management environment and allow agents to respond to abnormalities according to user policy. For instance, an agent that is designed to monitor the IBM Tivoli Provisioning Manager product can report a problem with IBM Tivoli Provisioning Manager server to the Monitoring Application so that the user can take any necessary action.

We discuss the following available agents:

- ▶ 7.2, “IBM Tivoli Provisioning Manager agent” on page 322
- ▶ 7.3, “IBM Tivoli Business Service Manager agent” on page 326
- ▶ 7.4, “IBM Tivoli Netcool/OMNIBus monitoring agent” on page 327
- ▶ 7.5, “IBM Tivoli Application Dependency Discovery Manager” on page 328
- ▶ 7.6, “IBM Tivoli Network Manager for IP monitoring” on page 330
- ▶ 7.7, “IBM Tivoli Workload Scheduler agent monitoring” on page 331
- ▶ 7.8, “IBM Tivoli Netcool/Impact self-monitoring agent” on page 336

7.1 Self monitoring overview

Monitoring the health of the management environment is as critical as managing the business processes and applications. Tivoli's ability to self-monitor the availability and performance of the management system enables you to differentiate the problems occurring in the business applications or processes from those problems occurring in the management system, thereby aiding speedy problem determination and recovery.

You can use the monitoring agents that are used to bring information from other Tivoli tools to IBM Tivoli Monitoring to minimize the impact of receiving data that is used for integration.

Additionally, IBM Tivoli Monitoring can monitor application agents to help ensure their availability. For instance, if the agent responsible for monitoring IBM Tivoli Provisioning Manager server's availability exits unexpectedly, the IBM Tivoli Monitoring operating system agent can automatically restart it using its embedded Agent Management Services capability.

The option to manage or unmanage an agent lets you to decide whether a Proxy Agent Service instance takes control of a particular agent. You can monitor the status of these agents through Agent management workspaces with views.

7.2 IBM Tivoli Provisioning Manager agent

IBM Tivoli Provisioning Manager 7.1 comes with Tivoli Monitoring Agent, which allows you to fully monitor its resources. The agent provides Workspaces and Situations to monitor the availability and provisioning task status.

You can navigate from IBM Tivoli Monitoring to IBM Tivoli Provisioning Manager through a Monitoring Agent to IBM Tivoli Provisioning Manager. This agent comes in the IBM Tivoli Provisioning Manager media and needs to be installed in the IBM Tivoli Provisioning Manager Server machine. The media also contains the necessary support files for IBM Tivoli Monitoring Hub Tivoli Enterprise Monitoring Server, Tivoli Enterprise Portal Server, and Tivoli Enterprise Portal client. It comes with predefined workspaces, which contain anchors pointing to IBM Tivoli Provisioning Manager URLs about tasks.

7.2.1 Requirements

Before the IBM Tivoli Provisioning Manager agent can work properly, you must install application support on Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server. In our environment, Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server are installed on Linux, so we take the IBM Tivoli Provisioning Manager agent from the IBM Tivoli Provisioning Manager core component package for Linux. For more detailed steps about installing application support, see:

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.1/itm_admin.htm

7.2.2 Installation

You can install the agent together with the IBM Tivoli Provisioning Manager core products. The integrated installation assumes that the IBM GSKit is available in the path of \IBM\GSK7. If your GSKit is installed somewhere else, you might have a problem starting the agent. You might need to define a dummy path for GSK7 to uninstall and reinstall the agent.

The stand-alone agent installation in Windows is provided at this Web site:

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.1/main_win.htm

To change the location of IBM GSKit in the Windows registry, use the Windows registry editor to navigate to \HKEY_LOCAL_MACHINE\SOFTWARE\IBM\GSKIT7\CurrentVersion and change the following keys:

BinPath	C:\ibm\itm\gsk7\bin
InstallPath	C:\ibm\itm\gsk7
LibPath	C:\ibm\itm\gsk7\lib

Perform the actual installation of IBM Tivoli Monitoring Agent for IBM Tivoli Provisioning Manager from the IBM Tivoli Provisioning Manager core component package for Windows. Extract the ITM_Agent_V71_Windows.zip file, and run Windows\Setup.exe. Follow the wizard.

Note: We recommend to *not* include IBM Tivoli Monitoring Agent in the IBM Tivoli Provisioning Manager core component installation. Install the IBM Tivoli Provisioning Manager agent after installing the IBM Tivoli Monitoring Agent for Windows Operating System agent.

Even if you install the agent manually, do not forget to change the GSKit7 Windows registry keys of Binpath, InstallPath, and LibPath in the actual GSKit7 directory before installing IBM Tivoli Monitoring Agent for IBM Tivoli Provisioning Manager.

7.2.3 Configuration

Follow these steps to configure IBM Tivoli Monitoring Agent for IBM Tivoli Provisioning Manager:

1. Open Manage Tivoli Monitoring Services by navigating to **Start → All Programs → IBM Tivoli Monitoring → Manage Tivoli Monitoring Services**.
2. Right-click **Monitoring Agent for IBM Tivoli Provisioning Manager**, and click **Configure**.
3. Make sure that **Protocol1: IP.PIPE** is chosen, and click **OK**.
4. Enter the Tivoli Enterprise Monitoring Server server in the Host name or IP Address field in the IP.PIPE Settings. In our environment, it is `itm.itso.ral.ibm.com`.
5. Click **OK** to save it.
6. On the Agent Configuration tab panel, enter the following information:
 - WebSphere Application Server Administrator User Name: `wasadmin`
 - WebSphere Application Server Administrator Password: `xxxxxxx`
 - Confirm WebSphere Application Server Administrator Password: `xxxxxxx`
 - Tasks Information Time Range: 3 Days
 - CPC Port: 2093
 - CPC Log Level: Minimum

Figure 7-1 on page 325 shows the Agent Configuration tab panel.

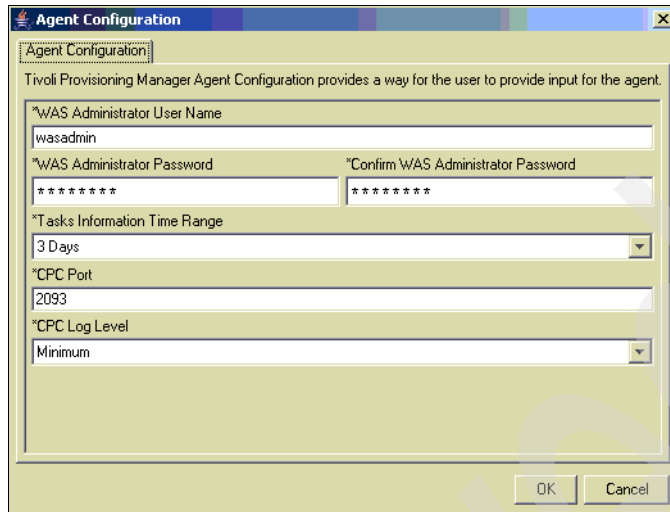


Figure 7-1 Final configuration of Agent Configuration panel

Custom Provider Client (CPC) is a process that runs in the IBM Tivoli Provisioning Manager server. Its responsibility is to collect data that is required by IBM Tivoli Monitoring Agent. For example, it collects provisioning tasks information at every certain interval, which will be used by Tivoli Enterprise Portal Server to display the information in the workspace.

7. Click **OK** to save it.
8. Right-click **Monitoring Agent for Tivoli Provisioning Manager**, and click **Start**.
9. After IBM Tivoli Monitoring Agent for IBM Tivoli Provisioning Manager has successfully connected to Tivoli Enterprise Monitoring Server, we can see the Tivoli Enterprise Portal Server workspace, as shown in Figure 7-2 on page 326.

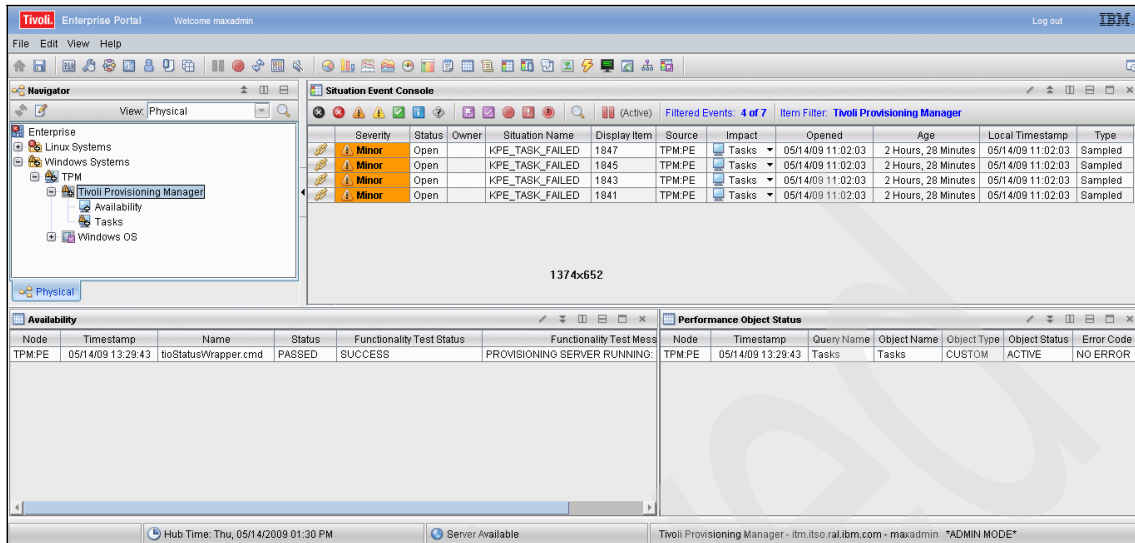


Figure 7-2 IBM Tivoli Monitoring Agent for IBM Tivoli Provisioning Manager workspace

7.3 IBM Tivoli Business Service Manager agent

You can monitor the IBM Tivoli Business Service Manager product and business services using IBM Tivoli Monitoring by implementing a IBM Tivoli Business Service Manager agent in the IBM Tivoli Business Service Manager Server machine. It brings IBM Tivoli Business Service Manager data to IBM Tivoli Monitoring. It also allows you to use the IBM Tivoli Business Service Manager Historical Reporting function, which uses the Data Warehouse feature of IBM Tivoli Monitoring to record historical IBM Tivoli Business Service Manager data.

7.3.1 Requirements

To use this agent, perform these tasks:

- ▶ Install the IBM Tivoli Business Service Manager support files in Hub Tivoli Enterprise Monitoring Server.
- ▶ Install the IBM Tivoli Business Service Manager support files for Tivoli Enterprise Portal Server and Tivoli Enterprise Portal client.
- ▶ Install the IBM Tivoli Business Service Manager agent on the IBM Tivoli Business Service Manager server.

7.3.2 Installation and configuration

The following Web site describes the installation of the IBM Tivoli Business Service Manager agent:

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tivoli.itbsm.doc/installguide/bsmi_t_installing_tbsm_agent.html

The agent code is r9. Configuring the IBM Tivoli Business Service Manager agent requires the following parameters:

- ▶ **CPC port:** The custom provider client uses this port to talk with the agent. The default is 2092.
- ▶ **RAD_eventbroker.log:** You use this file to monitor and fail over. It is in the TBSMProfile/logs/tbsm/ directory. This parameter requires the full path.
- ▶ **msgGTM_XT.log:** You use the Discovery Library Toolkit installation directory to monitor the directory.

7.4 IBM Tivoli Netcool/OMNIBus monitoring agent

You monitor the health checking of IBM Tivoli Netcool/OMNIBus tool by IBM Tivoli Monitoring by implementing a Tivoli Monitoring Agent for Omnibus. You implement this agent in the IBM Tivoli Netcool/OMNIBus Object Server machine, and it provides IBM Tivoli Netcool/OMNIBus information by connecting to IBM Tivoli Monitoring Tivoli Enterprise Monitoring Server. You can monitor the following information:

- ▶ Availability of IBM Tivoli Netcool/OMNIBus
- ▶ Event distribution and history
- ▶ Metrics about events and the Object Server
- ▶ Historical data collection

7.4.1 Requirements

Collecting IBM Tivoli Netcool/OMNIBus event status by IBM Tivoli Monitoring requires the following tasks:

- ▶ Implement event synchronization in IBM Tivoli Netcool/OMNIBus Object Server machine.
- ▶ Configure IBM Tivoli Monitoring Hub Tivoli Enterprise Monitoring Server to send events to IBM Tivoli Netcool/OMNIBus.
- ▶ Install Application support files on the Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server.

The following document describes the requirements for installation of the Monitoring agent for IBM Tivoli Netcool/OMNIBus:

http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.netcool_OMNIBus.doc_7.2.1/om_net_agent_template26.htm

7.4.2 Implementation

For details about the installation and configuration of the Tivoli Monitoring Agent for Omnibus, refer to 3.4, “IBM Tivoli Netcool installation overview” on page 34. It installs a Tivoli Omnibus Object Server Agent V07.20.00.00 Agent with IBM Tivoli Monitoring code of no.

7.5 IBM Tivoli Application Dependency Discovery Manager

A self-monitoring tool comes with the IBM Tivoli Application Dependency Discovery Manager media. It tracks the performance and availability of the IBM Tivoli Application Dependency Discovery Manager Server and its components, as well as errors and summaries of configuration item data that is stored in Configuration Management Database (CMDB). The information can then be stored in IBM Tivoli Monitoring Data Warehouse as a historical data collection. This agent is installed in the IBM Tivoli Application Dependency Discovery Manager Server machine and connects to IBM Tivoli Monitoring to provide the monitoring data. For more information, refer to this Web site:

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/AdminGuide/c_cmdb_moswos_using.html

IBM Tivoli Application Dependency Discovery Manager can detect changes in the configuration of the discovered environment. You can generate situation events in IBM Tivoli Monitoring when a change is detected. The IBM Tivoli Application Dependency Discovery Manager Operation Management Product Change Event Module from the Tivoli Open Process Automation Library explains this integration:

<http://www.ibm.com/software/tivoli/opal/details?catalog.label=1TW10CC1Q>

7.5.1 Requirements

The Self Monitoring solution for IBM Tivoli Application Dependency Discovery Manager has the following requirements:

- ▶ An existing IBM Tivoli Monitoring Hub Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server
- ▶ Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server support files for the IBM Tivoli Application Dependency Discovery Manager self-monitoring agent that is installed in the IBM Tivoli Monitoring environment
- ▶ OS agent and Universal agent installed and running in the IBM Tivoli Application Dependency Discovery Manager Server machine
- ▶ Installation of the IBM Tivoli Application Dependency Discovery Manager self-monitoring agent in the IBM Tivoli Application Dependency Discovery Manager Server machine.

Important: The self-monitoring agent does not support Windows machines.

The self-monitoring agent is a customization of the previously installed Universal Agent.

7.5.2 Implementation

Implement the agent:

1. Install the Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server support files in IBM Tivoli Monitoring environment:
 - a. Copy the media files to the IBM Tivoli Monitoring Hub Tivoli Enterprise Monitoring Server.
 - b. Open a command line in the IBM Tivoli Monitoring Hub Tivoli Enterprise Monitoring Server.
 - c. Go to the ITM-selfmon-support/SolutionInstaller directory in the IBM Tivoli Application Dependency Discovery Manager media.
 - d. Run the binary file that is associated with your operating system.
 - e. Fill in the configuration fields, and click **Install**.

2. Install the Self-Monitoring Agent from the IBM Tivoli Application Dependency Discovery Manager Server machine:
 - a. Go to the IBM Tivoli Monitoring installation path, which usually is /opt/IBM/ITM. Make sure that the Operating System agent and Universal agent are running by running the **cinfo -r** command.
 - b. Go to the \$COLLATION_HOME/itmconfig directory and run the binary file that is associated with your operating system:

```
cd /opt/IBM/cmdb/dist/itmconfig
./cfgSelfMonitoringLinux.bin
```
 - c. Type the required configuration information, and click **Install**.

For more details about the installation, refer to this Web site:

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/InstallGuide/c_cmdb_moswos_installoverview.html

7.6 IBM Tivoli Network Manager for IP monitoring

IBM Tivoli Monitoring can monitor the health of IBM Tivoli Network Manager for IP by using a Monitoring Agent for IBM Tivoli Network Manager for IP.

The agent collects data from Network Manager for IBM Tivoli Monitoring. IBM Tivoli Monitoring uses this data to monitor the availability and performance of network processes and to summarize the state of the network.

The agent can also store Simple Network Management Protocol (SNMP) and Internet Control Message Protocol (ICMP) polled data in the Tivoli Data Warehouse instead of storing data on the local Network Manager database. Storing polled data in the Tivoli Data Warehouse allows you to use the capabilities of the warehouse and to use the stored data for further analysis.

7.6.1 Requirements

This agent requires that you perform the following tasks:

- ▶ Install the IBM Tivoli Monitoring for IBM Tivoli Network Manager for IP support files for Hub Tivoli Enterprise Monitoring Server and Tivoli Enterprise Portal Server.
- ▶ Install the IBM Tivoli Monitoring for IBM Tivoli Network Manager for IP agent in the IBM Tivoli Network Manager for IP Server.

7.6.2 Implementation

Install the agent using the launchpad or the command-line option:

- ▶ To run the installation script using the launchpad:
 - a. Start the launchpad using the **1launchpad.sh** script on UNIX or the **1launchpad.exe** executable on Windows.
 - b. Click **Post-Installation** → **Install ITNM Agent for ITM** → **Start ITM Agent Installation**.
- ▶ To run the installation script using the command line, run the **ITMagent\WINDOWS\setup.exe** script on Windows or the **ITMagent/install.sh** script on UNIX from the scripts directory of the installation media.

It installs the IBM Tivoli Network Manager for IP V03.80.00.00 agent with an IBM Tivoli Monitoring code of np.

7.7 IBM Tivoli Workload Scheduler agent monitoring

You can monitor Tivoli Workload Scheduler with IBM Tivoli Monitoring through Universal Agents. The *Universal Agent* is a powerful and customizable IBM Tivoli Monitoring agent, which contains APIs that allow it to monitor information from most types of resources and to send the monitoring information to IBM Tivoli Monitoring Server. You can include the data that is collected by the Universal Agent in the Tivoli Data Warehouse through the history configuration, which means that any kind of data that is collected by this agent can be shown in historical reports.

The Customized Universal Agent for Tivoli Workload Scheduler includes several situations for many components in Tivoli Workload Scheduler, such as messages, jobman, mailman, netman, batchman, Tivoli Workload Scheduler Host, stdlist, and many others. It generates Situation Events that go to the Hub Tivoli Enterprise Monitoring Server and then are forwarded to other event management tools, such as IBM Tivoli Netcool/OMNibus. Tivoli Enterprise Portal Server in the Situation Event Console also displays these events.

7.7.1 Requirements

The required level of software includes these components:

- ▶ IBM Tivoli Monitoring 6.1 Fix Pack 5 or later
- ▶ Tivoli Workload Scheduler 84 Fix Pack 1 or later
- ▶ The previous installation of Universal Agent on each Tivoli Workload Scheduler Master Domain Manager that you plan to monitor

7.7.2 Configuration

The installation of Tivoli Workload Scheduler provides two scripts that are necessary to implement the Tivoli Universal Agent to monitor Tivoli Workload Scheduler. We execute the following steps to configure the Universal Agent and Tivoli Enterprise Portal Server:

1. Execute the **ITMconfig.sh** script. It configures the Universal agent to monitor Tivoli Workload Scheduler. The script follows this syntax:

```
ITMconfig -ua_home_dir <ua_home_dir>
```

The argument `ua_home_dir` is the installation path of the Universal Agent.

Example 7-1 shows the execution of the script from the `/opt/IBM/TWA/TWS/ITM` directory.

Example 7-1 Running the ITMconfig.sh script

```
[root@tw5 ITM]# ./ITMconfig.sh -ua_home_dir /opt/IBM/ITM
KUMPS001I Console input accepted.
KUMPS041I Command in progress ...
KUMPS041I Command in progress ...
KUMPS020I Import successfully completed for
/opt/IBM/TWA/TWS/ITM/ITM_TWS_84metafile_ux.mdl
```

```
KUICRA007I: Restarting UM agent(s).
```

```
KUICRA015I: All UM agents on the local system were restarted.
The IBM Tivoli Monitoring 6.1.5 integration has been successfully
enabled.
```

2. To configure application support, copy the following files to the Hub Tivoli Enterprise Monitoring Server machine.

For Windows-based Tivoli Enterprise Monitoring Server, copy these files:

- `ITMCreateSituations.cmd`
- `TM_TWS_85metafile_win.mdl`

For Unix or Linux hub Tivoli Enterprise Monitoring Server, copy these files:

- ITMCreateSituations.sh
- baseNameSituation.xml
- ITM_TWS_85metafile_ux.mdl

In our example, these scripts are in the /opt/IBM/TWA/TWS/ITM directory in the Tivoli Workload Scheduler Master machine.

The **ITMCreateSituations** command configures Tivoli Enterprise Portal Server to create and display IBM Tivoli Monitoring situations for Tivoli Workload Scheduler. The Situation is a monitor that is defined in Hub Tivoli Enterprise Monitoring Server and executed in the agents.

3. Run the following command in the Hub Tivoli Enterprise Monitoring Server:

```
ITMCreateSituations -user username -password passwd -host host  
-ITMHome ITMHome [-TWSHome TWSHome]
```

where:

username	Specifies the user ID of IBM Tivoli Monitoring user.
passwd	Specifies the password of IBM Tivoli Monitoring user.
host	Specifies the host name of Tivoli Enterprise Monitoring Server.
ITMHome	Specifies the home directory of Tivoli Enterprise Monitoring Server.
TWSHome	Specifies the home directory of the Tivoli Workload Scheduler. Use this option only if Tivoli Enterprise Portal and Tivoli Workload Scheduler are installed on the same machine.

Note: We must adjust the **tacmd createSit** commands in the **ITMCreateSituations.sh** script by putting the **-b** argument before the **-p** argument. Correct the line to look like this example:

```
$ITM_HOME/bin/tacmd createSit -s TWS_SU_schedlog_Info -b  
TWSBaseName -p formula="*IF *VALUE  
ITM_TWS_APPLTWS_SU_SCHEDLOG00.KBytes *LE 102400"  
Distribution="*CUSTOM_ITM_TWS_APPL00" Interval=0/001500
```

We use this command:

```
./ITMCreateSituations.sh -user sysadmin -password itso4you -host  
itm.itso.ral.ibm.com -ITMHome /opt/IBM/ITM
```

Each situation generates the KUICCS005I message that explains the situations that are created.

4. Run the following command in the Tivoli Workload Scheduler machine where the Universal Agent is installed:

```
TEPConfig -ua_home_dir <ua_home_dir>
[eventfile_path <eventfile_path>]
[-metafile_path <metafile_path>]
[-APPL_NAME <APPL_NAME>]
[-TTL <TTL>]
[-SAMPLE_FACTOR <SAMPLE_FACTOR>] [-UA]
```

<ua_home_dir>	The home directory path of the Universal Agent installation.
<eventfile_path>	Fully qualified path of the event log file. The default value is <TWS_HOME_DIR>/event.log.
<metafile_path>	Fully qualified path to the sample metafile. The default value is: <TWS_HOME_DIR>/TEP/TWS84metafile_Sample.mdl
<APPL_NAME>	Application name that will be stored in the APPL statement of the metafile. The default value is dTWS.
<TTL>	The amount of time in seconds ("Time To Live") that the monitored data will be kept by the Tivoli Enterprise Portal. This value is set in the TTL keyword of the NAME statement in the metafile. The default value is 86400 seconds.
<SAMPLE_FACTOR>	The value for the KUMP_DP_SAMPLE_FACTOR parameter of the Universal Agent initialization file. The default value is 5760. This value is used to calculate the sampling frequency that is used by the Universal Agent to access the event file. This frequency is obtained by dividing the TTL value by the sampling factor. The default frequency is every 15 seconds.
-UA	Use this parameter if you want to modify the Universal Agent-related parameters only (APPL_NAME, TTL, or SAMPLE_FACTOR) without changing any of the existing settings in the BmEvents.conf configuration file.

Example 7-2 on page 335 shows our configuration output. We run the script from the /opt/IBM/TWA/TWS/TEP path.

Example 7-2 Running the TEPconfig.sh script

```
[root@twS TEP]# ./TEPconfig.sh -ua_home_dir /opt/IBM/ITM
KUMPS001I Console input accepted.
KUMPS020I Import successfully completed for
/opt/IBM/TWA/TWS/TEP/TWS84metafile.mdl
```

```
KUICRA007I: Restarting UM agent(s).
```

```
KUICRA015I: All UM agents on the local system were restarted.
The TEP integration has been succesfully enabled.
```

5. Recycle the Tivoli Workload Scheduler master.

Figure 7-3 shows a sample workspace.

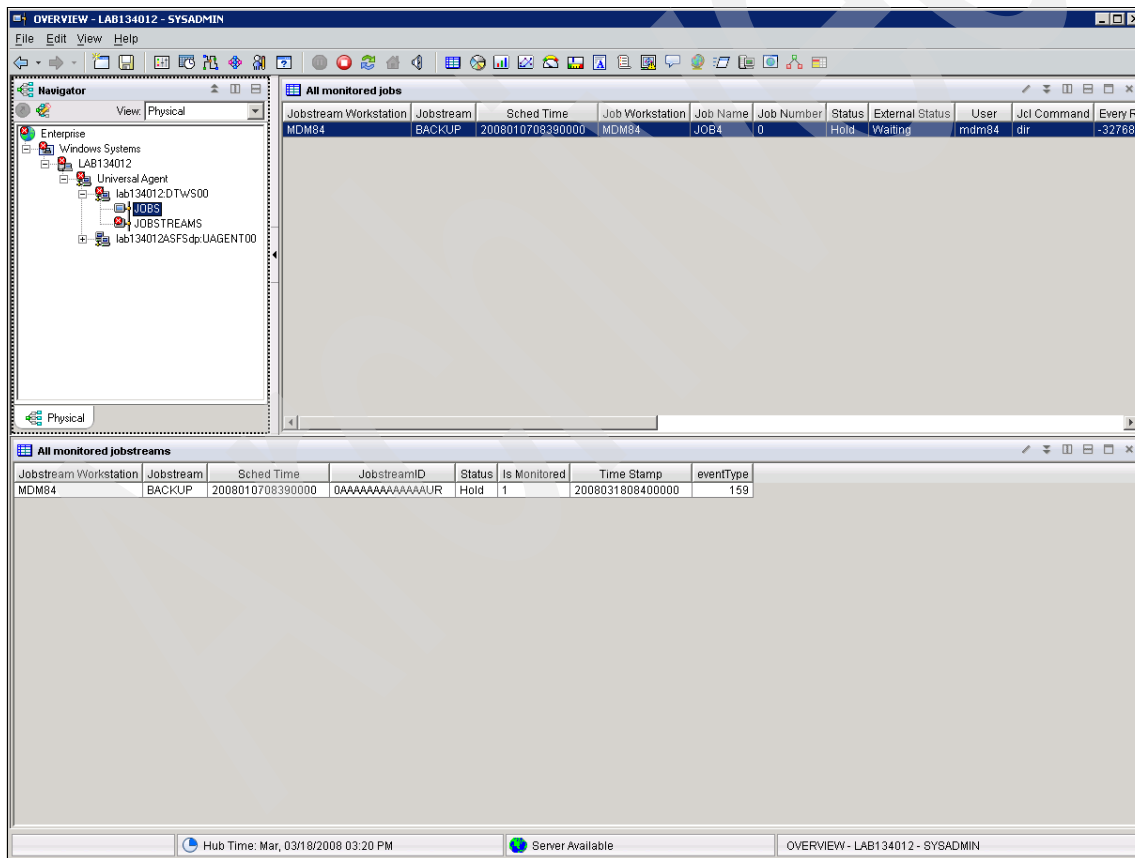


Figure 7-3 Sample monitoring for Tivoli Workload Scheduler

7.8 IBM Tivoli Netcool/Impact self-monitoring agent

Self monitoring is a feature that allows IBM Tivoli Netcool/Impact to monitor aspects of its own runtime performance and then to report the performance information to the Object Server as Netcool events.

This feature allows IBM Tivoli Netcool/Impact to monitor the following aspects of its performance:

- ▶ Memory status usage
- ▶ Event queue size
- ▶ Data source status
- ▶ Cluster status

The self-monitoring service is an IBM Tivoli Netcool/Impact service that manages the self-monitoring feature. This service is named `SelfMonitoring` and is created automatically when you install IBM Tivoli Netcool/Impact. You cannot create new instances of this service. To set up the self-monitoring feature, you configure the `SelfMonitoring` service using the IBM Tivoli Netcool/Impact GUI or command-line interface (CLI).

If you run multiple IBM Tivoli Netcool/Impact servers in a cluster, self monitoring operates on a per-server basis. Therefore, you must configure each server instance separately, and each server runs the self-monitoring feature independently of the other cluster members. Service configuration information is not propagated between members of the cluster, allowing you to customize self monitoring to accommodate each system where you run IBM Tivoli Netcool/Impact. Self monitoring includes these components:

- ▶ Memory status monitoring is the process by which the self-monitoring service checks the available Java and system memory and sends events to the Object Server regarding the memory status at intervals. IBM Tivoli Netcool/Impact monitors available memory in both the Java heap and in the system as a whole. The memory status is significant, because Netcool/Impact fails and reports an out-of-memory error if the maximum available memory is exceeded:
 - Java memory status
 - System memory status
 - Combined memory status
- ▶ Queue size monitoring is the process in which the self-monitoring service checks the size of event reader event queues at intervals and sends events to the Object Server regarding the event queue status. For each event, the self-monitoring service calculates the severity by determining the rate at which the queue size is increasing or decreasing since the last interval point.

Queue size monitoring monitors the following services:

- OMNIbusEventReader
 - DatabaseEventReader
 - OMNIbusEventListener
 - DatabaseEventListener
 - JMSMessageListener
- Data source monitoring is a process in which the self-monitoring service sends events to Object Server that report the status of IBM Tivoli Netcool/Impact database connections. IBM Tivoli Netcool/Impact supports data source monitoring for the following SQL database data sources:
- DB2 data source adapter (DSA)
 - Informix® DSA
 - MySQL™ DSA
 - Object Server DSA
 - Open Database Connectivity (ODBC) DSA
 - Oracle® DSA
 - PostgreSQL DSA
 - SQL Server® DSA
 - Sybase DSA
 - GenericSQL DSA
- Cluster status monitoring is a process in which the self-monitoring service sends events to Object Server that report the status of members in a IBM Tivoli Netcool/Impact server cluster.

The self-monitoring service must be running on primary and secondary cluster members in order for this feature to report all possible status data to the Object Server. You can start the self-monitoring service on secondary cluster members using the IBM Tivoli Netcool/Impact CLI.



Part 3

Scenario walk-through

In this part, we describe how to apply the integration of Tivoli products in specific scenarios. We choose several realistic scenarios and apply them in our environment.

Operational drill down

In this chapter, we provide a scenario where an operator can assess the overall system health and get more detail by drilling down from an operational console, Tivoli Enterprise Portal, into other products to view related information.

We discuss the following topics:

- ▶ 8.1, “Scenario overview” on page 342
- ▶ 8.2, “Scenario setup” on page 343
- ▶ 8.3, “Sample walk-through” on page 355
- ▶ 8.4, “Summary of benefits” on page 363

8.1 Scenario overview

This scenario describes an operator's dashboard view of the overall health of the IT environment. This view also allows the operator to drill down on a potential problem, to collect more information about a particular resource from various Tivoli operational management tools, and to navigate between them. The operational drill down flow that we use works from Tivoli Enterprise Portal operational workspaces and built-in Web browser launches to other management tools.

Figure 8-1 depicts the scenario flow.

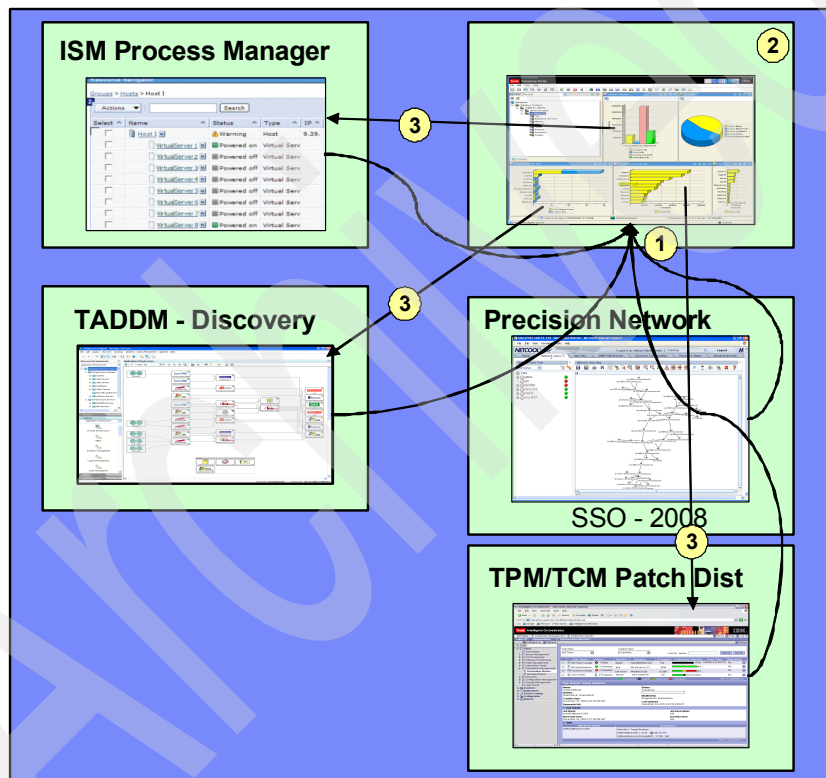


Figure 8-1 Scenario flow

We use these products in this scenario:

- ▶ IBM Tivoli Monitoring monitors resources (managed systems) and sends alerts when given conditions are met. Tivoli Enterprise Portal is the name of its graphical user interface. This is the primary operation console in this scenario.
- ▶ IBM Tivoli Application and Dependency Discovery Manager uncovers application dependencies and configurations.
- ▶ IBM Tivoli Provisioning Manager is built on a service-oriented architecture (SOA). This provisioning software enhances the usability for executing changes while keeping server and desktop software compliant.
- ▶ IBM Tivoli Netcool Network Manager For IP provides the network analysis software that is needed to manage complex networks.

8.2 Scenario setup

In this section, we discuss the scenario setup that we performed. Part 2, “Implementation” on page 15 discusses the scenario generally. These steps are necessary for this scenario:

1. Set up security using LDAP authentication for single sign-on (SSO):
 - IBM Tivoli Provisioning Manager in 4.4, “Tivoli Process Automation Engine security setup” on page 73
 - IBM Tivoli Application Dependency Discovery Manager in 4.5, “IBM Tivoli Application Dependency Discovery Manager security setup” on page 92
 - IBM Tivoli Network Manager for IP in 4.6, “IBM Tivoli Netcool products LDAP configuration” on page 95
 - IBM Tivoli Monitoring in 4.7, “IBM Tivoli Monitoring” on page 120
2. Exchange resource data to establish the application context that is depicted in the following sections:
 - 5.2.1, “IBM Tivoli Integration Composer” on page 136
 - 5.2.2, “Promoting Actual CIs to Authorized CIs” on page 137
 - 5.2.3, “IBM Tivoli Provisioning Manager integration” on page 142
 - 5.2.4, “IBM Tivoli Monitoring integration” on page 150
 - 5.2.6, “IBM Tivoli Network Manager for IP integration” on page 163

3. We describe providing launching out from Tivoli Enterprise Portal to other applications in 6.3.3, “IBM Tivoli Monitoring” on page 277. We describe building the specific URLs in these sections:
 - IBM Tivoli Application Dependency Discovery Manager in 6.2.1, “IBM Tivoli Application Dependency Discovery Manager” on page 233
 - IBM Tivoli Provisioning Manager in 6.2.2, “Tivoli Process Automation Engine-based products” on page 235
 - IBM Tivoli Network Manager for IP in 6.2.4, “Tivoli Integrated Portal” on page 249

We describe more detailed integration connectivity in this section. We discuss these specific integrations:

- ▶ 8.2.1, “Defining a new dashboard workspace” on page 344
- ▶ 8.2.2, “Launching an IBM Tivoli Application Dependency Discovery Manager window within Tivoli Enterprise Portal” on page 347
- ▶ 8.2.3, “Launching IBM Tivoli Provisioning Manager” on page 350
- ▶ 8.2.4, “Launching IBM Tivoli Network Manager for IP” on page 353

8.2.1 Defining a new dashboard workspace

We choose to use workspaces with URL links for the launching out invocation. The workspaces then can be linked from other Tivoli Enterprise Portal pages. It is beneficial to create the workspace to be as generic as possible so that it can be created one time and distributed to all monitored servers. The following steps describe the generic procedure to build the workspace:

1. You must identify a suitable place for the configuration information. We use the Operating System-level navigator item, because one exists on every server. Navigate to the desired node in the Linux Operating System agent and create the Web browser workspace, as seen in Figure 8-2 on page 345.

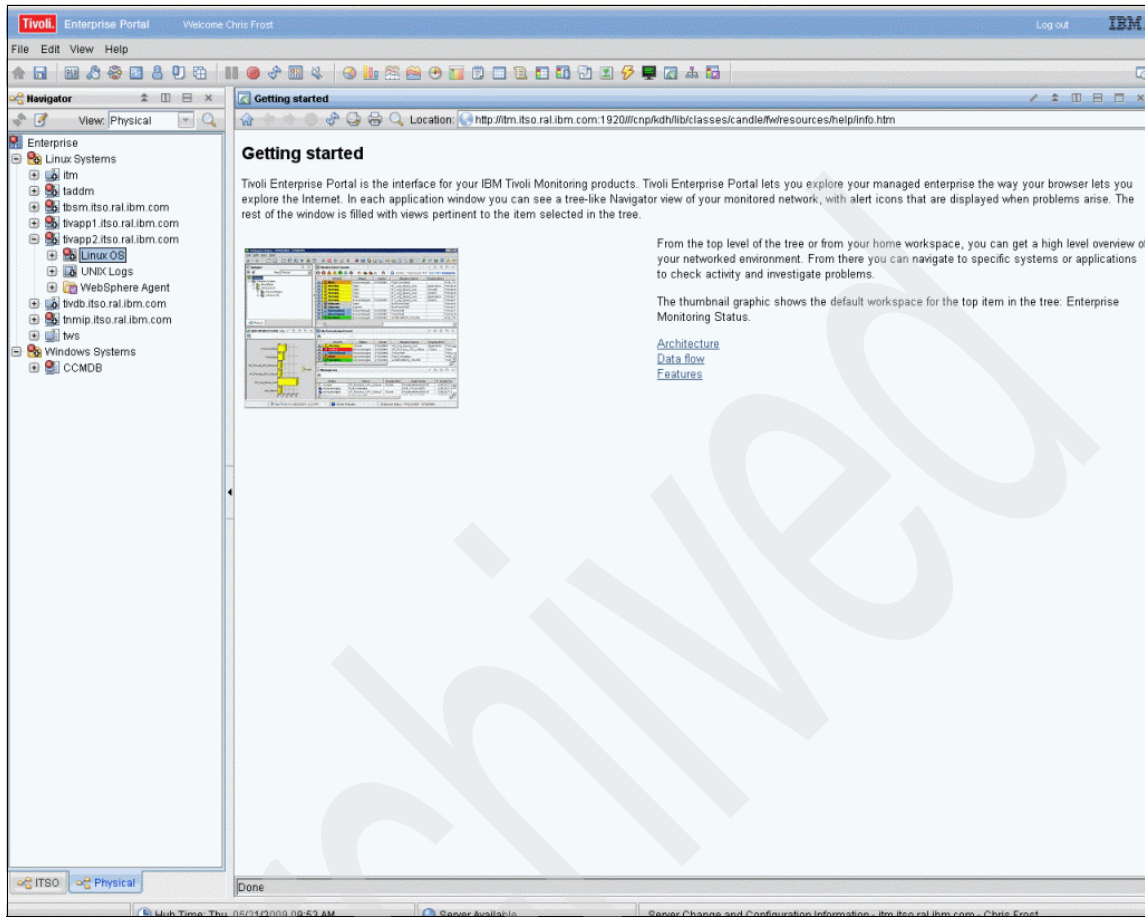


Figure 8-2 Tivoli Enterprise Portal browser window

2. Enter the URL of the Tivoli application to which you will link. Edit the properties of the browser window, and enter the URL in the space provided. Select **Use Provided Location**. The properties window now looks similar to Figure 8-3 on page 346. Click **OK**.

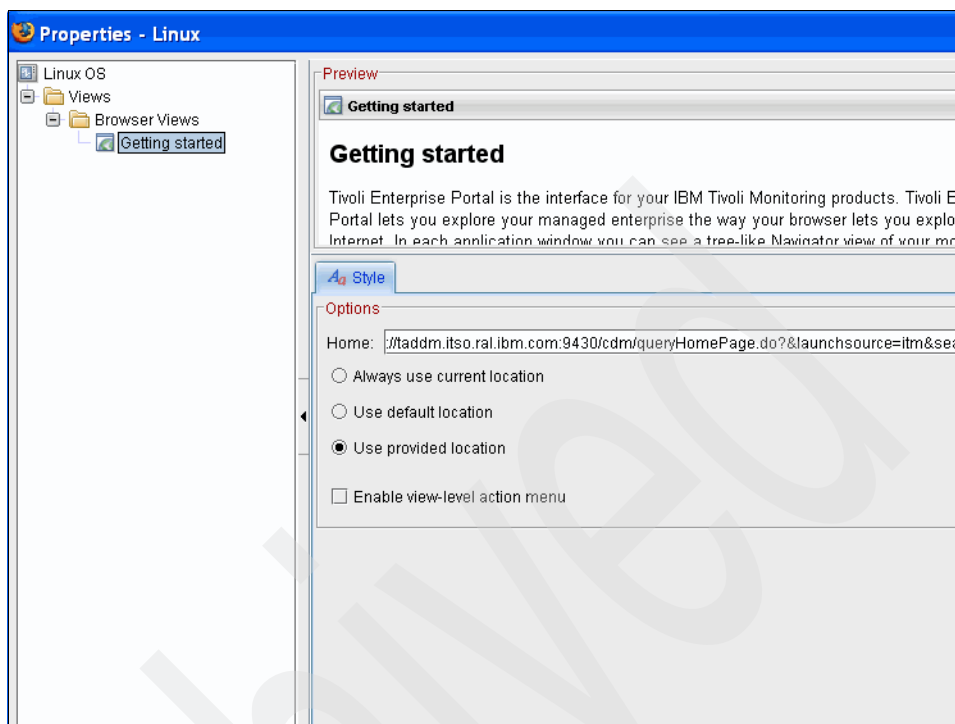


Figure 8-3 Workspace Properties window

3. Save the workspace by clicking **File** → **Save Workspace As**. Choose a suitable name for the workspace (Figure 8-4).

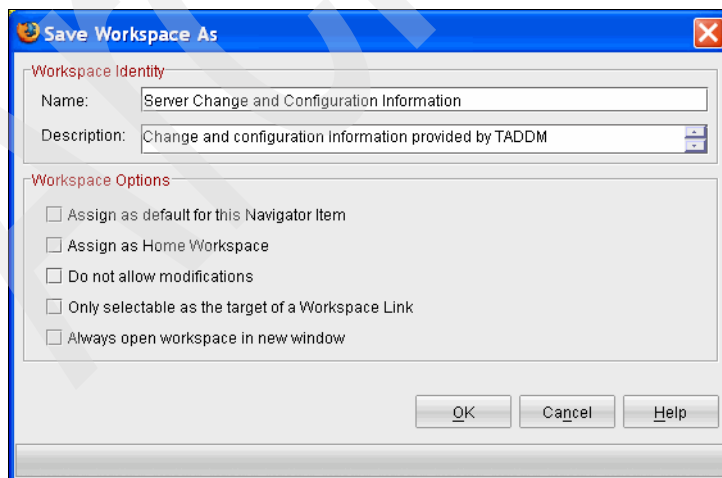


Figure 8-4 Save Workspace window

4. Navigate to another page, and then return to the newly created workspace, as seen in Figure 8-5.

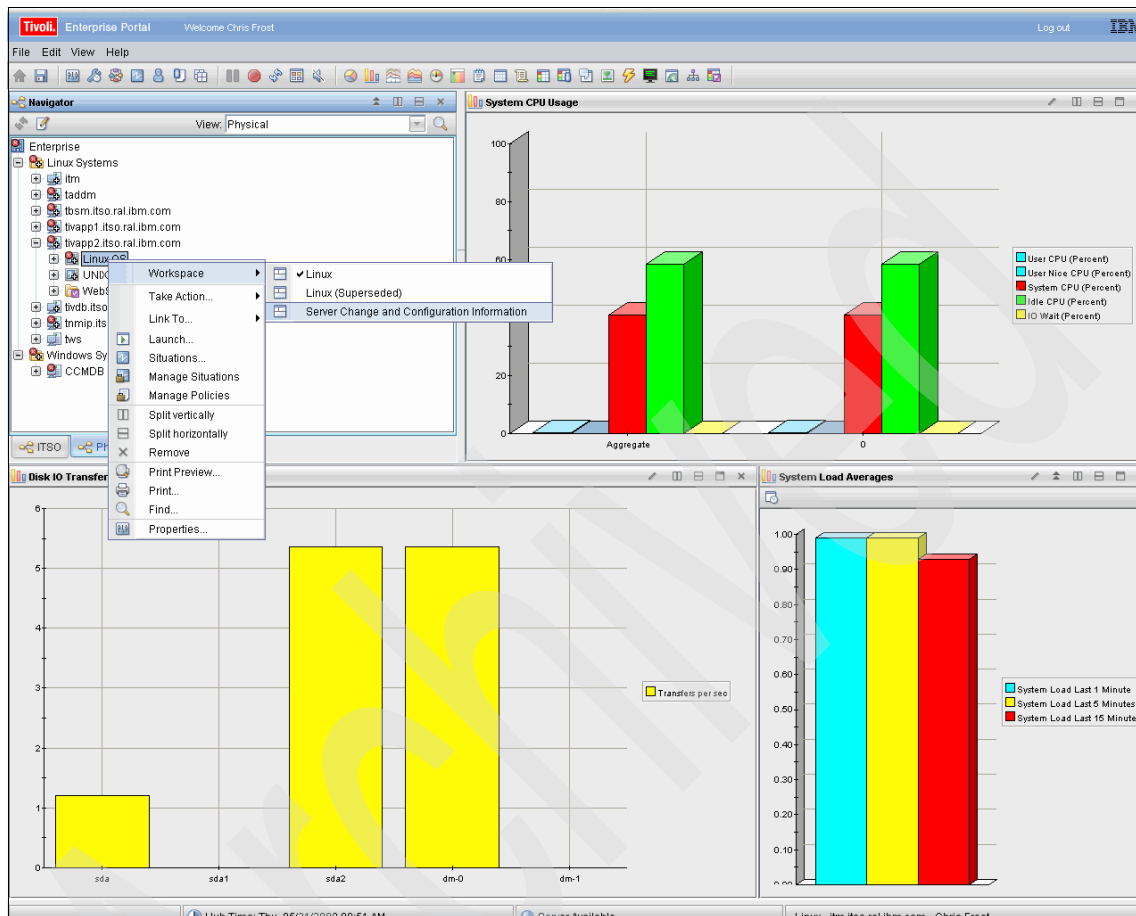


Figure 8-5 Tivoli Enterprise Portal Workspace menu

8.2.2 Launching an IBM Tivoli Application Dependency Discovery Manager window within Tivoli Enterprise Portal

In this section, you see Tivoli Enterprise Portal to IBM Tivoli Application Dependency Discovery Manager integration showing a IBM Tivoli Application Dependency Discovery Manager window within Tivoli Enterprise Portal. Automatic sign-on to IBM Tivoli Application Dependency Discovery Manager is done by using Tivoli Enterprise Portal user credentials. Note that the host name in IBM Tivoli Application Dependency Discovery Manager corresponds to the

system in Tivoli Enterprise Portal from which we launched the link. Edit the properties of Tivoli Enterprise Portal browser window to enter the following URL:

`http://taddm.itso.ral.ibm.com:9430/cdm/queryHomePage.do?&launchsource=itm&searchtext=$-1038$`

This URL launches a IBM Tivoli Application Dependency Discovery Manager page in Tivoli Enterprise Portal. The variable, `$-1038$`, resolves to the IP address of the current navigator item in Tivoli Enterprise Portal. This IP address is then searched in IBM Tivoli Application Dependency Discovery Manager, revealing a page with the correct configuration item, as seen in Figure 8-6.

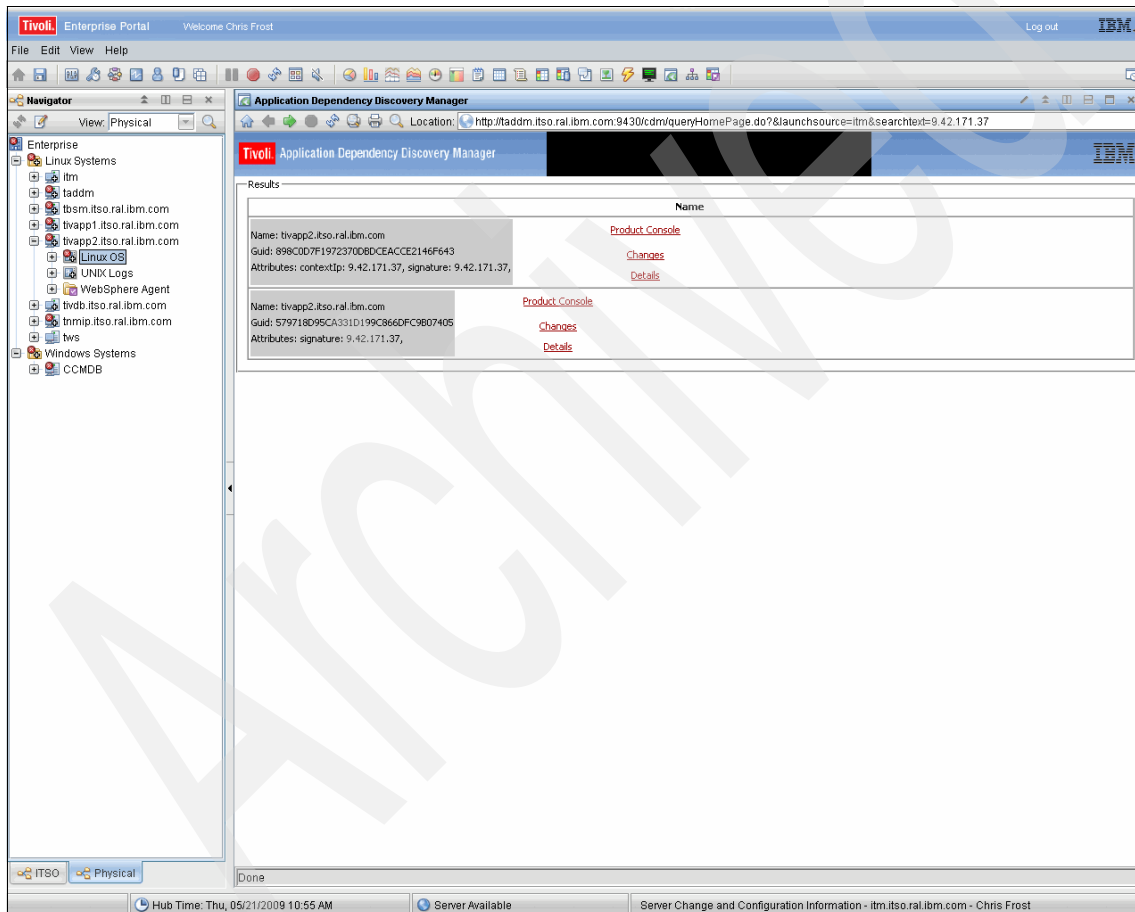


Figure 8-6 IBM Tivoli Application Dependency Discovery Manager window in Tivoli Enterprise Portal

From this window, you can launch to three places:

- ▶ IBM Tivoli Application Dependency Discovery Manager Console opens a separate Java Web-start console for IBM Tivoli Application Dependency Discovery Manager, allowing the operator to view dependencies, configurations, and other functions.
- ▶ Details opens the configuration details for the server.
- ▶ The changes link navigates to the list of changes for the server, while remaining within Tivoli Enterprise Portal, as shown in Figure 8-7.

The screenshot displays the Tivoli Enterprise Portal interface. On the left, the 'Navigator' pane shows a tree structure under 'Enterprise' with categories like 'Linux Systems' and 'Windows Systems'. The main area is titled 'Application Dependency Discovery Manager' and shows a table of changes for the component 'bvapp2.itso.ral.ibm.com'. The table has columns for Component, Change, Date, Attribute, Old, and New. The status bar at the bottom indicates 'Hub Time: Thu, 05/21/2009 10:56 AM' and 'Server Available'.

Component	Change	Date	Attribute	Old	New
	Created	Thursday, April 30, 2009 3:15:34 PM EDT			
bvapp2.itso.ral.ibm.com	Updated	Tuesday, May 5, 2009 3:47:41 PM EDT	manufacturer	VMware, Inc.	VMware
bvapp2.itso.ral.ibm.com	Updated	Tuesday, May 5, 2009 3:47:42 PM EDT	model	VMware Virtual Platform	None
bvapp2.itso.ral.ibm.com	Updated	Tuesday, May 5, 2009 3:47:42 PM EDT	primaryMACAddress		000C29F5ACBF
bvapp2.itso.ral.ibm.com	Updated	Tuesday, May 5, 2009 3:47:42 PM EDT	memorySize	1905262592	1905278976
bvapp2.itso.ral.ibm.com	Updated	Tuesday, May 5, 2009 3:47:42 PM EDT	label		bvapp2.itso.ral.ibm.com
bvapp2.itso.ral.ibm.com	Updated	Tuesday, May 5, 2009 3:47:42 PM EDT	sourceToken		6174
bvapp2.itso.ral.ibm.com	Updated	Tuesday, May 5, 2009 3:47:42 PM EDT	name	Linux:	Red Hat Enterprise Linux AS release 4
bvapp2.itso.ral.ibm.com	Updated	Tuesday, May 5, 2009 3:47:42 PM EDT	OSVersion	Red Hat Enterprise Linux AS release 4 (Nahant Update 4)	4.0
bvapp2.itso.ral.ibm.com	Updated	Tuesday, May 5, 2009 3:47:42 PM EDT	sourceToken		3645-6174
bvapp2.itso.ral.ibm.com	Updated	Tuesday, May 5, 2009 3:47:42 PM EDT	OSName	Linux:	LINUX

Figure 8-7 Changes viewed in Tivoli Enterprise Portal

8.2.3 Launching IBM Tivoli Provisioning Manager

In this section, we use Tivoli Enterprise Portal as a starting point to create a link to IBM Tivoli Provisioning Manager. To launch IBM Tivoli Provisioning Manager from Tivoli Enterprise Portal, you need to create a workspace in the same way as when launching IBM Tivoli Application Dependency Discovery Manager from Tivoli Enterprise Portal.

The only difference is that the link for IBM Tivoli Provisioning Manager must be this URL:

```
https://ccmdb.itso.ra1.ibm.com:9045/maximo/ui/?event=loadapp&value=tpdc  
mfind&additionalevent=useqbe&additionaleventvalue=Name=$-1071$
```

You must replace `ccmdb.itso.ra1.ibm.com:9045` with your IBM Tivoli Provisioning Manager server host name and port.

Now, you see a IBM Tivoli Provisioning Manager window within Tivoli Enterprise Portal, as shown in Figure 8-8 on page 351. Sign-on to IBM Tivoli Provisioning Manager occurs automatically by using Tivoli Enterprise Portal user credentials. The context that is passed from Tivoli Enterprise Portal to IBM Tivoli Provisioning Manager is the managed system host name (variable `$-1071$`).

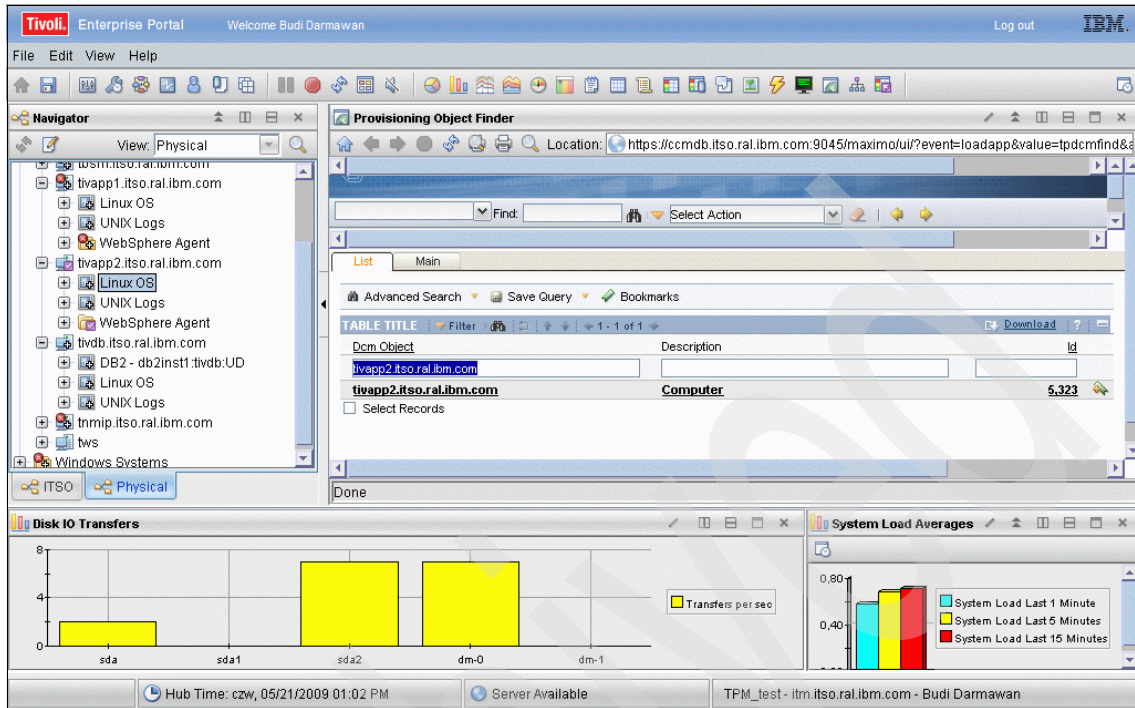


Figure 8-8 IBM Tivoli Provisioning Manager window inside Tivoli Enterprise Portal workspace

You can then click a host name in the IBM Tivoli Provisioning Manager window to see general information on that computer, as shown in Figure 8-9 on page 352.

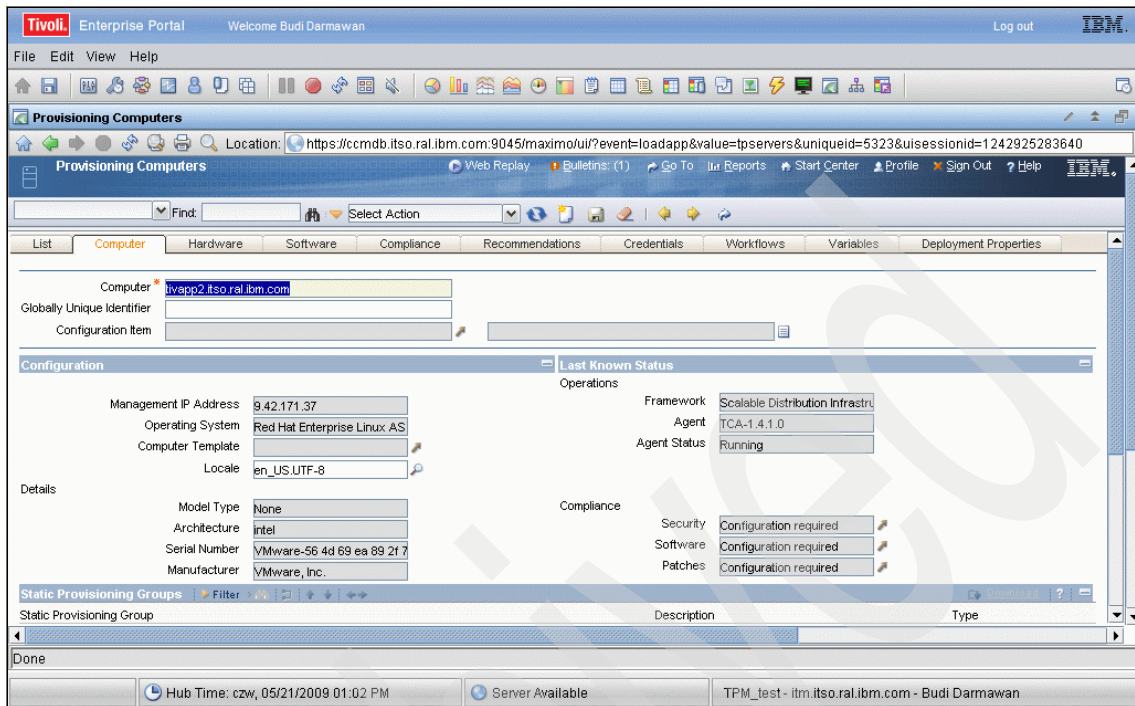


Figure 8-9 Computer information

On the Software tab, you can view the software that is installed on this computer (Figure 8-19 on page 363).

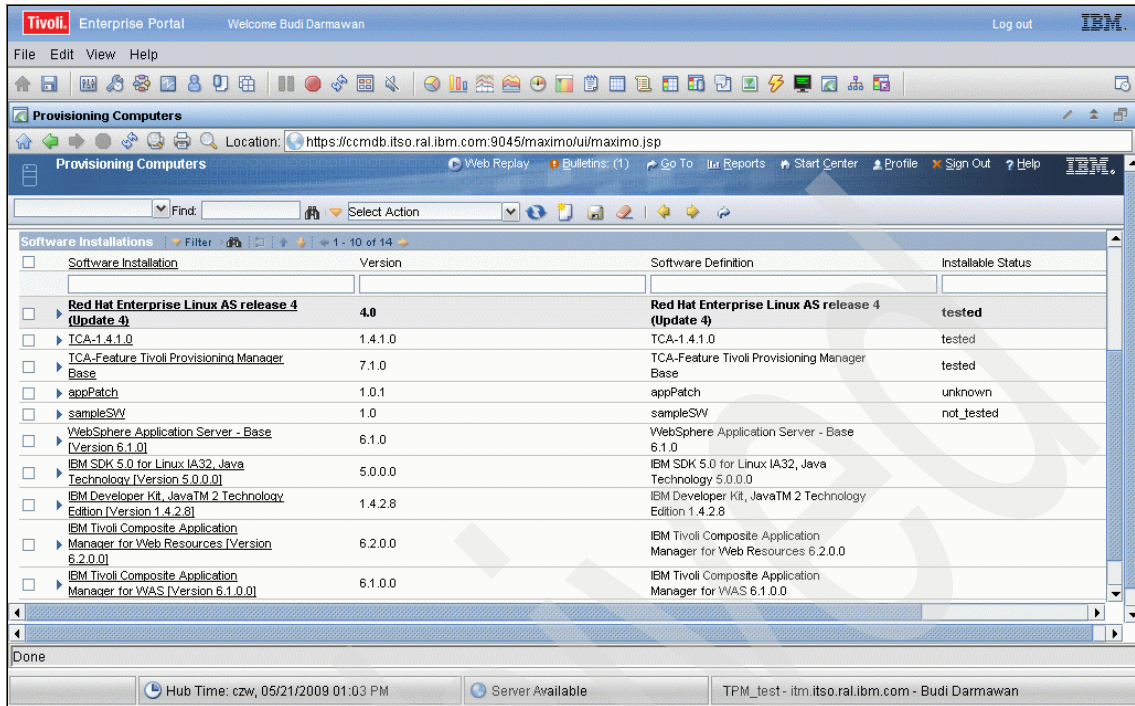


Figure 8-10 Software tab

8.2.4 Launching IBM Tivoli Network Manager for IP

In this section, we use Tivoli Enterprise Portal as a starting point to create a link to IBM Tivoli Network Manager for IP. To launch IBM Tivoli Network Manager for IP from Tivoli Enterprise Portal, you need to create a workspace in the same way as when launching IBM Tivoli Application Dependency Discovery Manager from Tivoli Enterprise Portal.

The only difference is that the link for IBM Tivoli Network Manager for IP is this URL:

[https://tnmip.itso.ral.ibm.com:16316/ibm/console/ncp_topoviz/HopView.do?hops=2&domain=ITS0&connectivity=layer3&seed=\\$-1071\\$](https://tnmip.itso.ral.ibm.com:16316/ibm/console/ncp_topoviz/HopView.do?hops=2&domain=ITS0&connectivity=layer3&seed=-1071)

This URL launches the IBM Tivoli Network Manager for IP Hop View in Tivoli Enterprise Portal. The \$-1071\$ variable resolves to the device node name of the current navigator item in Tivoli Enterprise Portal. This IP address is then located in IBM Tivoli Network Manager for IP, revealing a page showing the device topology.

Now, you see an IBM Tivoli Network Manager for IP window within Tivoli Enterprise Portal, as shown in Figure 8-11. Sign-on to IBM Tivoli Network Manager for IP occurs automatically by using Tivoli Enterprise Portal user credentials, as described in 4.6, “IBM Tivoli Netcool products LDAP configuration” on page 95. The context passed from Tivoli Enterprise Portal to IBM Tivoli Network Manager for IP is the managed system host name (variable \$-1071\$).

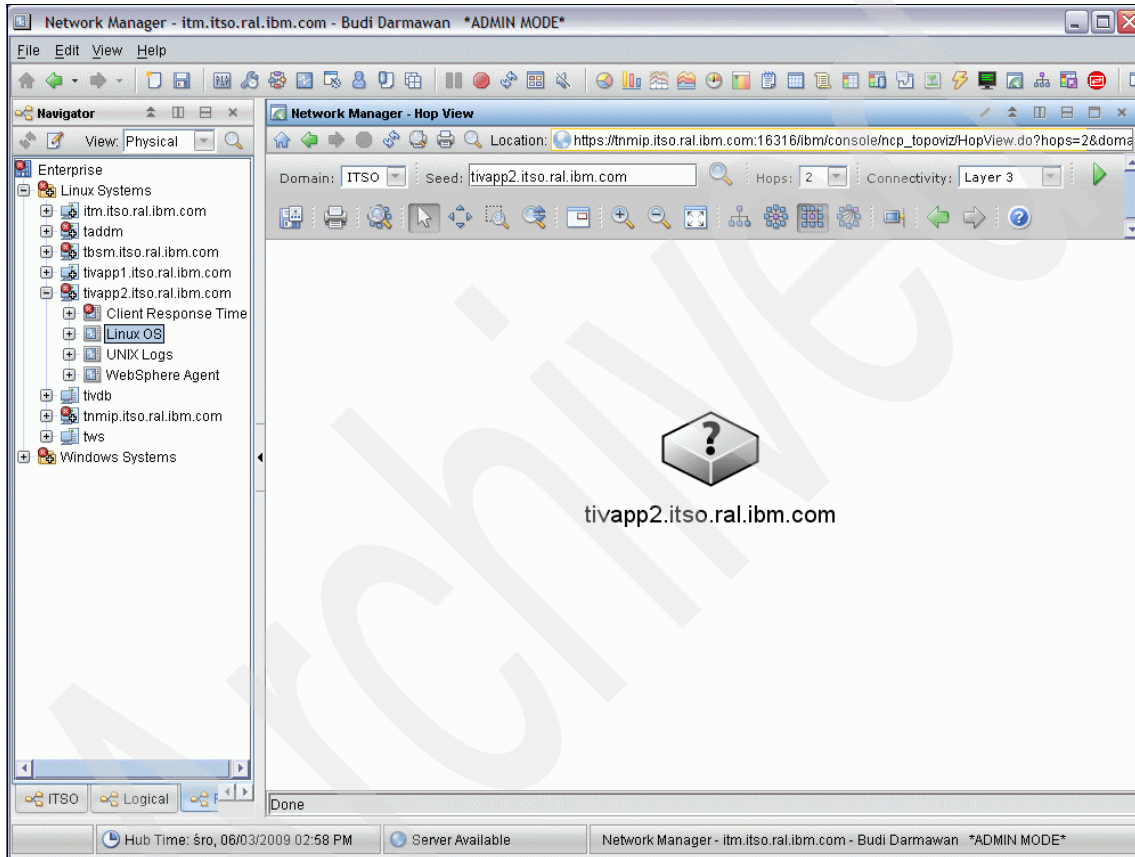


Figure 8-11 IBM Tivoli Network Manager for IP in Tivoli Enterprise Portal workspace

You can then click a device icon in the IBM Tivoli Network Manager for IP window to see more device-specific information.

Important: There is an open defect for launching the IBM Tivoli Network Manager for IP Hop View report from Tivoli Enterprise Portal when using Internet Explorer 7.0, Firefox 2.0, or Firefox 3.0. Based on problem management record (PMR) 66538, you can display the textual representation of the topology view using this URL:

```
https://<HOST>:<PORT>/ibm/console/ncp_mibbrowser/Launch.do?domain=<DOMAIN>&host=<IP>&variable=sysUpTime&resultsOnly=true
```

8.3 Sample walk-through

After you perform all the prerequisite tasks in 8.2, “Scenario setup” on page 343, the integration allows the execution of the scenario. In this section, we walk you through the scenario components.

This scenario starts with an operator viewing the current resource metrics for one server in Tivoli Enterprise Portal, that consist of the capacity usage information, as shown in Figure 8-12 on page 356.

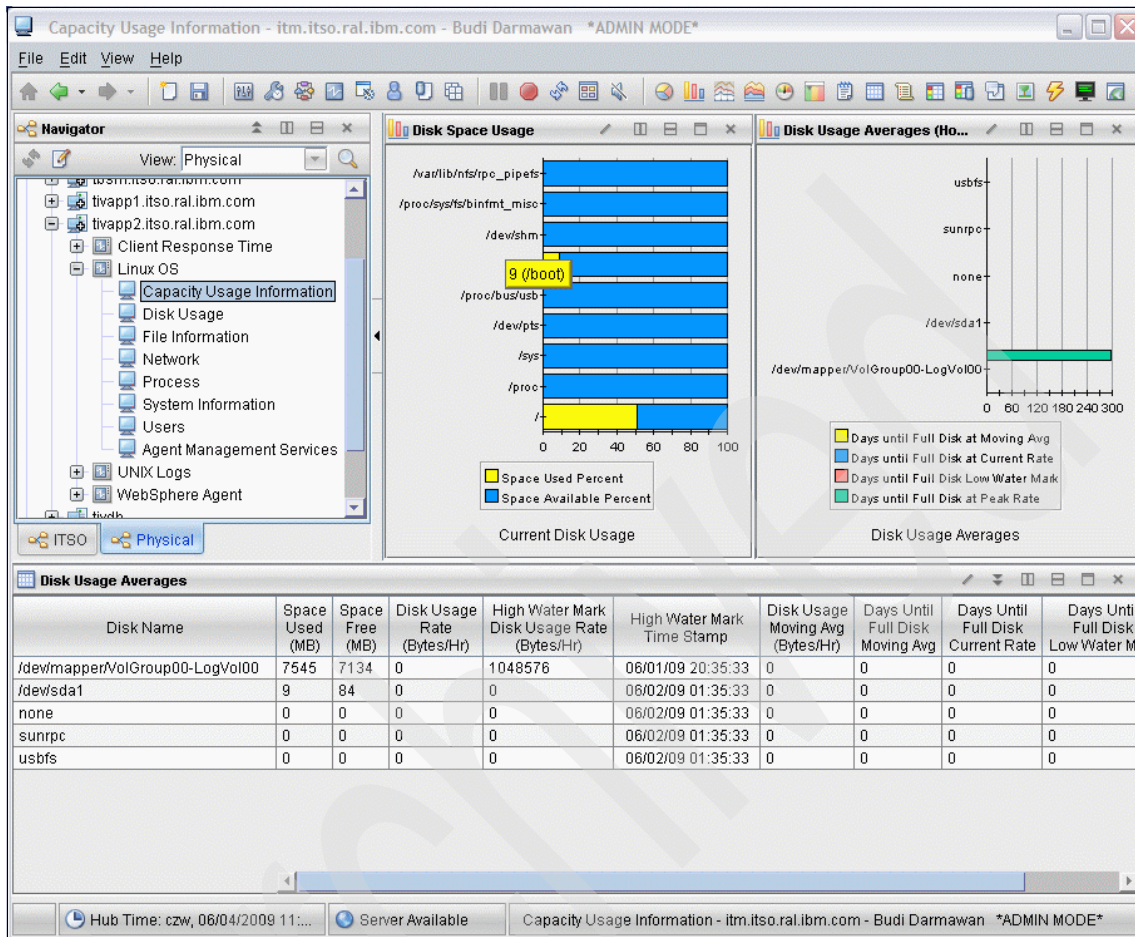


Figure 8-12 Capacity Usage Information in Tivoli Enterprise Portal

Then, the operator wants more information about the role that this server plays in the IT environment, so the operator goes to the IBM Tivoli Application Dependency Discovery Manager application. The operator switches to the IBM Tivoli Application Dependency Discovery Manager workspace that is created under Linux OS to see a IBM Tivoli Application Dependency Discovery Manager window inside Tivoli Enterprise Portal, as shown in Figure 8-13 on page 357.

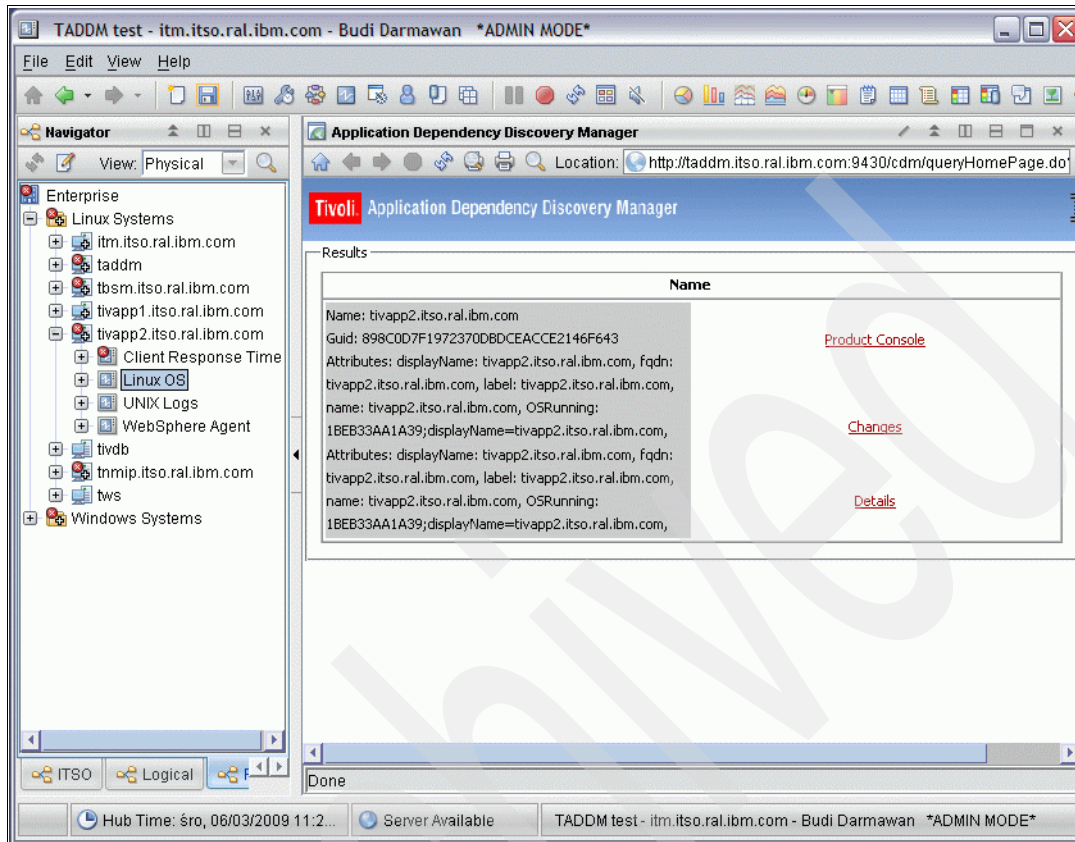


Figure 8-13 Workspace with IBM Tivoli Application Dependency Discovery Manager window in Tivoli Enterprise Portal

The operator clicks **Details** and clicks the **General** tab to see summary information about this configuration item, as shown in Figure 8-14 on page 358.

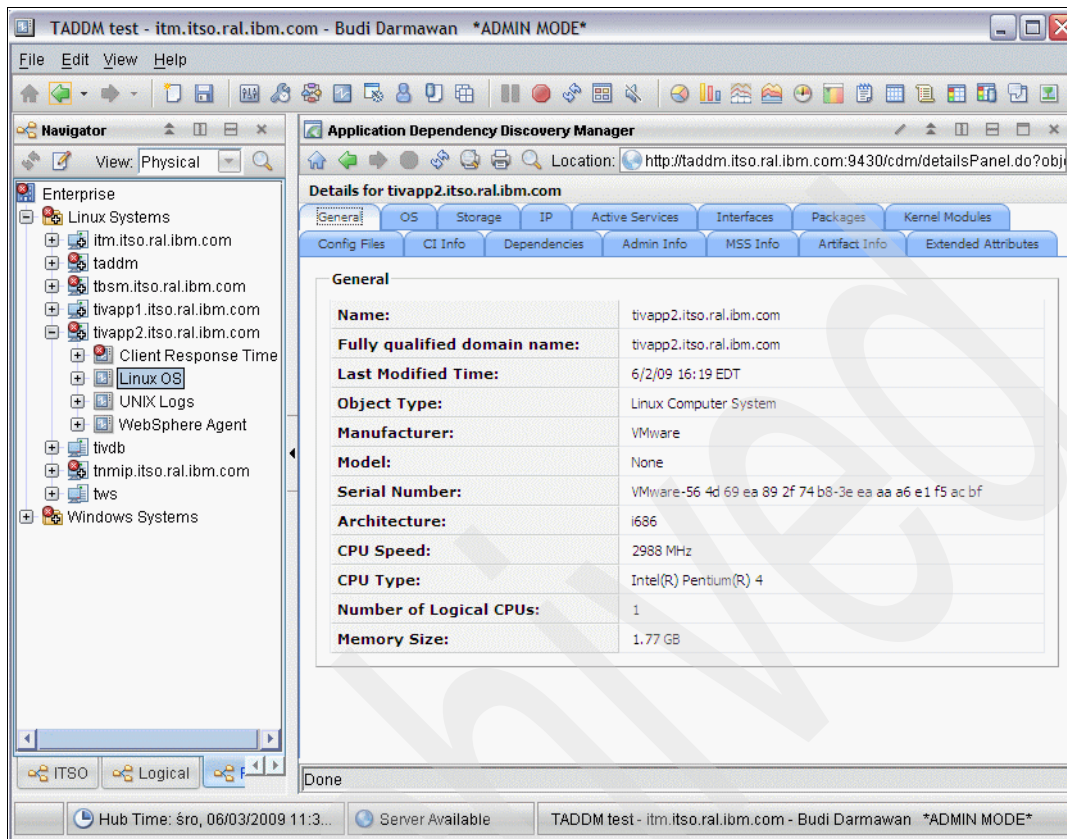


Figure 8-14 General tab in IBM Tivoli Application Dependency Discovery Manager

To understand the server's role, the operator clicks **Dependencies**. The operator is redirected to the tab that shows that this item is a part of the Trader business Service and that it contains the application server, WebSphere, Apache, and others (see Figure 8-15 on page 359).

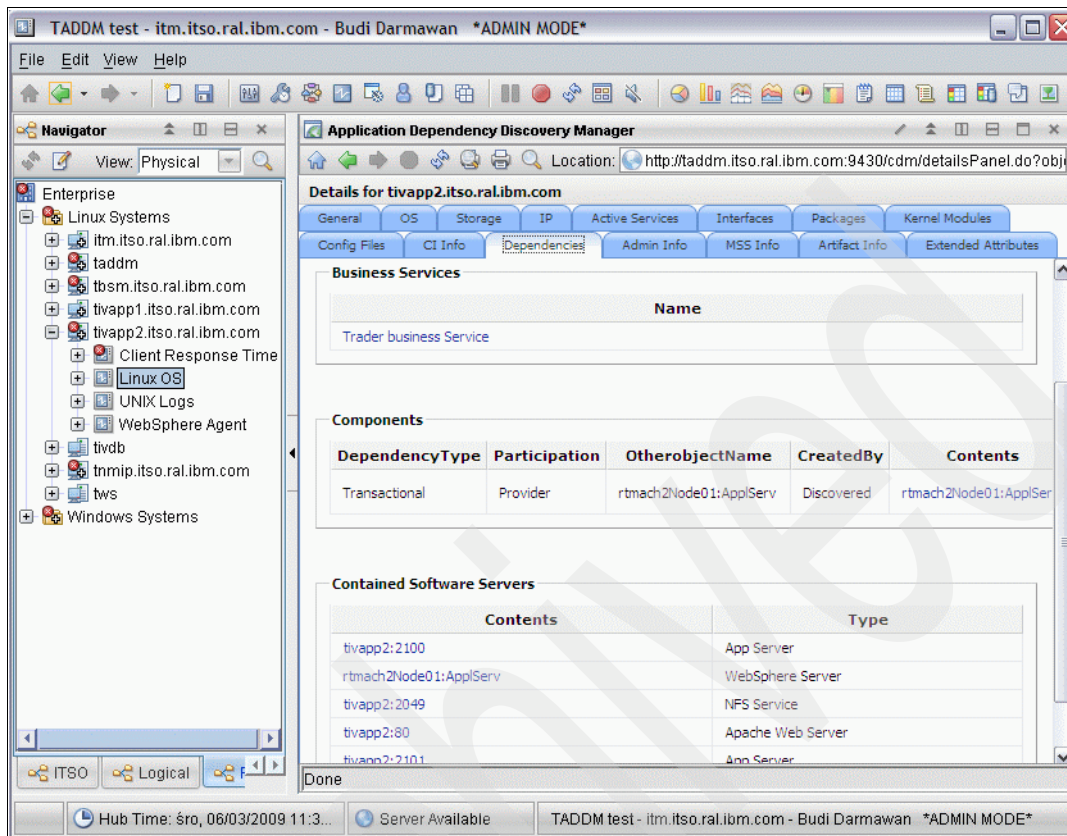


Figure 8-15 Dependencies tab in IBM Tivoli Application Dependency Discovery Manager

Then, the operator wants to see the software that is installed on this particular server and switches to the Tivoli Provisioning Manager workspace (under Linux OS) to view the software list (Figure 8-16 on page 360).

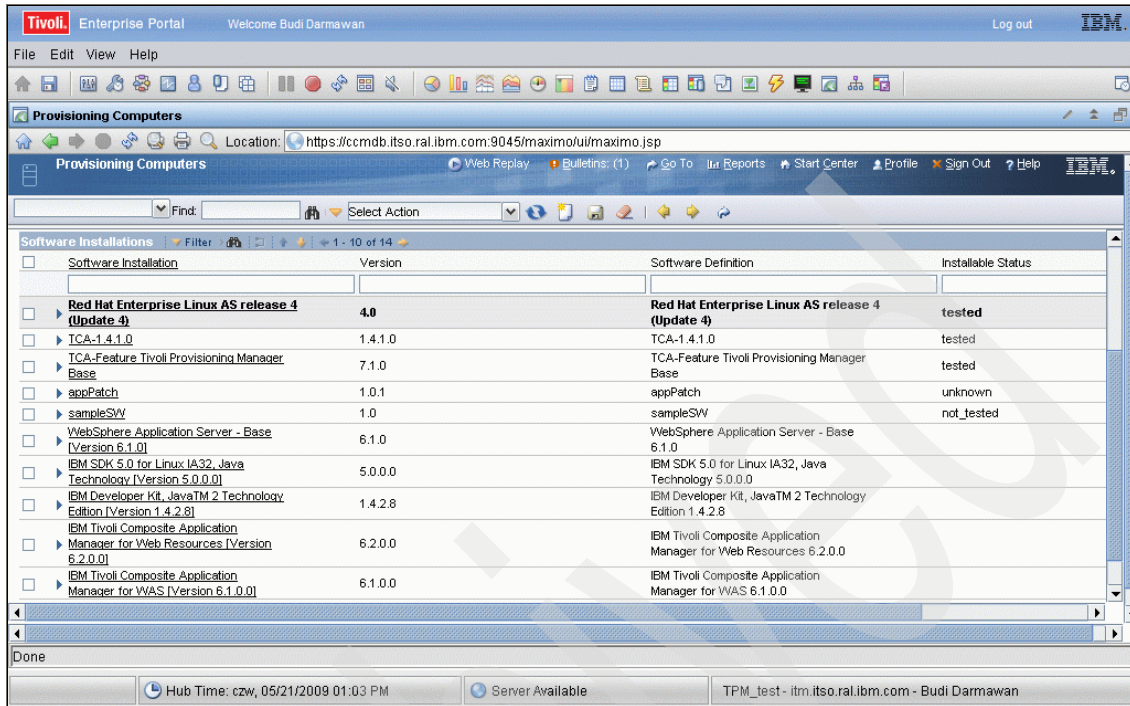


Figure 8-16 List of software installations from IBM Tivoli Provisioning Manager

The operator can view more information about the item in IBM Tivoli Provisioning Manager, depending on the operator's current needs. The operator then goes to the Network workspace to view the networking information of the node (from Tivoli Monitoring), as shown in Figure 8-17 on page 361.

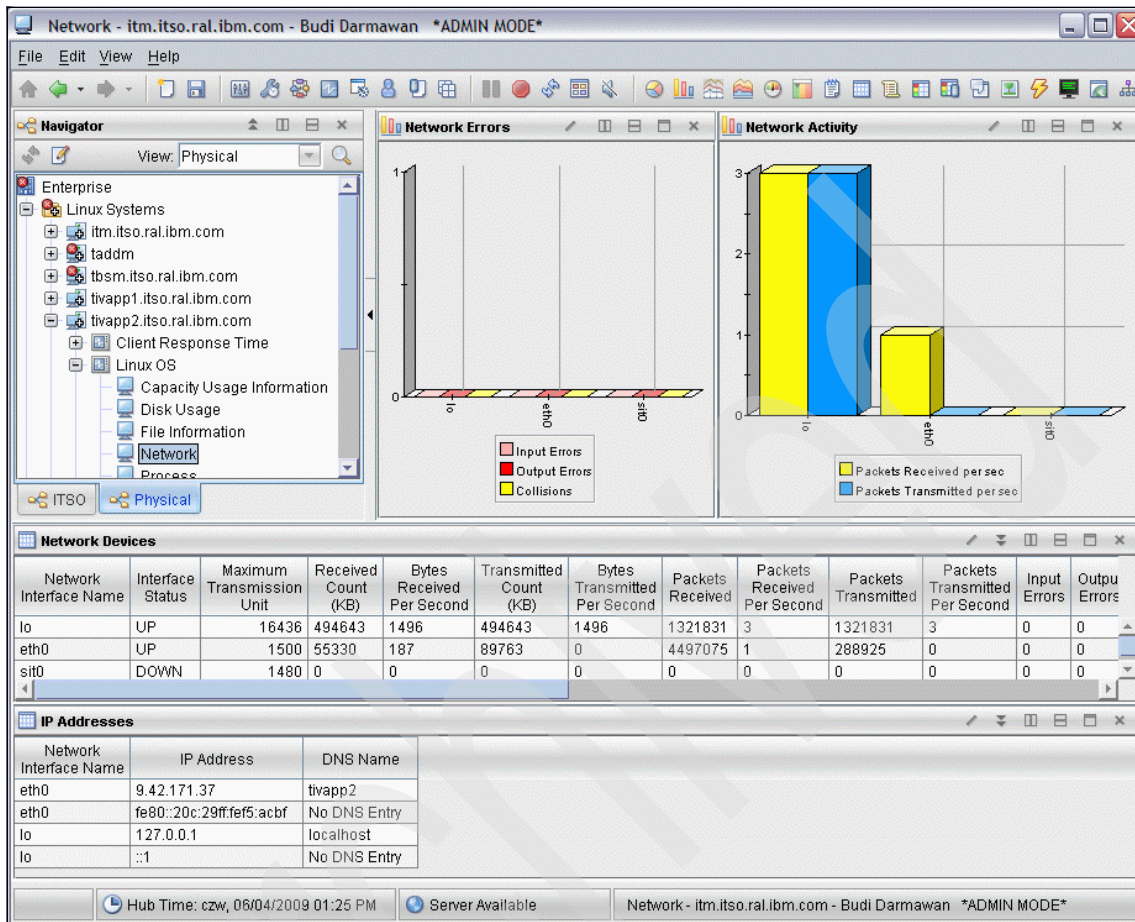


Figure 8-17 Network workspace in Tivoli Enterprise Portal

Then, the operator goes to IBM Tivoli Network Manager for IP by switching to the Network Manager workspace (see Figure 8-18 on page 362).

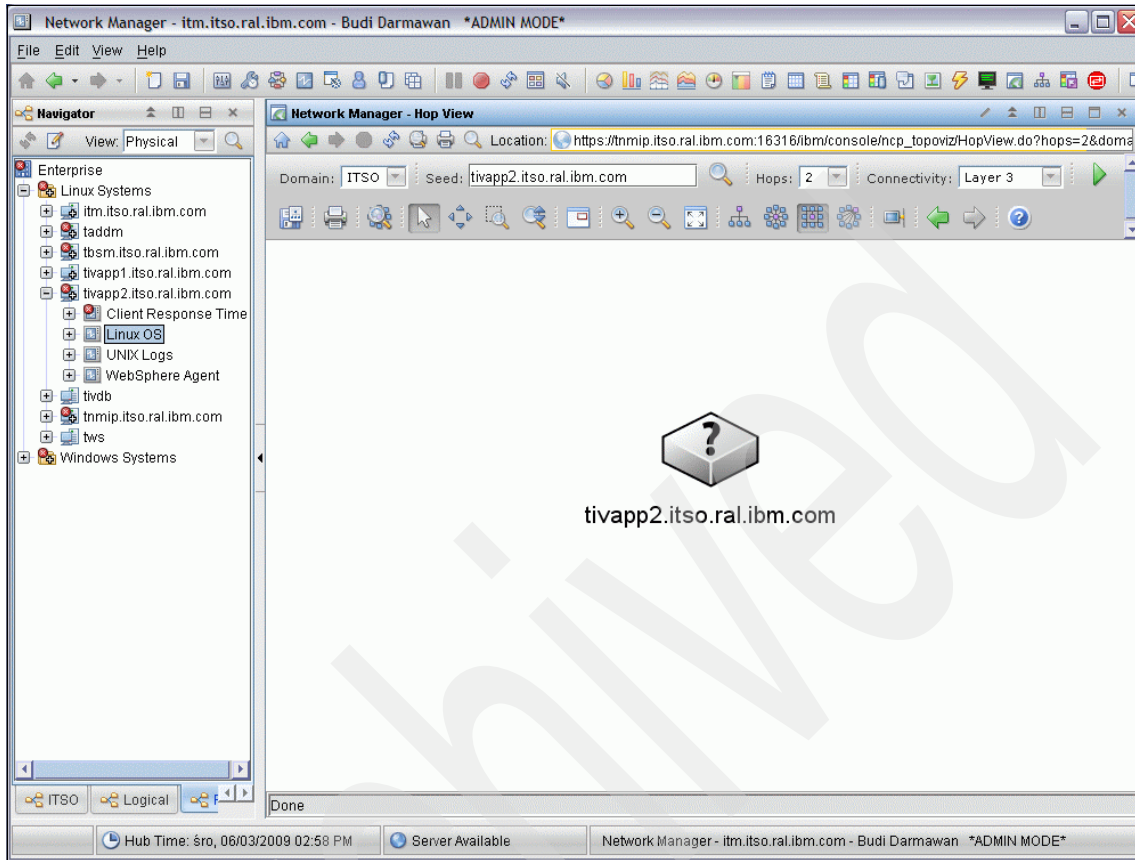


Figure 8-18 Network node viewed in IBM Tivoli Network Manager for IP inside Tivoli Enterprise Portal

Then, the operator views the structure of the network node, as shown in Figure 8-19 on page 363. Other information about the network can be displayed by the Network Manager.

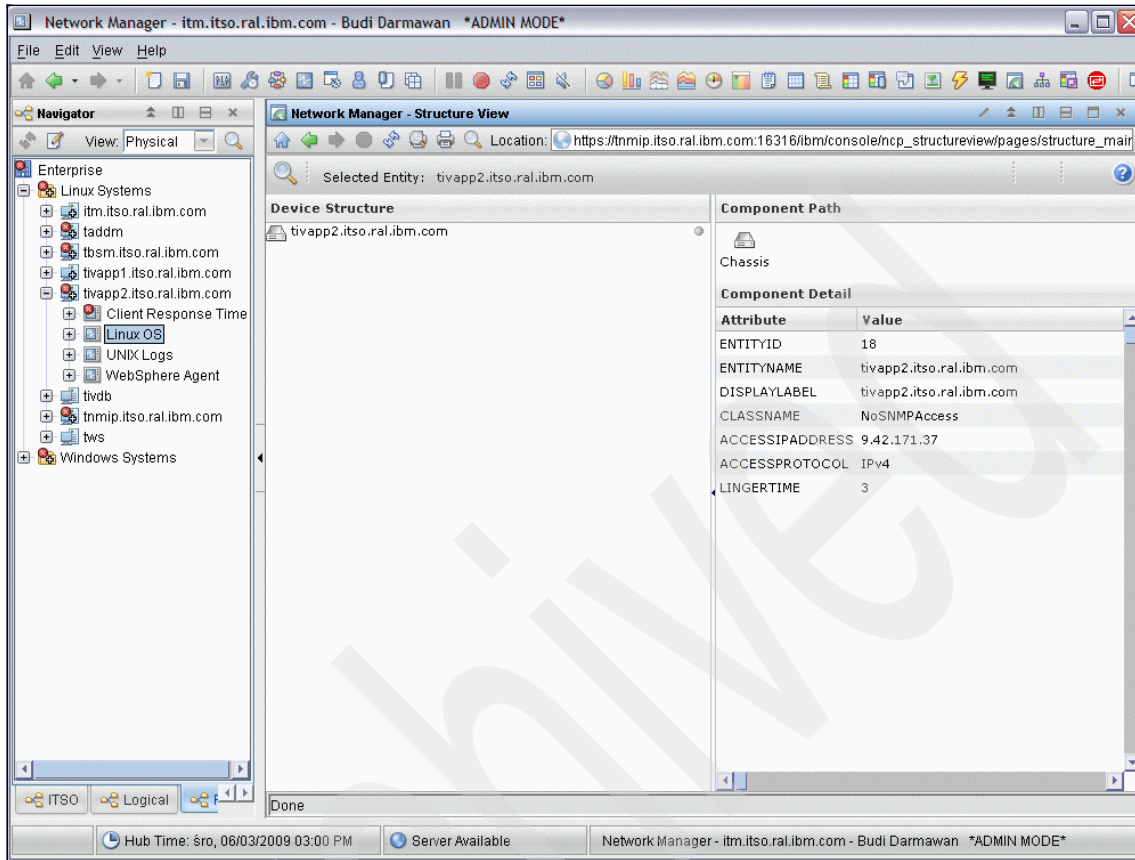


Figure 8-19 Node structure

8.4 Summary of benefits

The benefit of having Tivoli products work together is that the same resource can be viewed from various perspectives: an operator can look at a resource as part of a network (in IBM Tivoli Network Manager for IP), then, as an element of application infrastructure (IBM Tivoli Application Dependency Discovery Manager), or as the target of software distribution or as a software inventory (IBM Tivoli Provisioning Manager).

Having all this information in one place (Tivoli Enterprise Portal) and the ability to launch in context makes the job of the administrator and the job of the operator easier. It is also less error-prone, because the name of the resource is automatically passed among applications, which is especially important in

complex environments with lot of servers and in those environments with machines with similar names, such as `tivapp1` and `tivapp2`.

Also, the Single Sign-On capability, which enables an operator to switch between applications without needing to log in with the password every time, is a great benefit.

Automated application management scenario

The automated application management scenario that we discuss in this chapter is based on the existing operational drill-down scenario that we discuss in Chapter 8, “Operational drill down” on page 341. We extend the scenario to add the application management scenario that deals with the application as a unit, instead of as a group of individual resources.

We discuss this scenario in the following sections:

- ▶ 9.1, “Scenario overview” on page 366
- ▶ 9.2, “Scenario setup” on page 367
- ▶ 9.3, “Sample walk-through” on page 387
- ▶ 9.4, “Summary of benefits” on page 402

9.1 Scenario overview

This scenario describes how an operator can view information a particular server from various Tivoli operational management tools and navigate between them. The operational drill-down flow that we use works from Tivoli Enterprise Portal operational workspaces and built-in Web browser launches to other management tools.

Figure 9-1 depicts the scenario flow.

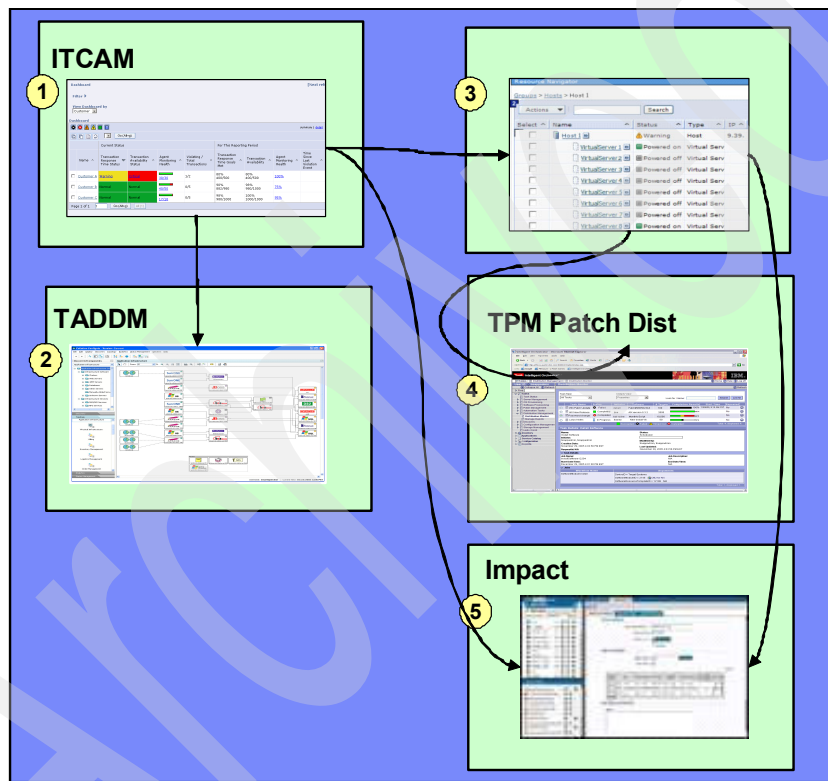


Figure 9-1 Scenario flow

This scenario uses these products:

- ▶ Tivoli Monitoring V6.2.1
- ▶ Tivoli Composite Application Manager for Transactions V7.1
- ▶ Tivoli Application and Dependency Discovery Manager
- ▶ IBM Tivoli Provisioning Manager
- ▶ IBM Tivoli Netcool/OMNIBus V7.2.1

9.2 Scenario setup

This scenario requires the following setup:

- ▶ Single sign-on between IBM Tivoli Monitoring, IBM Tivoli Application Dependency Discovery Manager, and IBM Tivoli Provisioning Manager (see Chapter 4, “Security integration” on page 45)
- ▶ The launch in context function defined for IBM Tivoli Monitoring to IBM Tivoli Provisioning Manager and IBM Tivoli Provisioning Manager to IBM Tivoli Application Dependency Discovery Manager (see Chapter 6, “Navigation integration” on page 231)
- ▶ Common Event Console setup as discussed in 8.2, “Scenario setup” on page 343

We discuss these specific integration points:

- ▶ 9.2.1, “Setting the managed system name” on page 367
- ▶ 9.2.2, “Setting up the Common Event Console” on page 369
- ▶ 9.2.3, “Navigation from Tivoli Enterprise Portal” on page 377
- ▶ 9.2.4, “Tivoli Enterprise Portal to IBM Tivoli Application Dependency Discovery Manager integration specifics” on page 381
- ▶ 9.2.5, “IBM Tivoli Provisioning Manager integration specifics” on page 384

9.2.1 Setting the managed system name

To ease the integration between Tivoli products, we recommend that you update the managed system name for all IBM Tivoli Monitoring and IBM Tivoli Composite Application Manager agents. The managed system name is required when launching IBM Tivoli Monitoring in context from another application, and it enables you to filter events from the same host in the event console (IBM Tivoli Netcool/OMNIbus or Common Event Console), even though the events originated from separate Tivoli products.

Updating the Operating System agent managed system name

The managed system name is the name of the IBM Tivoli Monitoring installed agent or component, as it appears in the command `tacmd listSystems` output or on the Tivoli Enterprise Portal view.

After installation, the Operating System agent name is set to the default format: *Instance:Hostname:NodeType*, which is not useful to correlate information between agents on the same system.

To update the managed system name, you can use these environment variables:

- ▶ CTIRA_HOSTNAME defines the host name portion of the managed system name as it appears in the **tacmd listSystems** command output and Managed System Status view on Tivoli Enterprise Portal.
- ▶ CTIRA_SYSTEM_NAME defines the host name as it appears in the Navigator view of Tivoli Enterprise Portal.
- ▶ CTIRA_SUBSYSTEM_ID defines the instance portion of the managed system name.

Important: CTIRA_HOSTNAME and CTIRA_SYSTEM_NAME must be the same for all agents residing on the same host.

To update the managed system name, perform the following steps:

1. Stop the agent either by using the **itmcmd** command in UNIX/Linux or by stopping the Services in Windows.
2. Modify the environment variables:
 - In UNIX/Linux, edit the `/opt/IBM/ITM/config/<pc>.ini` file to set the variables to the following values:
 - CTIRA_HOSTNAME=<full hostname>
 - CTIRA_SYSTEM_NAME=<full hostname>
 - CTIRA_SUBSYSTEM_ID= (a blank)
 - In Windows, using Manage Tivoli Enterprise Monitoring Services, right-click the agent, and select **Advanced** → **Edit Variables**. Set the variables to the following values:
 - CTIRA_HOSTNAME=<full hostname> .TYPE=REG_EXPAND_SZ
 - CTIRA_SYSTEM_NAME=<full hostname>
 - CTIRA_SUBSYSTEM_ID= (a blank)

CTIRA_SUBSYSTEM_ID has no value (blank), which removes the default instance Primary from the managed system name.
3. Restart the agent either by using the **itmcmd** command in UNIX/Linux or stopping the Services in Windows.

9.2.2 Setting up the Common Event Console

The Common Event Console allows notifications from various Tivoli products, such as IBM Tivoli Composite Application Manager and IBM Tivoli Application Dependency Discovery Manager, to be collected in one place and displayed from a given host perspective. The events are all stored in IBM Tivoli Netcool/OMNIBus.

Follow these steps to set up the Common Event Console:

1. Because IBM Tivoli Composite Application Manager technically is a part of IBM Tivoli Monitoring, you need to configure IBM Tivoli Monitoring Hub Tivoli Enterprise Monitoring Server to send events to IBM Tivoli Netcool/OMNIBus, as described in 5.3.1, “IBM Tivoli Netcool/OMNIBus and IBM Tivoli Monitoring” on page 177.
2. For the agents, define the situations for each IBM Tivoli Composite Application Manager agent, so that problems are detected and reported.
3. We must set up IBM Tivoli Application Dependency Discovery Manager to send events to IBM Tivoli Netcool/OMNIBus. We use a solution from the Open Process Automation Library (OPAL) that was already mentioned in this book in 7.5, “IBM Tivoli Application Dependency Discovery Manager” on page 328. This solution also allows Event Integration Facility events from IBM Tivoli Application Dependency Discovery Manager to be sent to IBM Tivoli Netcool/OMNIBus. Refer to this Web site:

<http://www.ibm.com/software/tivoli/opal/details?catalog.label=1TW10C1Q>

Apart from setting host names, ports, users, and passwords, which is described in the documentation, we create variables settings in the `omnibus.eif.properties` file (Example 9-1).

Example 9-1 The omnibus.eif.properties file

```
TADDEvent_Slot_msg='$TADDM_ATTRIBUTE_NAME $TADDM_CHANGE_TYPE'
TADDEvent_Slot_severity=Warning
TADDEvent_Slot_origin=$TADDM_HOST
TADDEvent_Slot_hostname=$TADDM_OBJECT_NAME
TADDEvent_Slot_sub_source=$TADDM_GUID
```

4. The setting in Example 9-1 differs from the default configuration. It is important to set these variables (especially the host name variable), because this setup uses them.
5. We configure this module to send events to the same IBM Tivoli Netcool/OMNIBus EIF Probe that IBM Tivoli Monitoring uses. By default, the

Event Integration Facility probe sets the TECHostname field in IBM Tivoli Netcool/OMNIBus to the value of hostname in the arriving event:

@TECHostname=\$hostname

- When set up correctly, after IBM Tivoli Application Dependency Discovery Manager discovers the changes on the managed system, you get an event in the Active Event List. In the Details tab, you can see that it came from IBM Tivoli Application Dependency Discovery Manager, as shown in Figure 9-2. Notice the values of the slots that were set up in Example 9-1 on page 369.

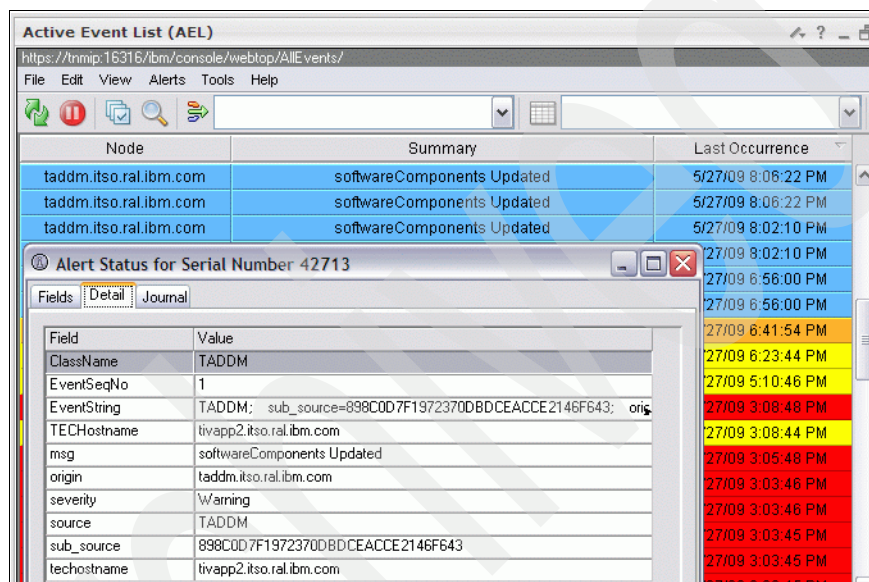


Figure 9-2 Change events from IBM Tivoli Application Dependency Discovery Manager in the Active Events List

After you complete these steps, you can proceed to Tivoli Enterprise Portal and configure the Common Event Console. Go to the Manage Tivoli Enterprise Monitoring Services GUI or use the `itmcmd manage` command on UNIX/Linux. Right-click **Tivoli Enterprise Portal Server**, and choose **Configure**. The Common Event Console Configuration is displayed. Change the name of the first extra column (Name of extra column 1) to Hostname, as shown in Figure 9-3 on page 371.

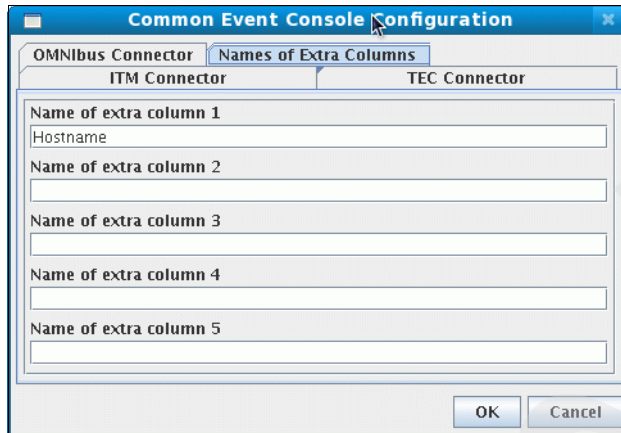


Figure 9-3 Setting the name of the extra column

Then, disable the IBM Tivoli Monitoring Connector (ITM1), as shown in Figure 9-4.

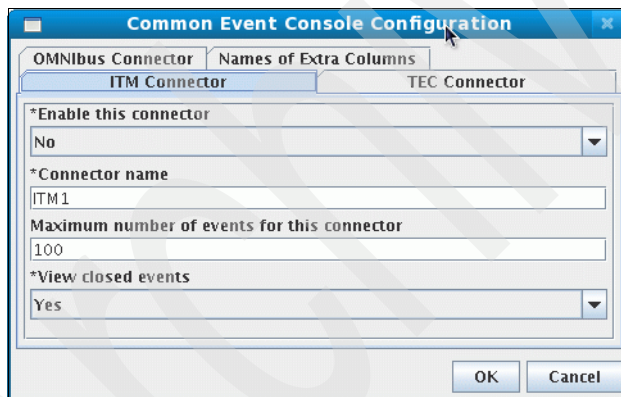


Figure 9-4 ITM1 connector disabled

We do not want IBM Tivoli Monitoring events in the Common Event Console, because we take all the events from IBM Tivoli Netcool/OMNIBus and also those events coming from IBM Tivoli Monitoring originally. We want to correlate events based on the hostname field, and the original IBM Tivoli Monitoring events do not have this field in the Common Event Console. For IBM Tivoli Monitoring events coming from IBM Tivoli Netcool/OMNIBus, the field is set by the Event Integration Facility probe.

Now, we need to modify the connector to include events that originally came from IBM Tivoli Monitoring to IBM Tivoli Netcool/OMNIBus. By default, they are

excluded (to avoid duplicates) by setting SQL WHERE clause that restricts events for common event console: ITMStatus = ''.

You need to clear this setting; leave it blank. Also, in this tab for the first extra column, select the field type **alerts.status** and field name **TECHostname**. Figure 9-5 shows the correct settings.

The screenshot shows the 'Common Event Console Configuration' dialog box. The 'Names of Extra Columns' tab is selected, and the 'ITM Connector' sub-tab is active. The configuration fields are as follows:

- *IBM Tivoli Netcool/OMNibus connector: Delete button
- *Connector name: itso_omnibus
- *Maximum number of events for this connector: 100
- *Computer name of event system: tbsm.itso.ral.ibm.com
- *Port number of event system: 4100
- *User name for accessing event system: itm
- Password: *****
- Confirm Password: *****
- SQL WHERE clause that restricts events for common event console: (blank)
- *View cleared events: Yes (dropdown menu)
- Time interval (in minutes) for checking for new or changed events: 1
- Time interval (in seconds) between reconnection attempts: 20
- Maximum number of consecutive reconnection attempts: 30
- Field type for extra column 1: alerts.status (dropdown menu)
- Field name for extra column 1: TECHostname
- Field type for extra column 2: (blank)

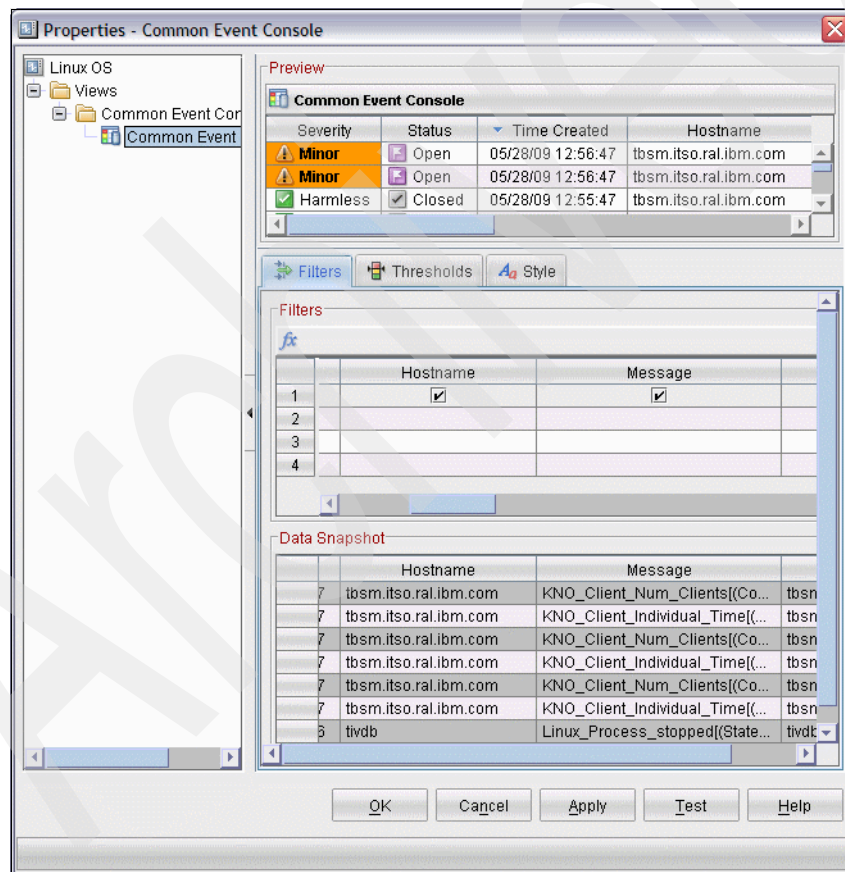
Buttons at the bottom: OK, Cancel.

Figure 9-5 Configuration of IBM Tivoli Netcool/OMNibus Connector

Now, save the settings. Tivoli Enterprise Portal Server restarts and applies the changes. Now, create a workspace with the Common Event Console in Tivoli Enterprise Portal. Select an element in Navigator (for Example, Linux OS), click the Common Event Console icon on the top menu bar (Figure 9-6 on page 373),



When the Common Event Console window is displayed, go to its Properties (icon with / and downward arrow in the top right corner) window to make sure that the Hostname and Message columns are selected to be displayed (by selecting the check box, as shown in Figure 9-7). You can then remove all other graphs and make the Common Event Console the only graph in the workspace. Save workspace so that you can access it in the future.



Your workspace looks similar to Figure 9-8 on page 374.

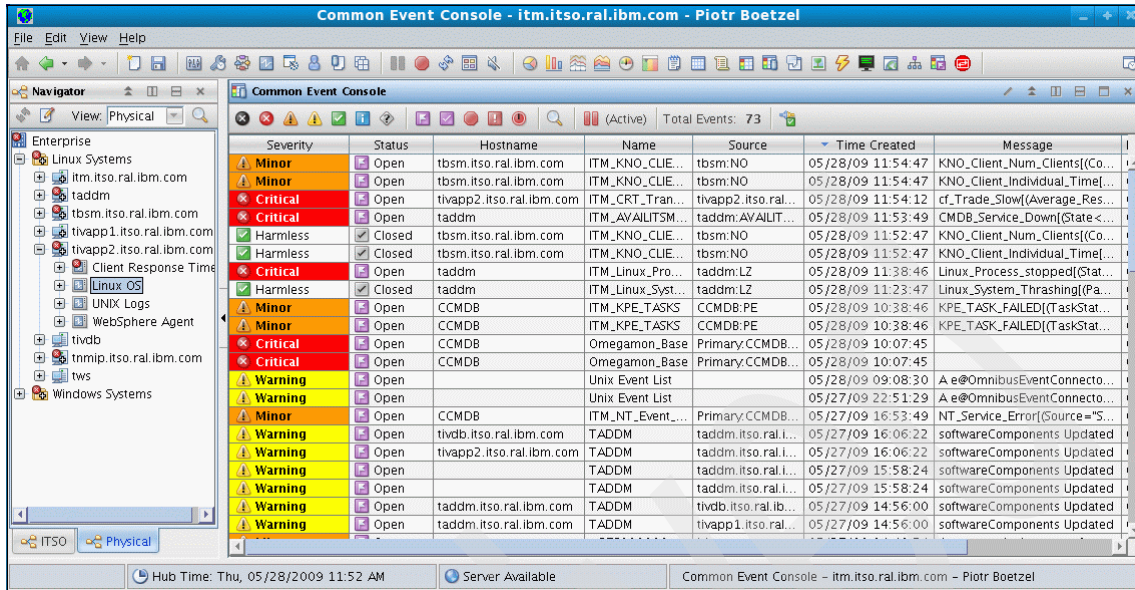


Figure 9-8 Workspace with the Common Event Console and the Navigator only

Suppose that you notice, in the Common Event Console, an event from IBM Tivoli Composite Application Manager Client Response Time that states that the Trade application is slow, as shown in Figure 9-9 on page 375.

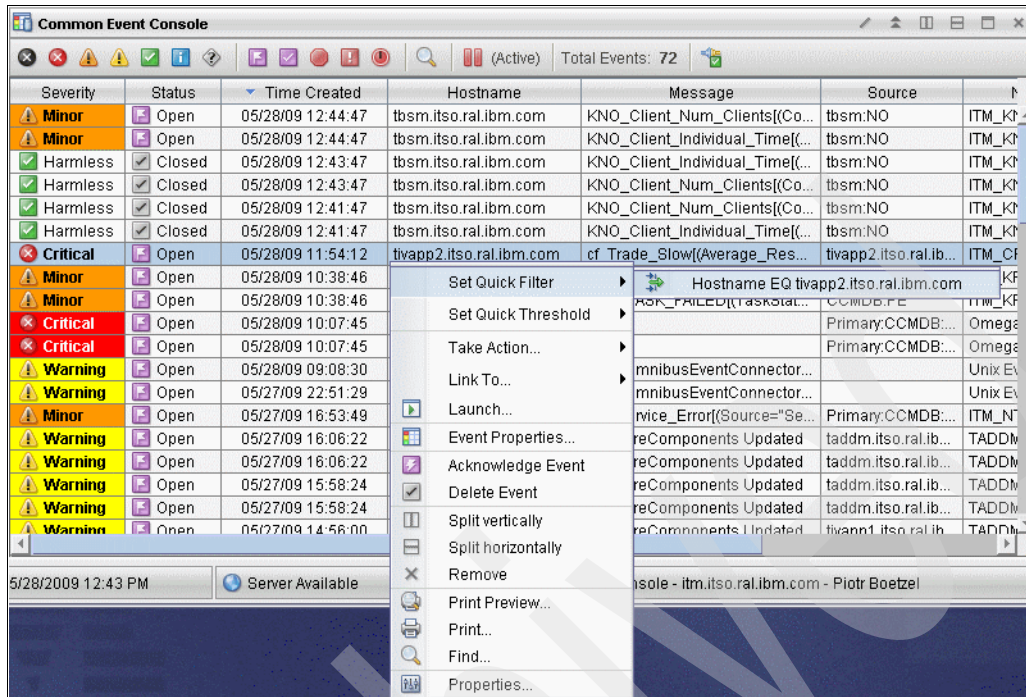


Figure 9-10 Enable Quick Filter

You can see all of the events that relate to that host, as shown in Figure 9-11.

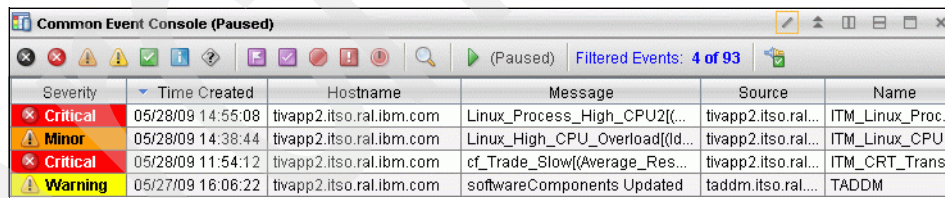


Figure 9-11 Filtered view in the Common Event Console

In this view, you can see that IBM Tivoli Application Dependency Discovery Manager discovered a software change (softwareComponents updated) on this system the previous day, which might be a place to look for the source of the problem.

Notice that, in this view, you also have information that the last event came from IBM Tivoli Application Dependency Discovery Manager (Name column). The first and second events came from the IBM Tivoli Monitoring Linux agent and the fourth event came from IBM Tivoli Composite Application Manager Client

Response Time. So, in this single view, you see information from three separate Tivoli Products:

- ▶ IBM Tivoli Monitoring
- ▶ IBM Tivoli Composite Application Manager (Client Response Time)
- ▶ IBM Tivoli Application Dependency Discovery Manager

Those events are actually stored in IBM Tivoli Netcool/OMNIbus.

9.2.3 Navigation from Tivoli Enterprise Portal

To begin this scenario, set up a view in Tivoli Enterprise Portal, which displays configuration information for a Linux server. It is beneficial to create the workspace to be as generic as possible so that it can be created one time and distributed to all monitored servers:

1. First, identify a suitable place for the configuration information. In this example, the Linux OS navigator item was chosen, because a Linux OS navigator item exists on every server. Navigate to the desired node in the Linux OS agent and create a Web browser workspace, as seen in Figure 9-12 on page 378.

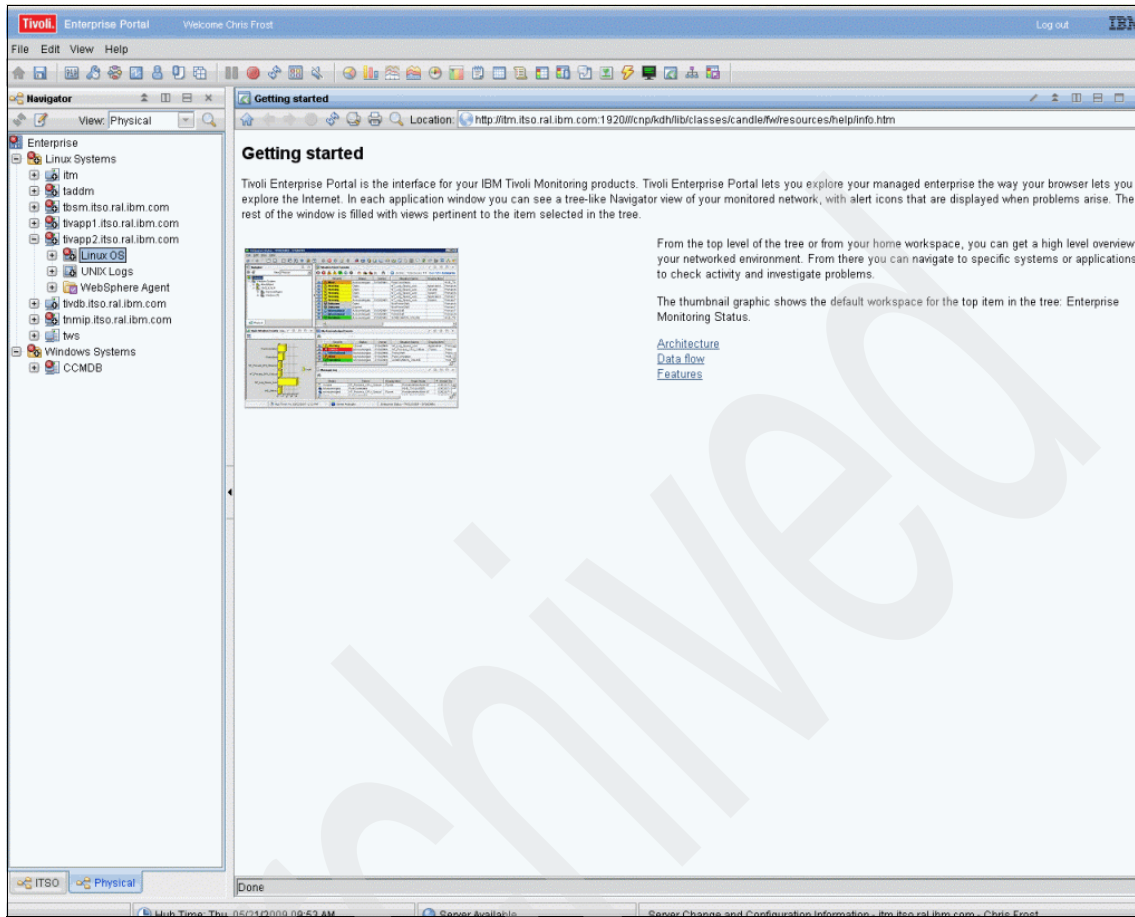


Figure 9-12 Tivoli Enterprise Portal browser window

2. In this step, you enter the URL of the Tivoli application server (that is, IBM Tivoli Application Dependency Discovery Manager, IBM Tivoli Provisioning Manager, IBM Tivoli Network Manager for IP, and so forth). Edit the properties of the browser window, and enter the URL in the space provided. Select **Use Provided Location**. The properties window now looks similar to Figure 9-13 on page 379. Click **OK**.

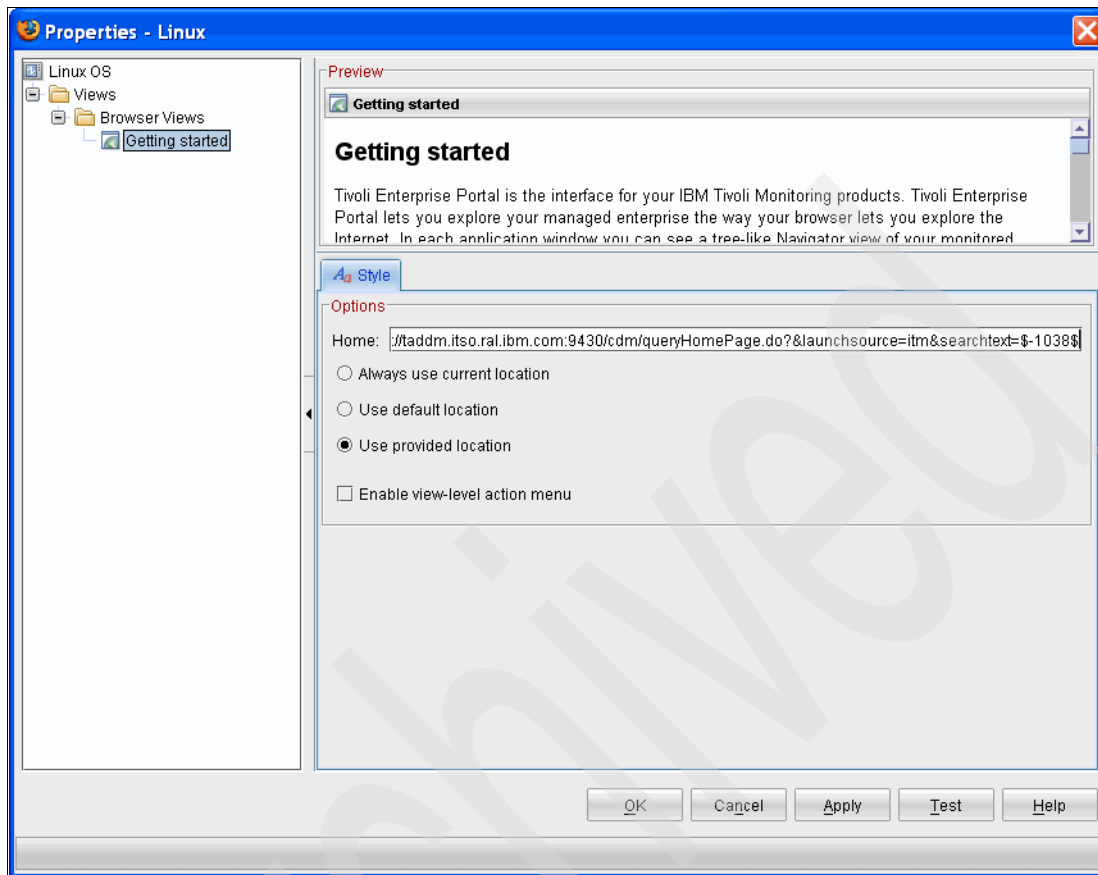


Figure 9-13 Workspace Properties window

3. Save the workspace. Click **File** → **Save Workspace As**. Choose a suitable name for the workspace. Figure 9-14 on page 380 shows an example.

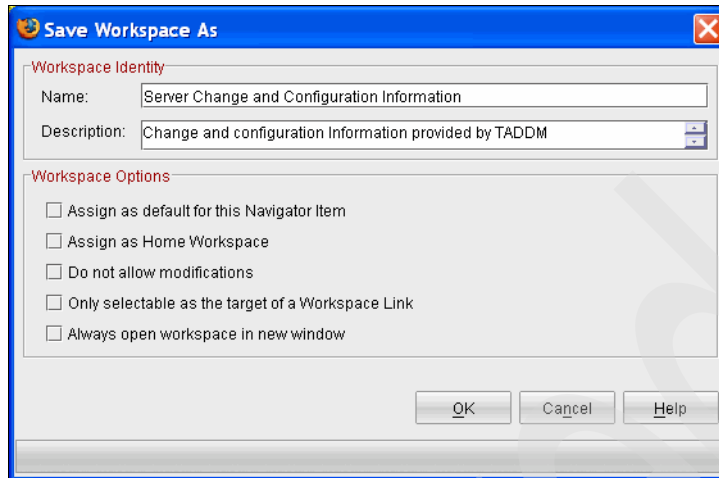


Figure 9-14 Save Workspace As window

4. Navigate to another page. Return to the newly created workspace, as shown in Figure 9-15 on page 381.

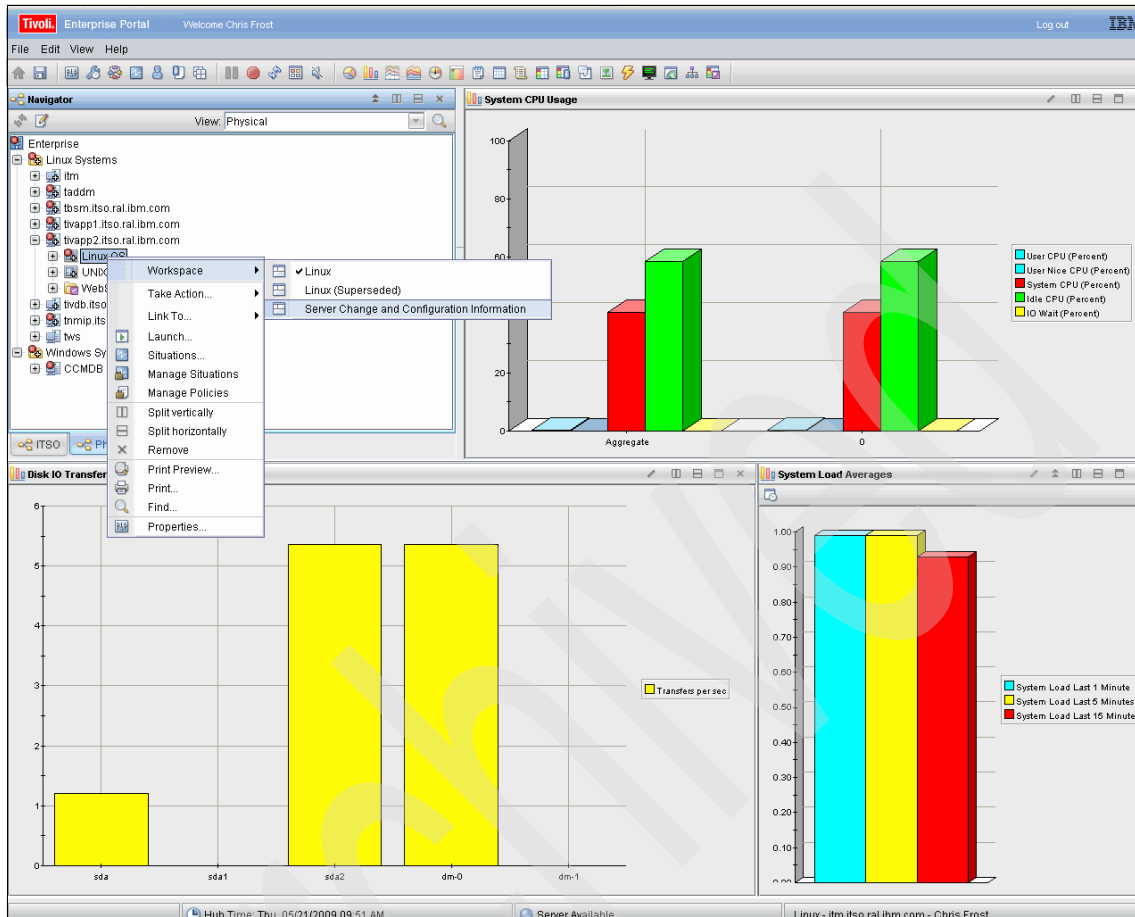


Figure 9-15 Tivoli Enterprise Portal Workspace menu

9.2.4 Tivoli Enterprise Portal to IBM Tivoli Application Dependency Discovery Manager integration specifics

In this section, you see Tivoli Enterprise Portal to IBM Tivoli Application Dependency Discovery Manager integration showing an IBM Tivoli Application Dependency Discovery Manager window within Tivoli Enterprise Portal. The automatic sign-on to IBM Tivoli Application Dependency Discovery Manager occurs using Tivoli Enterprise Portal user credentials, as described in 4.7, “IBM Tivoli Monitoring” on page 120. Note that the host name in IBM Tivoli Application Dependency Discovery Manager corresponds to the system in Tivoli Enterprise Portal from which we launched.

Edit the properties of the Tivoli Enterprise Portal browser window by entering the following URL:

`http://taddm.itso.ral.ibm.com:9430/cdm/queryHomePage.do?&launchsource=itm&searchtext=$-1038$`

This URL launches an IBM Tivoli Application Dependency Discovery Manager page in Tivoli Enterprise Portal. The `$-1038$` variable resolves to the IP address of the current navigator item in Tivoli Enterprise Portal. This IP address is then searched in IBM Tivoli Application Dependency Discovery Manager, revealing a page with the correct configuration item, as shown in Figure 9-16.

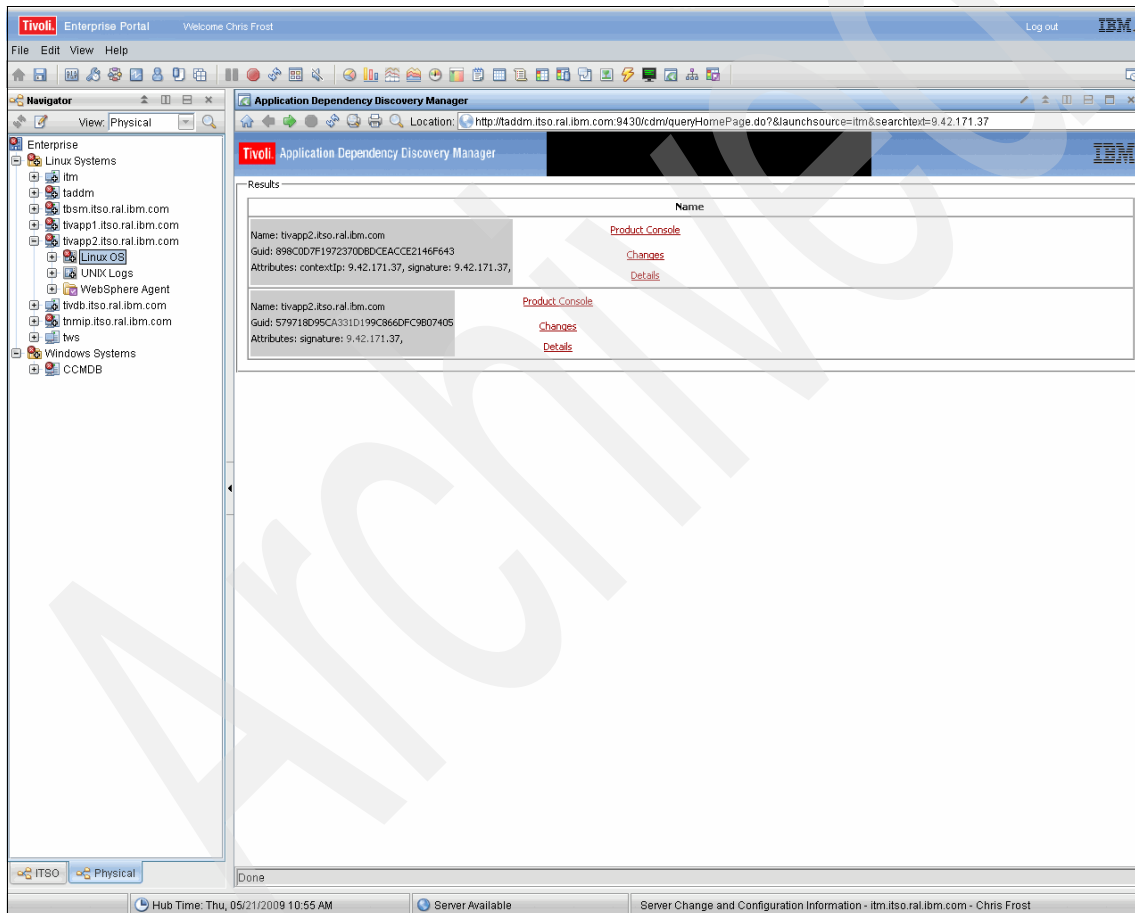


Figure 9-16 IBM Tivoli Application Dependency Discovery Manager window in Tivoli Enterprise Portal

From this window, the operator can now launch to three places:

- IBM Tivoli Application Dependency Discovery Manager Console opens a separate Java Web-start console for IBM Tivoli Application Dependency Discovery Manager, allowing the operator to view dependencies, configurations, and so forth.
- The changes link navigates to an IBM Tivoli Application Dependency Discovery Manager page of changes for the server, while remaining within Tivoli Enterprise Portal. See Figure 9-17 for an example.
- Details also opens an IBM Tivoli Application Dependency Discovery Manager page in Tivoli Enterprise Portal that shows the configuration details for the server.

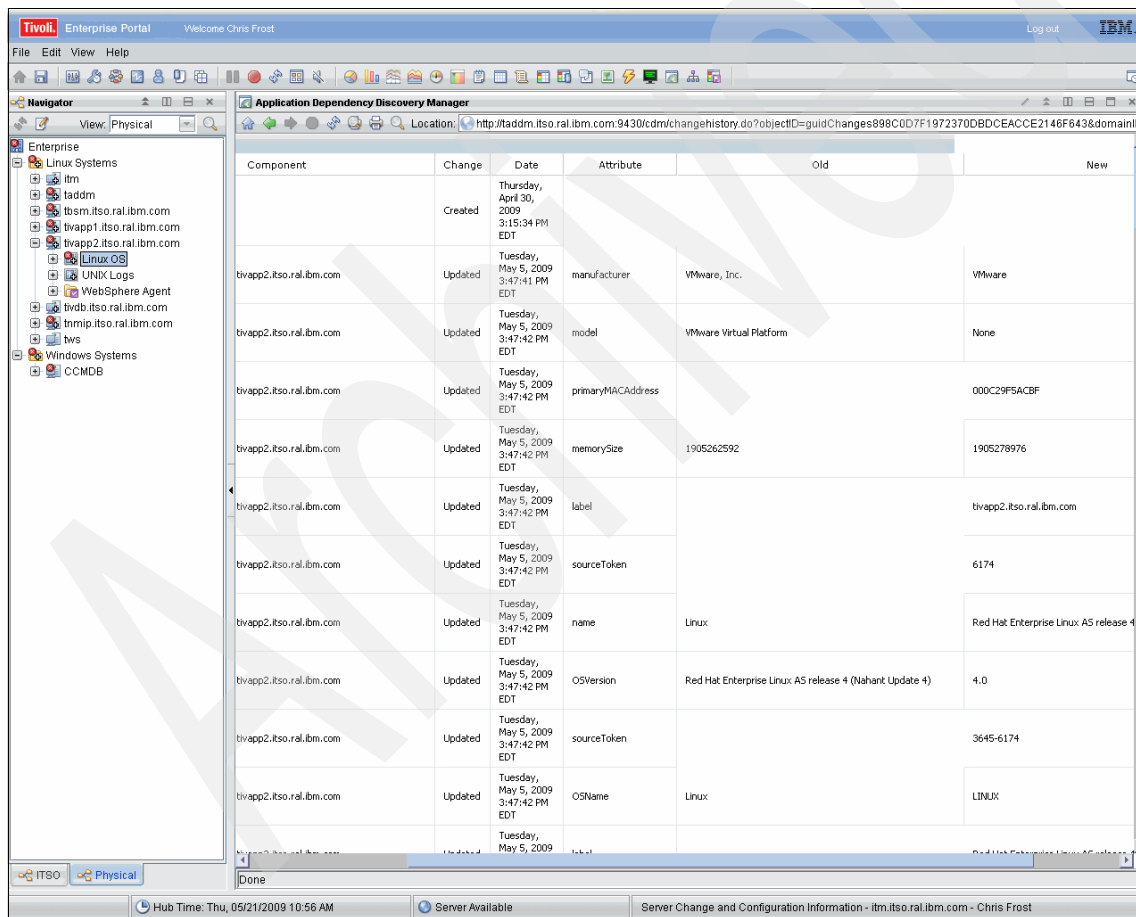


Figure 9-17 Changes viewed in Tivoli Enterprise Portal

9.2.5 IBM Tivoli Provisioning Manager integration specifics

In this section, we use Tivoli Enterprise Portal as a starting point and create a link to IBM Tivoli Provisioning Manager. To launch IBM Tivoli Provisioning Manager from Tivoli Enterprise Portal, you need to create a workspace in the same way as when launching IBM Tivoli Application Dependency Discovery Manager from Tivoli Enterprise Portal as described in 8.2, “Scenario setup” on page 343.

The only difference is that you use this link in terms of IBM Tivoli Provisioning Manager:

```
https://ccmdb.itso.ral.ibm.com:9045/maximo/ui/?event=loadapp&value=tpdc  
mfind&additionalevent=useqbe&additionaleventvalue=Name=$-1071$
```

You replace `ccmdb.itso.ral.ibm.com:9045` with your IBM Tivoli Provisioning Manager server host name and port. You can read more about the methods of launching IBM Tivoli Provisioning Manager in context in 6.2.2, “Tivoli Process Automation Engine-based products” on page 235.

Now, you see a IBM Tivoli Provisioning Manager window within Tivoli Enterprise Portal, as shown in Figure 9-2 on page 370. Automatic sign-on to IBM Tivoli Provisioning Manager occurs by using Tivoli Enterprise Portal user credentials, as described in 4.7, “IBM Tivoli Monitoring” on page 120. The context that is passed from Tivoli Enterprise Portal to IBM Tivoli Provisioning Manager is the managed system host name (variable `$-1071$`).

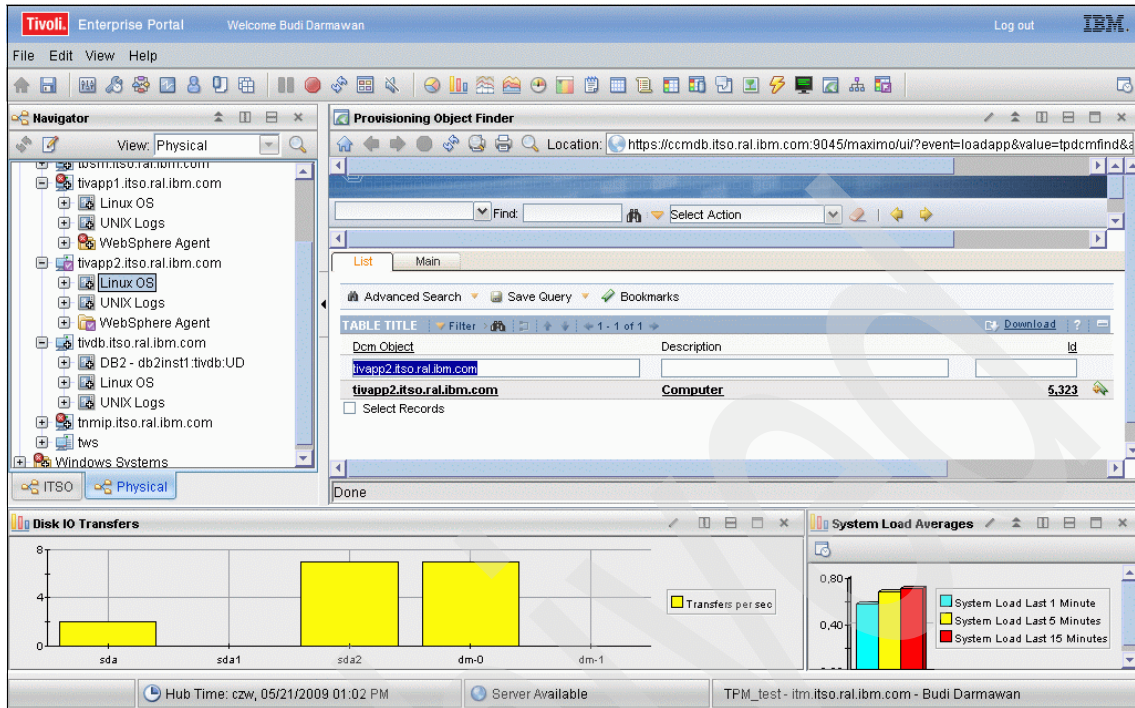


Figure 9-18 IBM Tivoli Provisioning Manager window inside Tivoli Enterprise Portal Workspace

You can then click a host name in IBM Tivoli Provisioning Manager window to see general information about that computer, as shown in Figure 9-4 on page 371.

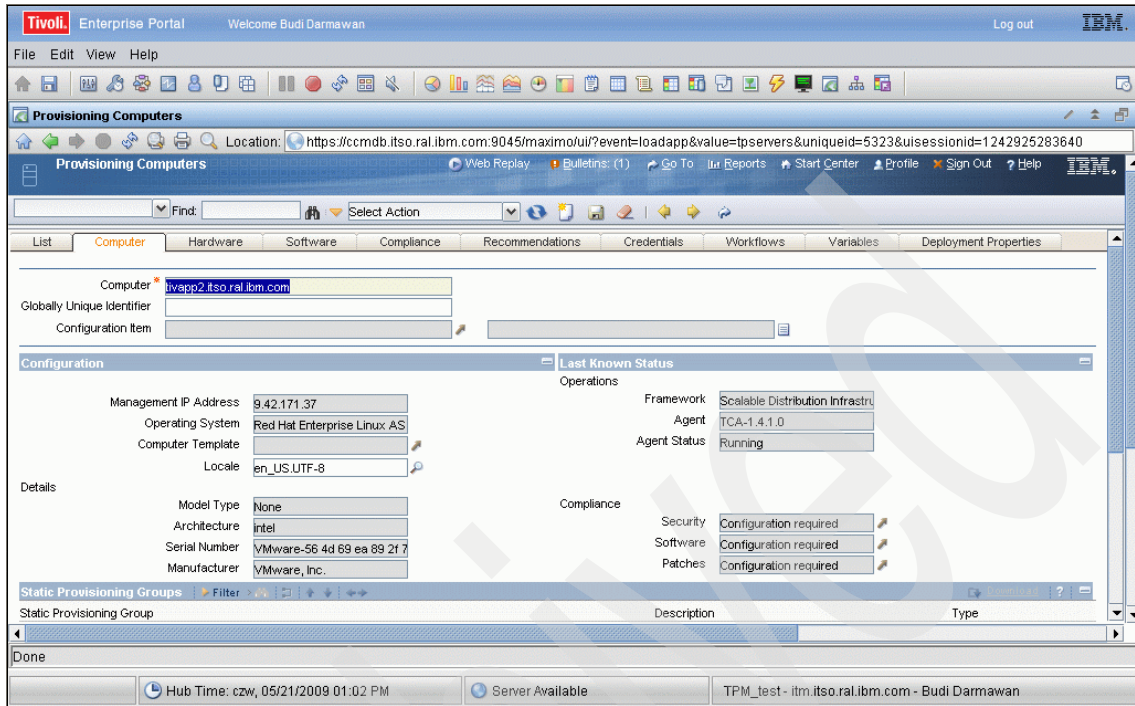


Figure 9-19 Computer information

On the Software tab, you can view the installed software on this computer, as shown in Figure 9-20 on page 387.

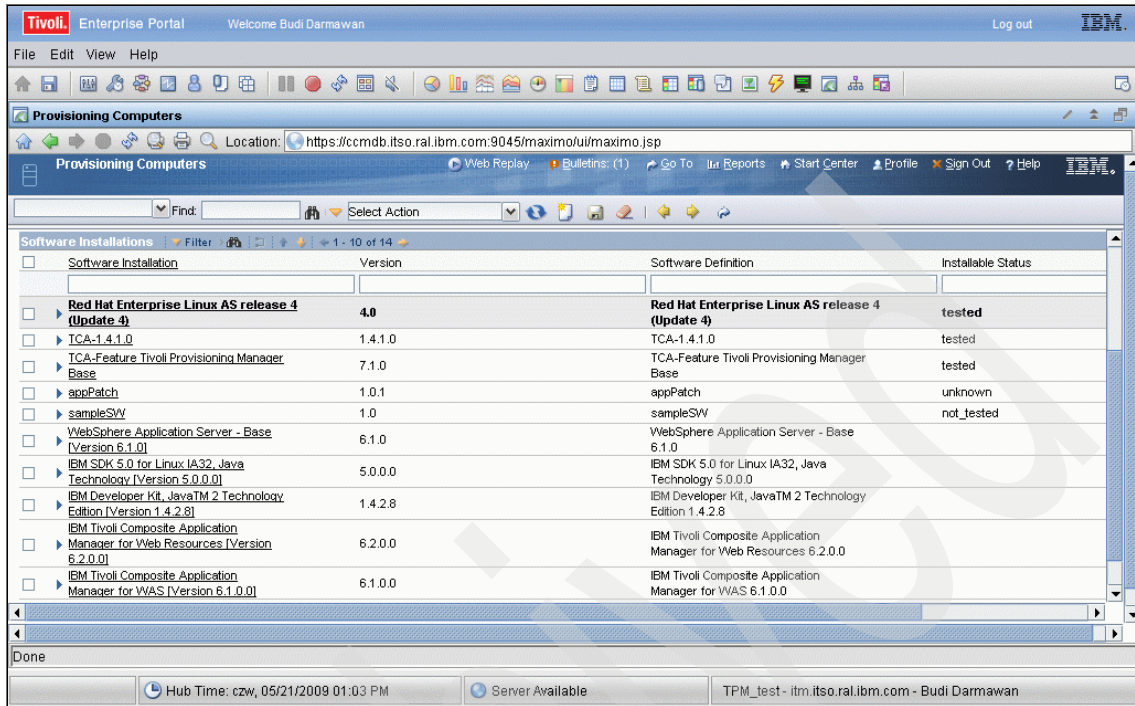


Figure 9-20 Software tab

9.3 Sample walk-through

Figure 9-21 on page 388 depicts the walk-through.

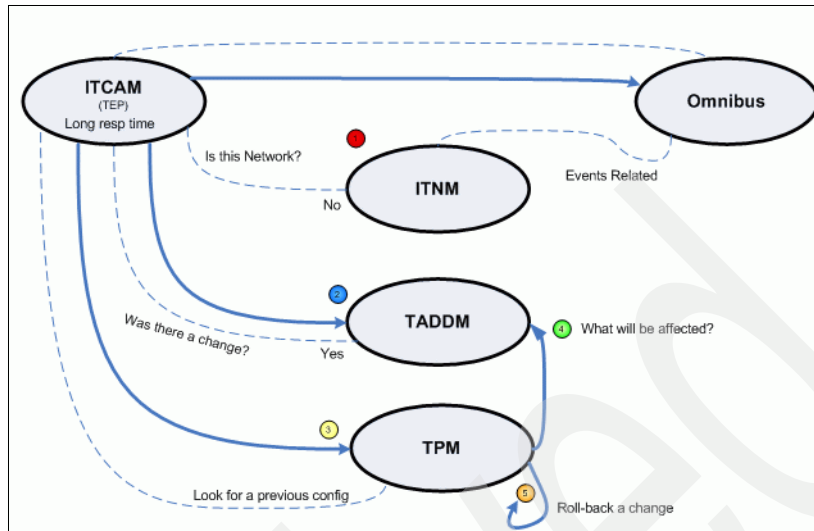


Figure 9-21 Application problem walk-through

The scenario in Figure 9-21 follows these steps:

1. IBM Tivoli Composite Application Manager detects a performance problem with the Trade application, so the operator checks for related events from IBM Tivoli Netcool/OMNIBus through the Common Event Console in Tivoli Enterprise Portal. No network events were received from IBM Tivoli Network Manager for IP.
2. An IBM Tivoli Application Dependency Discovery Manager event depicting a recent change to the Trade application server is detected and deemed to be the cause of the incident. So, the operator dynamically launches an IBM Tivoli Application Dependency Discovery Manager window to browse the change history without leaving Tivoli Enterprise Portal.
3. The operator then opens Tivoli Provisioning Manager within Tivoli Enterprise Portal to view the available software packages.
4. The operator then launches IBM Tivoli Provisioning Manager in context. From IBM Tivoli Provisioning Manager, the operator also checks the IBM Tivoli Application Dependency Discovery Manager dependencies to assess the impact of the change.
5. The operator rolls back the change in IBM Tivoli Provisioning Manager by deploying the master version of a configuration file.

The scenario begins with an operator observing the performance of the Trade application by utilizing the Client Response Time workspaces in Tivoli Enterprise Portal:

1. The operator browses to the Client Response Time workspace to view the current status of the transactions in the Trade application. Initially, all the transactions have an acceptable response time, as shown in Figure 9-22.

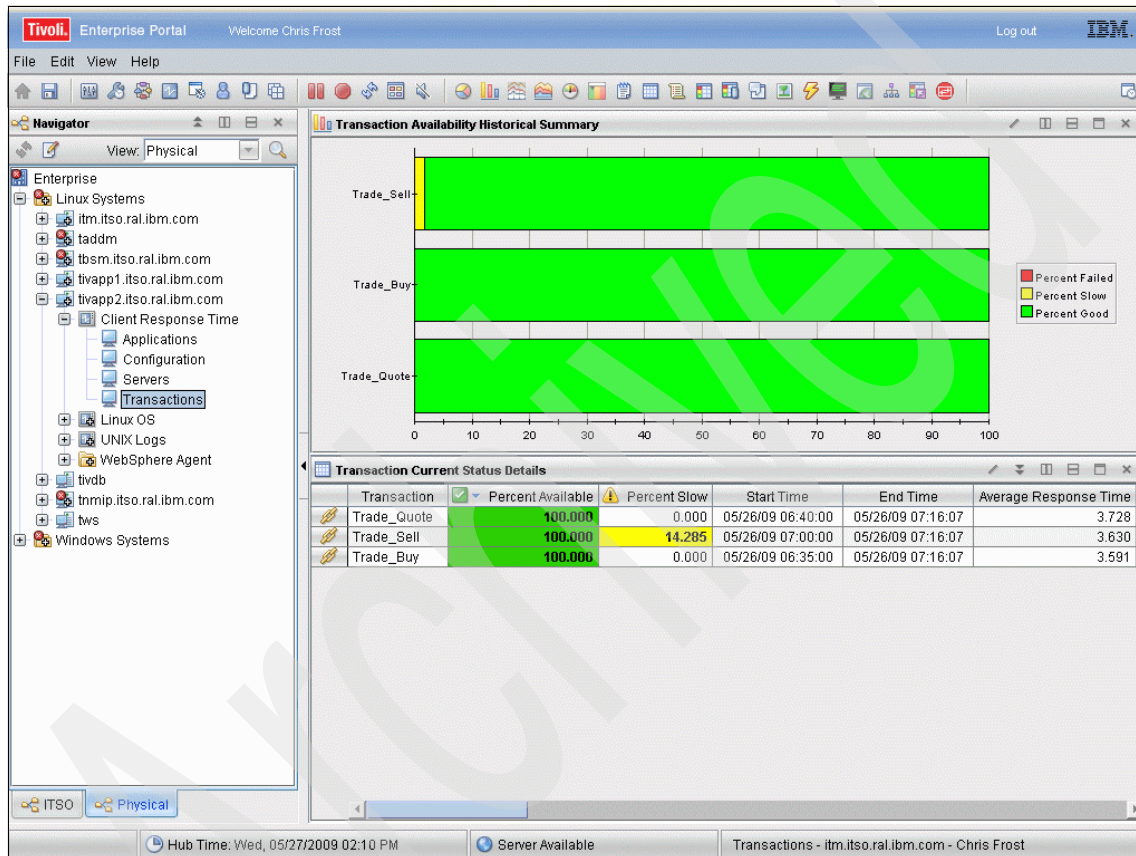


Figure 9-22 Client Response Time window showing good transaction availability

2. Later, the operator receives an alert for a slow transaction and, again, views the same workspace. This time, a high percentage of slow response times shows for each transaction, as shown in Figure 9-23 on page 390.

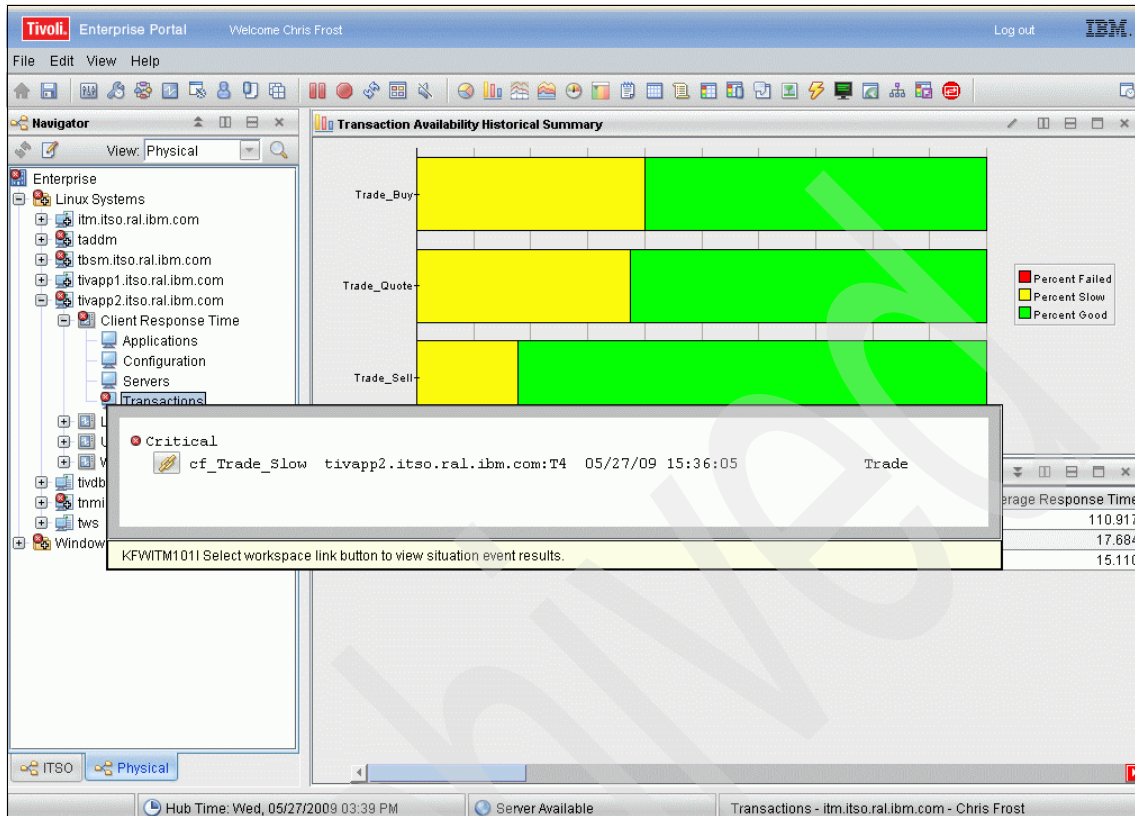


Figure 9-23 Client Response Time workspace showing slow transactions

3. In this case, the operator first looks for related events. A workspace has been set up with the Common Event Console as described in 8.2, "Scenario setup" on page 343. The operator navigates to this workspace (Figure 9-26 on page 393) and views the related events. The Common Event Console is showing events from IBM Tivoli Netcool/OMNIBus, which might include events from other monitoring systems, such as IBM Tivoli Application Dependency Discovery Manager or IBM Tivoli Network Manager for IP, as well as IBM Tivoli Monitoring events.
4. Initially, in the Common Event Console, the operator sees a large list of events, as shown in Figure 9-24 on page 391. In order to see the events that are related directly to the affected server, `tivapp2.itso.ral.ibm.com`, the operator first locates an event with the correct host name, `tivapp2.itso.ral.ibm.com`.

Tip: We created the Hostname field specifically for this purpose, as documented in 9.2, “Scenario setup” on page 367.

Severity	Status	Time Created	Hostname	Message	Name	Source
Minor	Open	05/28/09 11:4...	tbsm.itso.ral.ibm.com	KNO_Client_Individual_Time[...	ITM_KNO_CLIE...	tbsm:NO
Minor	Open	05/28/09 11:4...	tbsm.itso.ral.ibm.com	KNO_Client_Num_Clients(Co...	ITM_KNO_CLIE...	tbsm:NO
Harmless	Closed	05/28/09 11:3...	tbsm.itso.ral.ibm.com	KNO_Client_Num_Clients(Co...	ITM_KNO_CLIE...	tbsm:NO
Harmless	Closed	05/28/09 11:3...	tbsm.itso.ral.ibm.com	KNO_Client_Individual_Time[...	ITM_KNO_CLIE...	tbsm:NO
Harmless	Closed	05/28/09 11:3...	tbsm.itso.ral.ibm.com	KNO_Client_Num_Clients(Co...	ITM_KNO_CLIE...	tbsm:NO
Harmless	Closed	05/28/09 11:3...	tbsm.itso.ral.ibm.com	KNO_Client_Individual_Time[...	ITM_KNO_CLIE...	tbsm:NO
Critical	Open	05/28/09 11:3...	taddm	Linux_Process_stopped(Stat...	ITM_Linux_Pro...	taddm:LZ
Harmless	Closed	05/28/09 11:3...	taddm	CMDB_Service_Down(Stat<...	ITM_AVAILTSM...	taddm:AVAILT...
Critical	Open	05/28/09 11:2...	taddm	Linux_System_Thrashing(Pa...	ITM_Linux_Syst...	taddm:LZ
Critical	Open	05/28/09 11:1...	tivapp2.itso.ral.ibm.com	cf_Trade_Slow(Average_Res...	ITM_CRT_Trans...	tivapp2.itso.ral...
Minor	Open	05/28/09 10:3...	CCMDB	KPE_TASK_FAILED(TradeSta...	ITM_KPE_TASKS...	CCMDB:RE...
Minor	Open	05/28/09 10:3...	CCMDB	KPE_TASK_FAILED(TradeSta...	ITM_KPE_TASKS...	CCMDB:RE...
Critical	Open	05/28/09 10:0...	CCMDB	cf_Trade_Slow(Average_Response_Time>=8.000) ON tivapp2.itso.ral.ibm.com:T4		DB...
Critical	Open	05/28/09 10:0...	CCMDB	ON Trade (Average_Response_Time=10.224)		DB...
Warning	Open	05/28/09 09:0...		A e@OmnibusEvent		
Warning	Open	05/27/09 22:5...		A e@OmnibusEventConnecto...	Unix Event List	
Minor	Open	05/27/09 16:5...	CCMDB	NT_Service_Error(Source="S...	ITM_NT_Event...	Primary:CCMDB...
Warning	Open	05/27/09 16:0...	tivapp2.itso.ral.ibm.com	softwareComponents Updated	TADDMM	taddm.itso.ral.i...
Warning	Open	05/27/09 16:0...	tivdb.itso.ral.ibm.com	softwareComponents Updated	TADDMM	taddm.itso.ral.i...
Warning	Open	05/27/09 15:5...		softwareComponents Updated	TADDMM	taddm.itso.ral.i...
Warning	Open	05/27/09 15:5...		softwareComponents Updated	TADDMM	taddm.itso.ral.i...
Warning	Open	05/27/09 14:5...	taddm.itso.ral.ibm.com	softwareComponents Updated	TADDMM	tivdb.itso.ral.ib...
Warning	Open	05/27/09 14:5...	taddm.itso.ral.ibm.com	softwareComponents Updated	TADDMM	tivapp1.itso.ral...
Minor	Open	05/27/09 14:4...		Attempt to login as root fro...	e@7F000001...	blue
Minor	Open	05/27/09 14:2...	tivapp2	Linux_Process_High_Cpu(Bu...	ITM_Linux_Pro...	tivapp2:LZ
Minor	Open	05/27/09 14:2...	tivapp2	Linux_High_CPU_Overload(l...	ITM_Linux_CPU...	tivapp2:LZ
Minor	Open	05/27/09 13:1...	tbsm.itso.ral.ibm.com	KNO_Client_Individual_Time[...	ITM_KNO_CLIE...	tbsm:NO

Figure 9-24 Common Event Console window in IBM Tivoli Monitoring

- Next, the operator can right-click the Hostname field and select **Set Quick Filter** → **Hostname EQ tivapp2.itso.ral.ibm.com** (Figure 9-25 on page 392). This filter displays only events that match the criteria and allows the operator to quickly identify other related events from products, such as IBM Tivoli Application Dependency Discovery Manager and IBM Tivoli Network Manager for IP (Figure 9-26 on page 393).

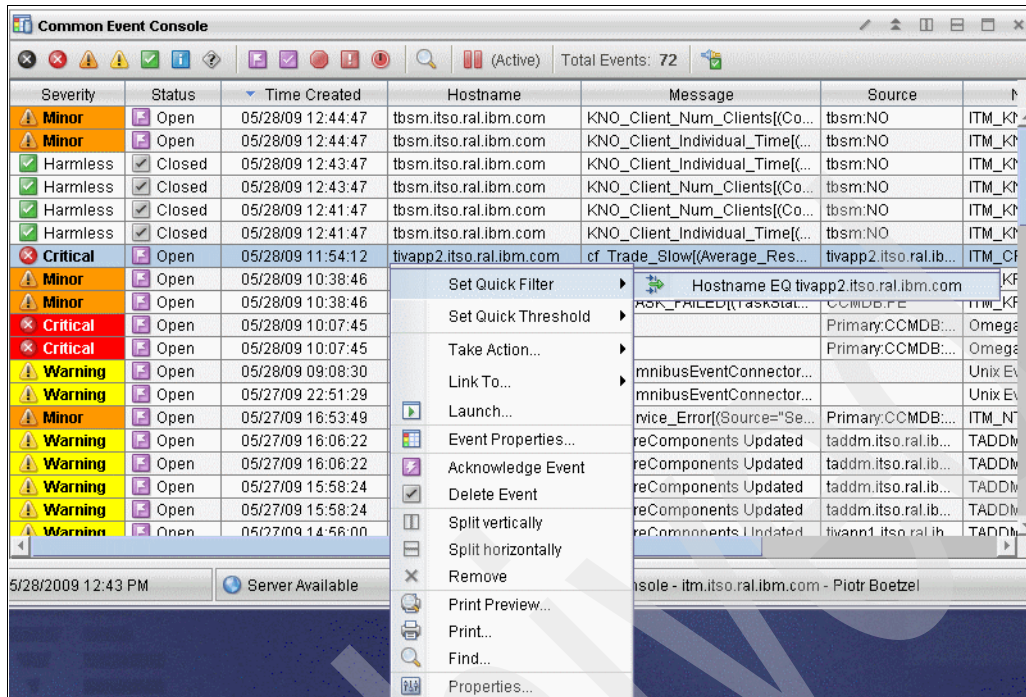


Figure 9-25 Common Event Console Filter menu

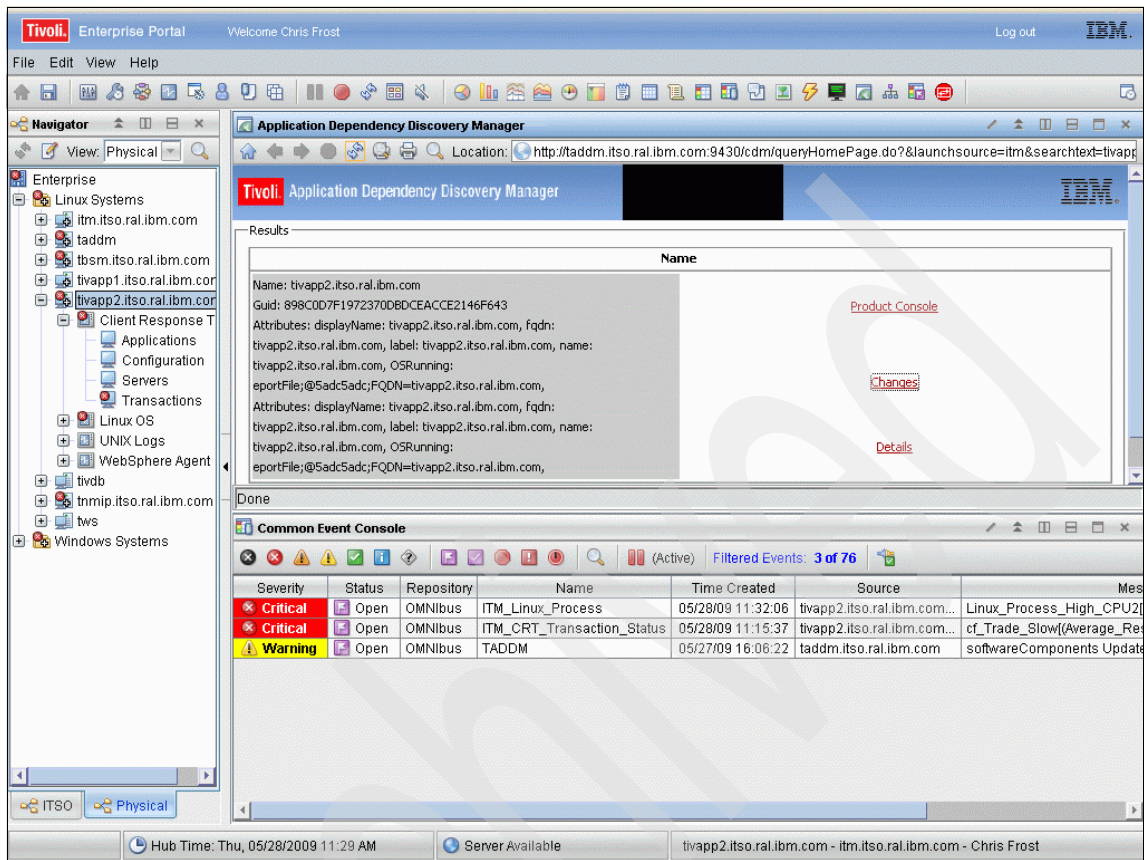


Figure 9-26 Common Event Console window in IBM Tivoli Monitoring

6. After filtering the information in the Common Event Console (Figure 9-26), the operator notices three events, including a Linux process using too much CPU, the slow transaction error, and an alert from a recent change to the application server that was received from IBM Tivoli Application Dependency Discovery Manager.

Because there are no related errors coming from the IBM Tivoli Composite Application Manager for WebSphere agent or network events, the Operator decides to investigate the change alert from IBM Tivoli Application Dependency Discovery Manager. The timing of the change closely matches the time when the trade transactions began to slow down.

This same workspace also has an IBM Tivoli Application Dependency Discovery Manager window, as described in 8.2.2, “Launching an IBM Tivoli Application Dependency Discovery Manager window within Tivoli Enterprise Portal” on page 347. From here, the operator can easily click the **changes**

link to display recent changes from the target system. Figure 9-27 shows the resulting window.

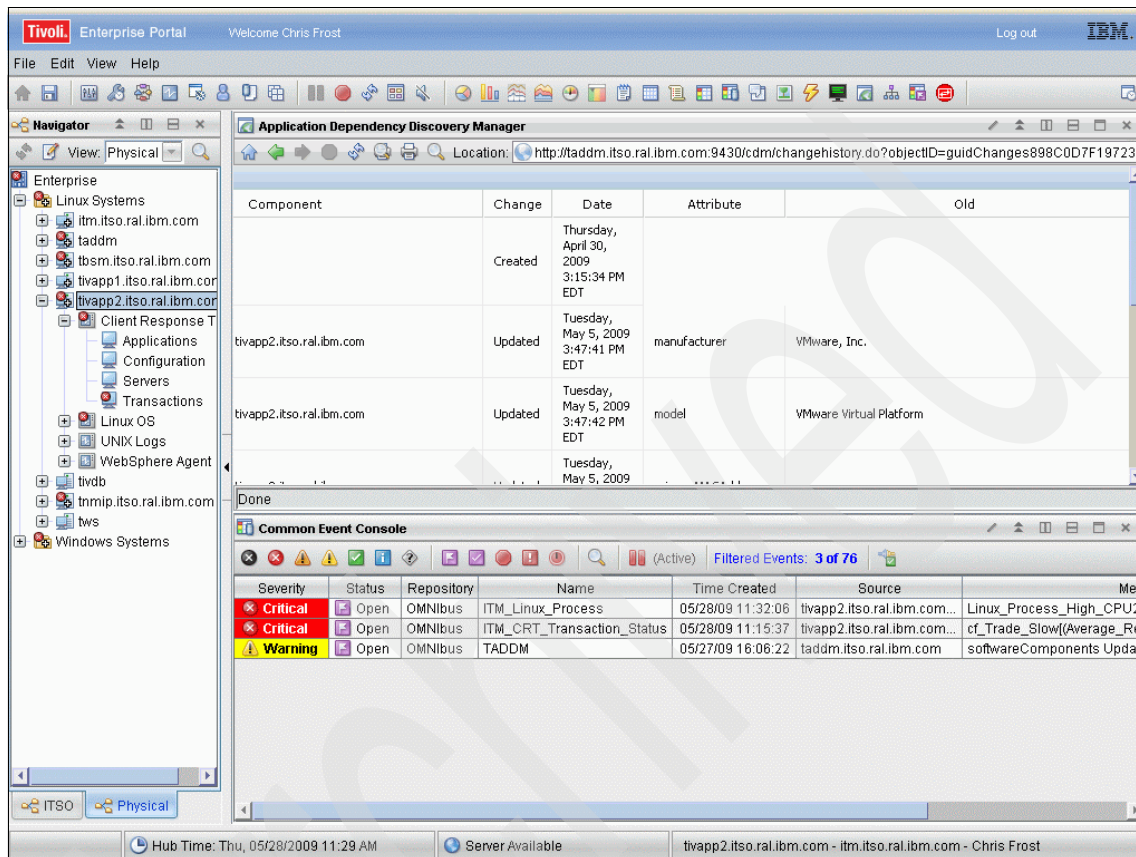


Figure 9-27 IBM Tivoli Application Dependency Discovery Manager Changes in IBM Tivoli Monitoring

7. After examining the change history, the operator decides that one of the changes, an updated configuration file, was the cause of the incident. The next step is to open Tivoli Provisioning Manager to determine whether the previous version of the configuration file can be rolled back.

The operator browses to the Tivoli Provisioning Manager workspace that was created in IBM Tivoli Monitoring (see 8.2.3, “Launching IBM Tivoli Provisioning Manager” on page 350).

In this case, the workspace has been created under the Linux OS agent.

8. Figure 9-28 on page 395 shows the resulting window. Notice how the operator has jumped straight to a relevant window in IBM Tivoli Provisioning Manager with `tivapp2.itso.ral.ibm.com` already selected. The automatic

sign-on has occurred as described in 4.4, “Tivoli Process Automation Engine security setup” on page 73.

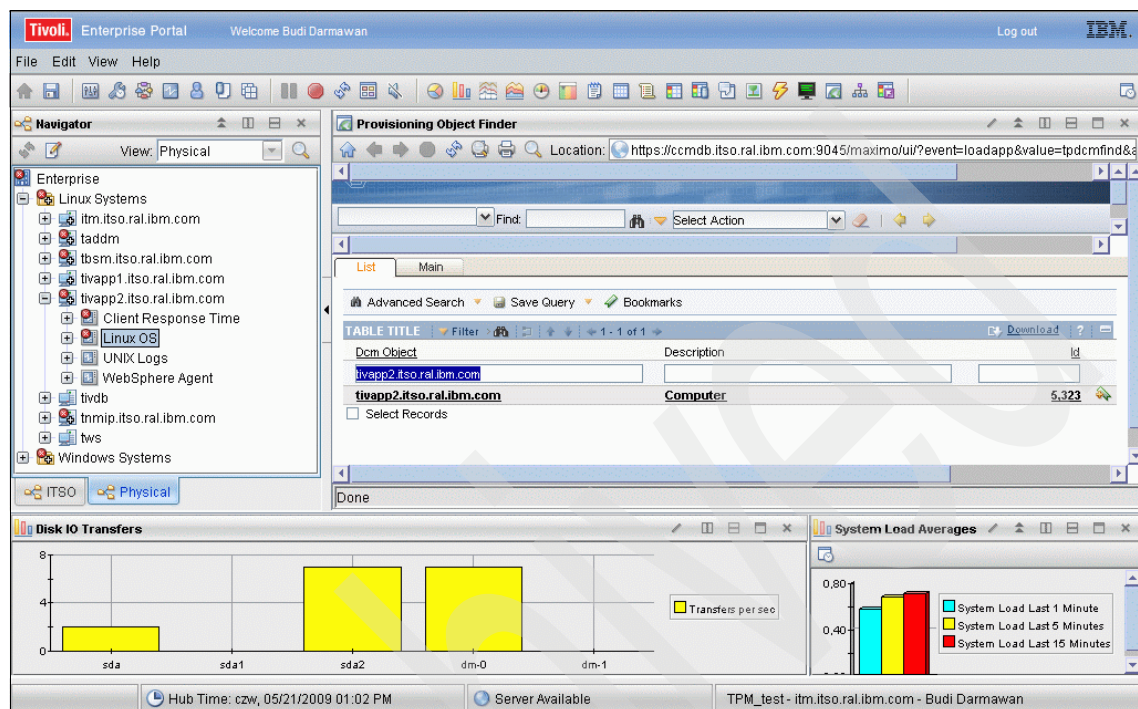


Figure 9-28 IBM Tivoli Provisioning Manager window in Tivoli Enterprise Portal

- By clicking the host name, the operator can now see more detailed information about the server, as shown in Figure 9-29 on page 396.

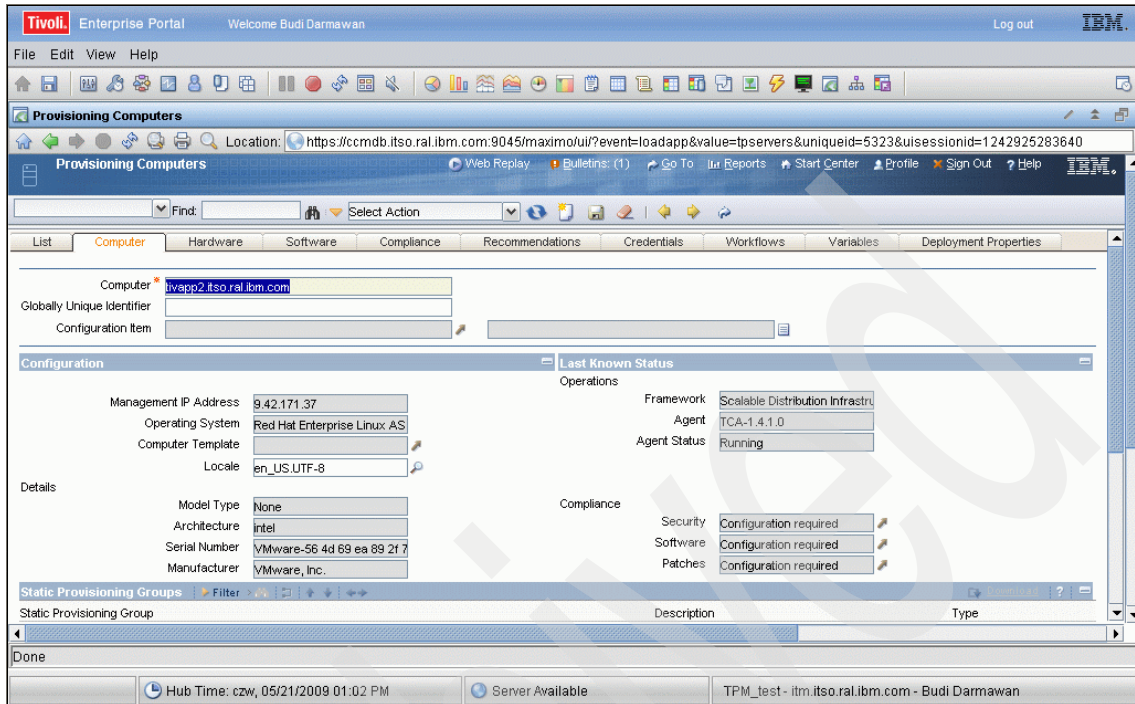


Figure 9-29 IBM Tivoli Provisioning Manager computer information in Tivoli Enterprise Portal

Note: The internal browser of the current Tivoli Enterprise Portal does not allow you to scroll the Software installation tab of IBM Tivoli Provisioning Manager LIC from Tivoli Enterprise Portal. Launch the IBM Tivoli Provisioning Manager console rather than launching IBM Tivoli Provisioning Manager LIC to avoid inconvenience.

10. The operator launches the IBM Tivoli Provisioning Manager console from Tivoli Enterprise Portal. Right-click **tivapp2.itso.ral.ibm.com**, and select **Launch**. Then, choose **LIC_to_TPM**, and click **Launch**, as shown in Figure 9-30 on page 397.

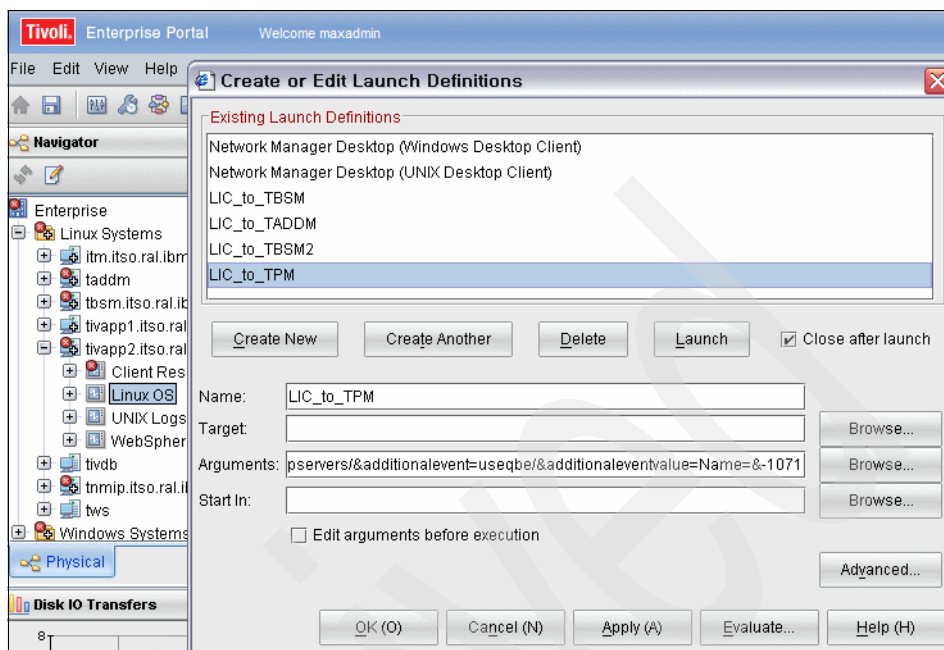


Figure 9-30 Launching IBM Tivoli Provisioning Manager

11. The operator wants to know what software is currently installed on the server and what packages are available, so the operator clicks the **Software** tab, as shown in Figure 9-31 on page 398.

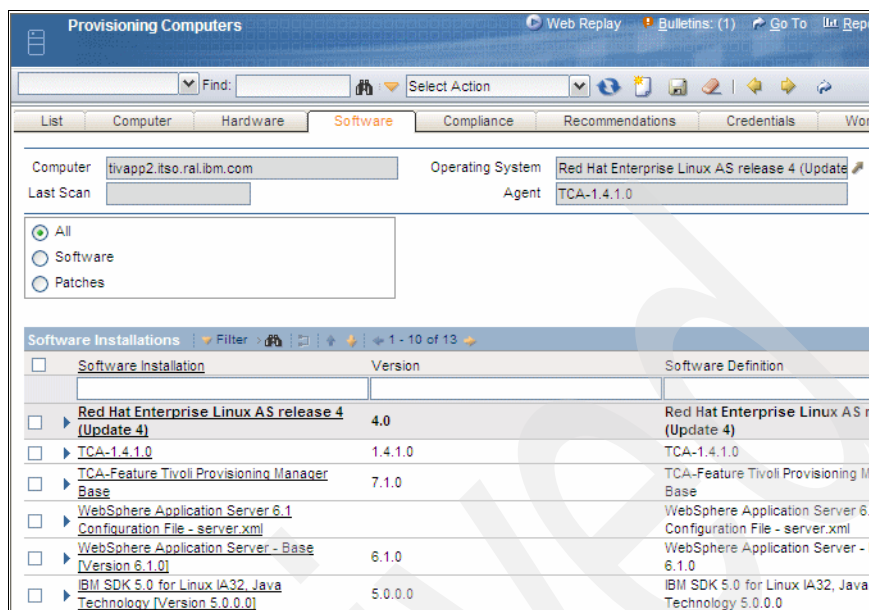


Figure 9-31 Software Components in IBM Tivoli Provisioning Manager

12. The operator discovers that the configuration file of WebSphere Application Server is installed on the machine by IBM Tivoli Provisioning Manager and assumes that someone updated the file after deployment. The operator decides to replace the configuration file on the machine with the configuration file in IBM Tivoli Provisioning Manager, because it is the master configuration file.
13. First, the operator wants to check which applications might be affected by a change to the application server, so the operator launches IBM Tivoli Application Dependency Discovery Manager in context to check the dependencies. The operator selects **Select Action** → **Launch Topology**, as shown in Figure 9-32 on page 399.

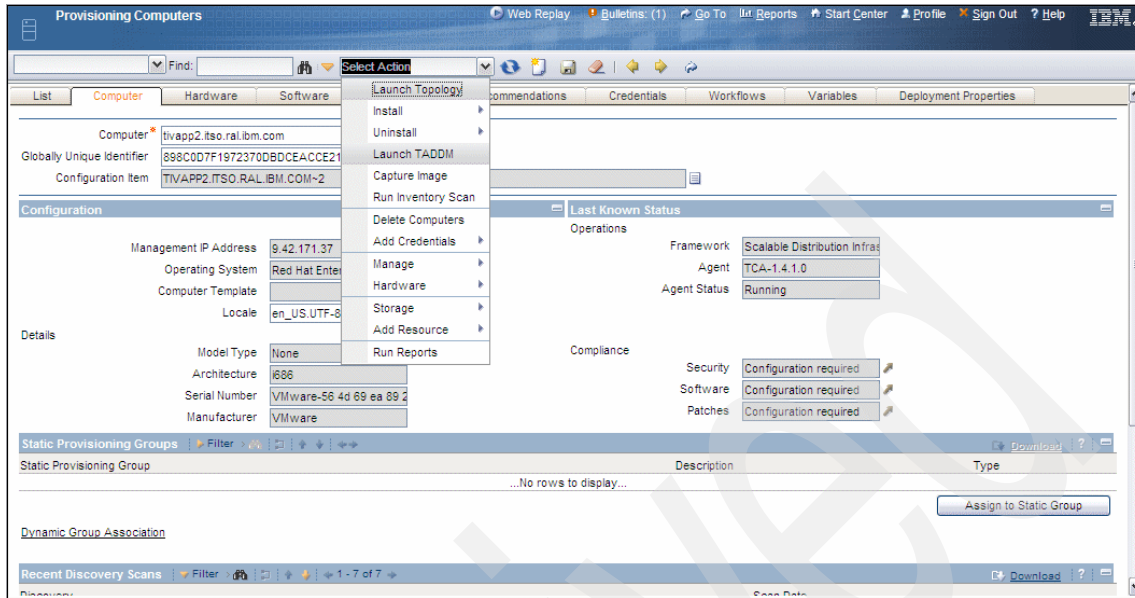


Figure 9-32 Launching IBM Tivoli Application Dependency Discovery Manager from IBM Tivoli Provisioning Manager

14. This action displays an IBM Tivoli Application Dependency Discovery Manager window in the context of the application server, as shown in Figure 9-33 on page 400. Satisfied that the change will have no adverse effects on any other applications, the operator switches back to the IBM Tivoli Provisioning Manager window to begin the software installation.

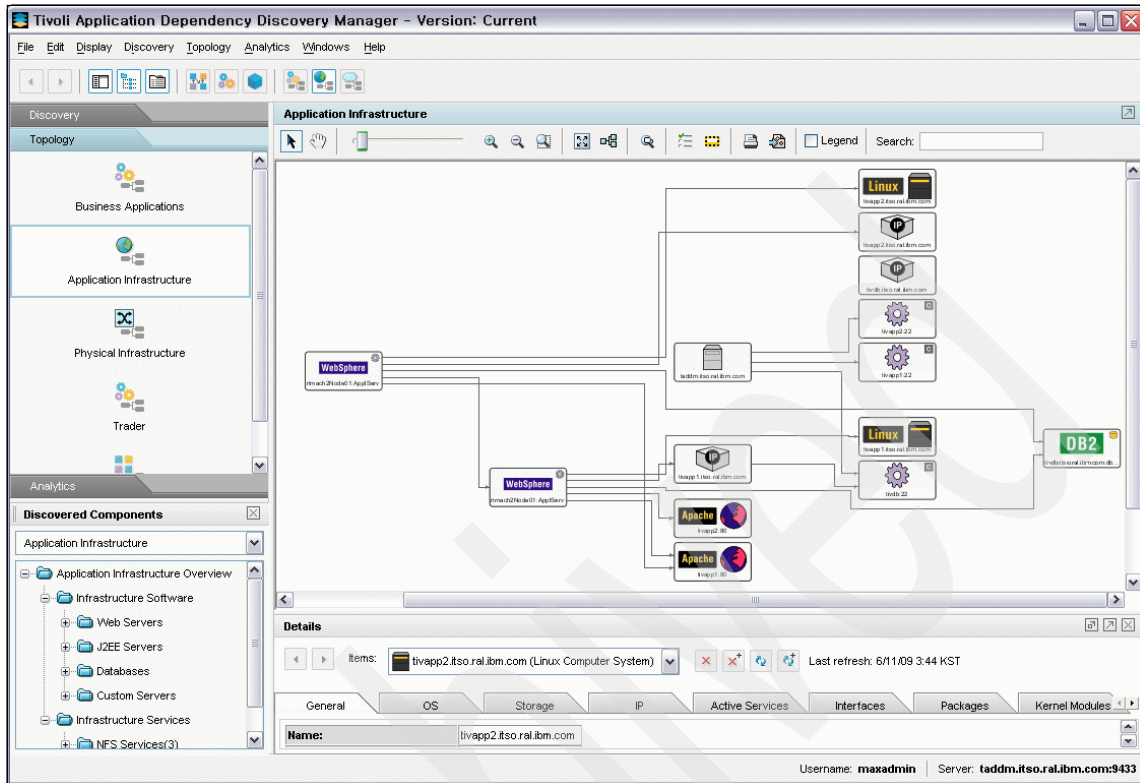


Figure 9-33 IBM Tivoli Application Dependency Discovery Manager dependency view

15. The operator selects **Select Action** → **Install Software Product**. The operator clicks **Select Software** to select WebSphere Application Server 6.1 Configuration file as shown in Figure 9-34 on page 401. The operator submits the software installation task.

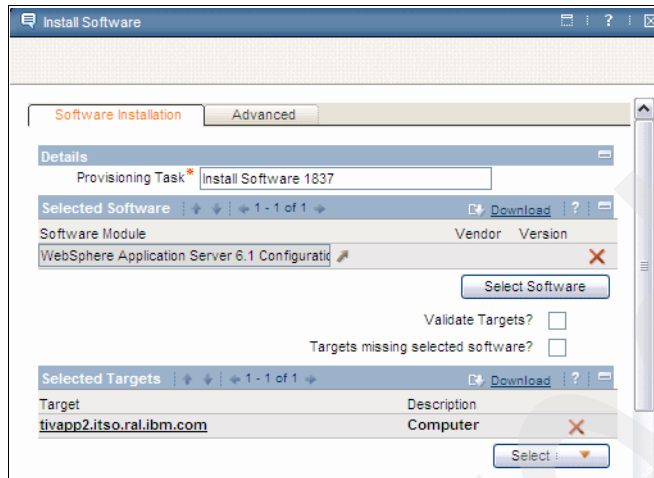


Figure 9-34 Finding WebSphere Application Server Configuration file

16. The operator verifies whether the deployment task succeeded, as shown in Figure 9-35.

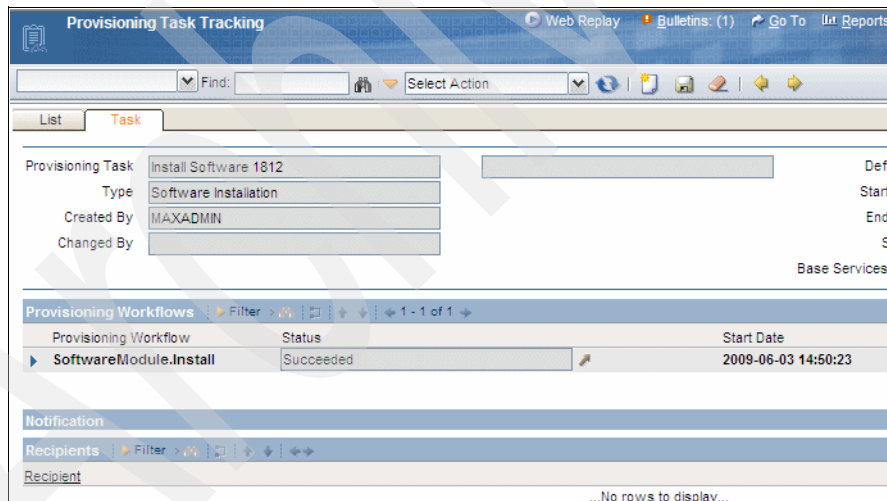


Figure 9-35 Checking deployment task

9.4 Summary of benefits

The product integrations that are demonstrated in this scenario reflect two major benefit categories:

- ▶ Reduce mean time to recovery (MTTR):
 - Integrating the events from IBM Tivoli Network Manager for IP, IBM Tivoli Monitoring, IBM Tivoli Composite Application Manager, and IBM Tivoli Application Dependency Discovery Manager into IBM Tivoli Netcool/OMNIBus and displaying those events in IBM Tivoli Monitoring through the Common Event Console allow the operator to quickly visualize every item related to the incident and to quickly identify the root cause. Without these technologies, this task takes considerably longer, because the operator must switch between product consoles and try to match events, logs, and time stamps.
 - The Single Sign-On feature reduces the need for multiple sets of credentials to be remembered and entered when checking data from various sources.
 - The Single Sign-On and Launch In Context features also enable portlets from products, such as IBM Tivoli Application Dependency Discovery Manager, IBM Tivoli Network Manager for IP, and IBM Tivoli Provisioning Manager to display in Tivoli Enterprise Portal, as though they were part of the product. This seamless integration allows you to redefine generic workspaces so that operators have all the information they need in one place without needing the knowledge and time to browse between multiple products.
- ▶ Reduce costs:
 - The integrations into Tivoli Enterprise Portal that allow the operator to browse directly to relevant workspaces in other platforms mean that less operator training is required.
 - Less time taken to diagnose and resolve a problem means staff can be more effectively utilized and might mean less overtime cost.
 - Automated provisioning reduces time and labor cost and minimizes the number of errors.

While there have been significant benefits demonstrated, there is still room to improve. This scenario lacks a business view of the Trade application, allowing operators, business service owners, and management to get a real-time view.

Executive IBM Tivoli Business Service Manager scenario

In this chapter, we describe an environment where we use IBM Tivoli Business Service Manager for an executive interface to look at the overall system health. The system health is provided by various monitoring subsystems, such as IBM Tivoli Network Manager for IP, Tivoli Application and Dependency Discovery Manager, and IBM Tivoli Service Request Manager.

We discuss these topics in this chapter:

- ▶ 10.1, “Scenario overview” on page 404
- ▶ 10.2, “Scenario setup” on page 405
- ▶ 10.3, “Sample walk-through” on page 405
- ▶ 10.4, “Summary of benefits” on page 410

10.1 Scenario overview

This scenario describes an executive using IBM Tivoli Business Service Manager to detect and diagnose a system or device problem by using integration with other Tivoli operational management tools. The monitored environment runs the Trade application, which runs on WebSphere servers. Figure 10-1 shows the scenario environment.

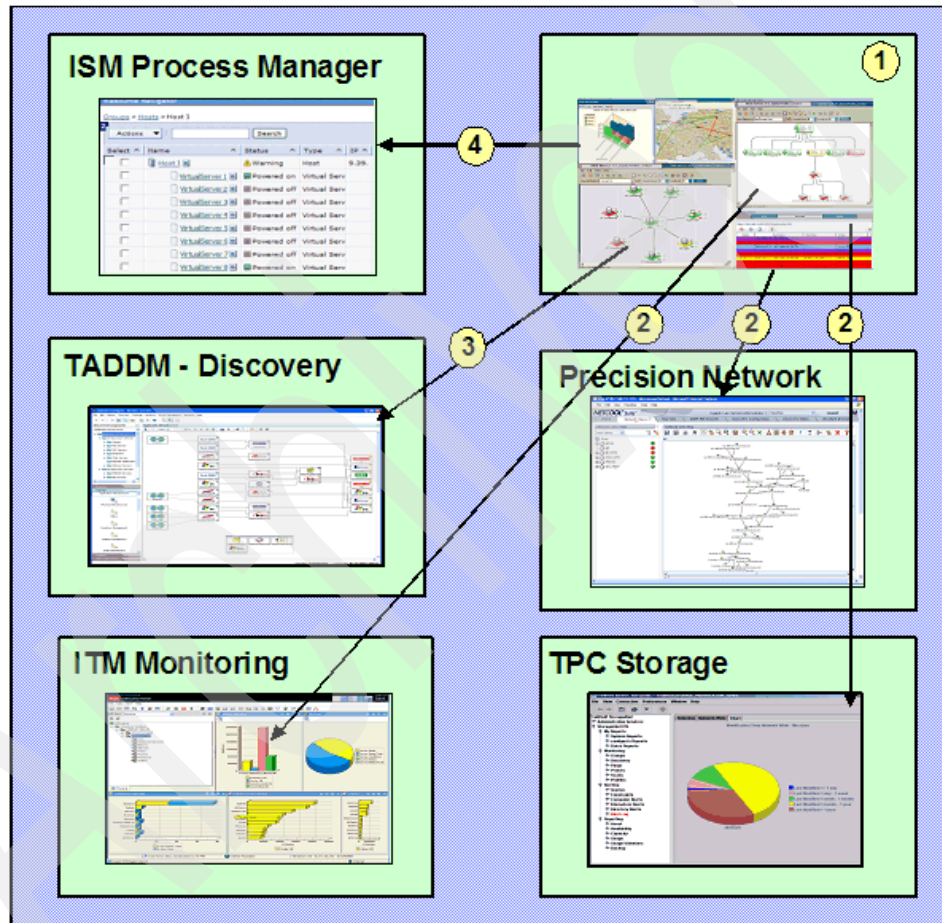


Figure 10-1 Scenario flow

This scenario includes the following products:

- ▶ IBM Tivoli Business Service Manager V4.2
- ▶ IBM Tivoli Network Manager for IP V3.8
- ▶ Tivoli Application and Dependency Discovery Manager

- ▶ IBM Tivoli Netcool/OMNIBus V7.2.1

10.2 Scenario setup

You must perform these tasks first:

- ▶ Security setup for single sign-on as discussed in 4.6, “IBM Tivoli Netcool products LDAP configuration” on page 95 and 4.5, “IBM Tivoli Application Dependency Discovery Manager security setup” on page 92
- ▶ Data integration as discussed in 5.2, “Resource data integration” on page 135
- ▶ Navigation integration as discussed in 6.3.5, “IBM Tivoli Business Service Manager” on page 296 and target URL definitions from 6.2.4, “Tivoli Integrated Portal” on page 249, and 6.2.1, “IBM Tivoli Application Dependency Discovery Manager” on page 233

This setup uses the application monitoring environment and resource monitoring environment that are integrated in Chapter 8, “Operational drill down” on page 341 and Chapter 9, “Automated application management scenario” on page 365.

10.3 Sample walk-through

A proactive executive becomes concerned after hearing of a performance problem at a competitor. To be assured that the company’s operations are fine, the executive uses the company portal to log in to IBM Tivoli Business Service Manager to see how reliably the company’s customer-facing applications performing. The executive is able to easily browse through a single dashboard depicting a representation of how the company’s applications and networking equipment interconnect. Real-time application performance and availability information is displayed, assuring the executive that their customers are not experiencing the problems that were seen at the competitor. Using SSO (single sign-on), the executive can then “drill down” into underlying applications to view systems and network inter-connectivity (IBM Tivoli Network Manager for IP), application and system configurations and relationships (IBM Tivoli Application Dependency Discovery Manager), and systems and application performance (IBM Tivoli Monitoring):

1. Concerned that a recent event might have an impact on the company’s operations, the executive uses SSO to log in to IBM Tivoli Business Service Manager to view the real-time status of the company’s customer-facing applications.

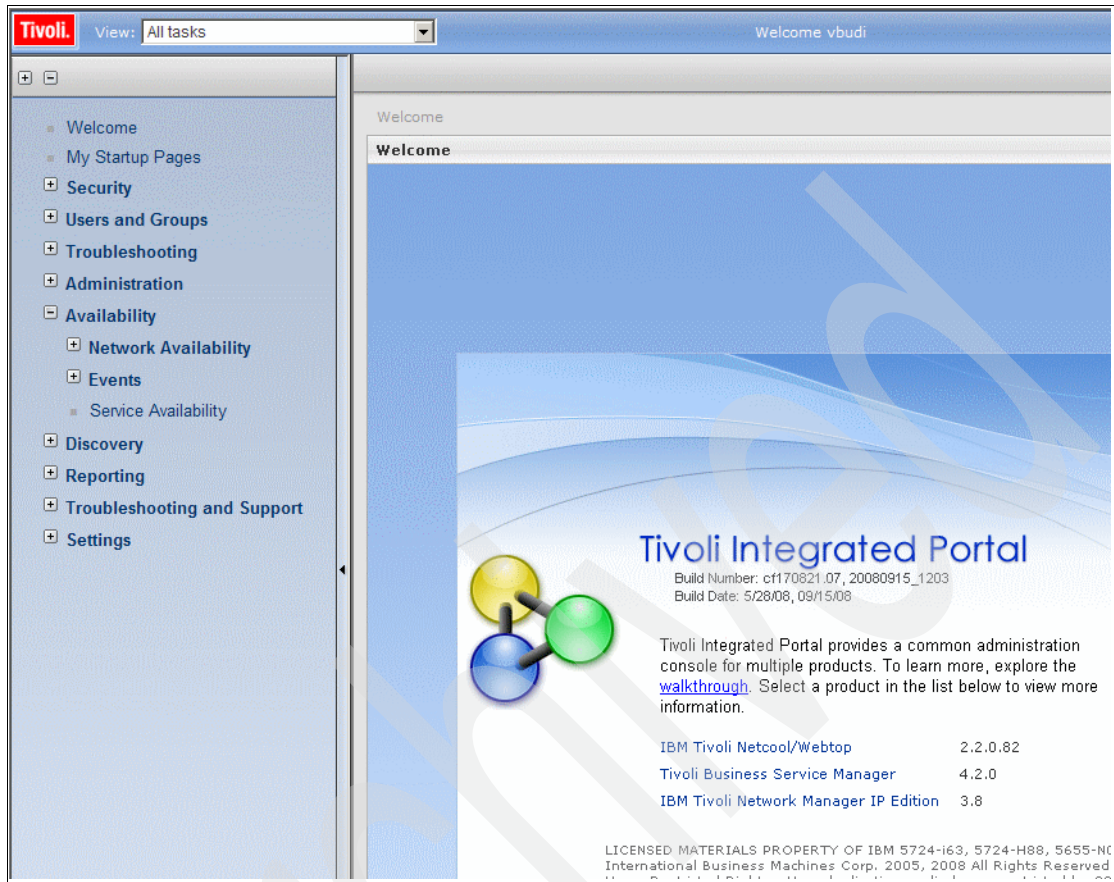


Figure 10-2 Tivoli Integrated Portal Initial view

2. The executive uses IBM Tivoli Business Service Manager to “drill down” into IBM Tivoli Monitoring to view the performance-related data for a new customer-facing application. The executive can see the current and historical details for how the application operates.

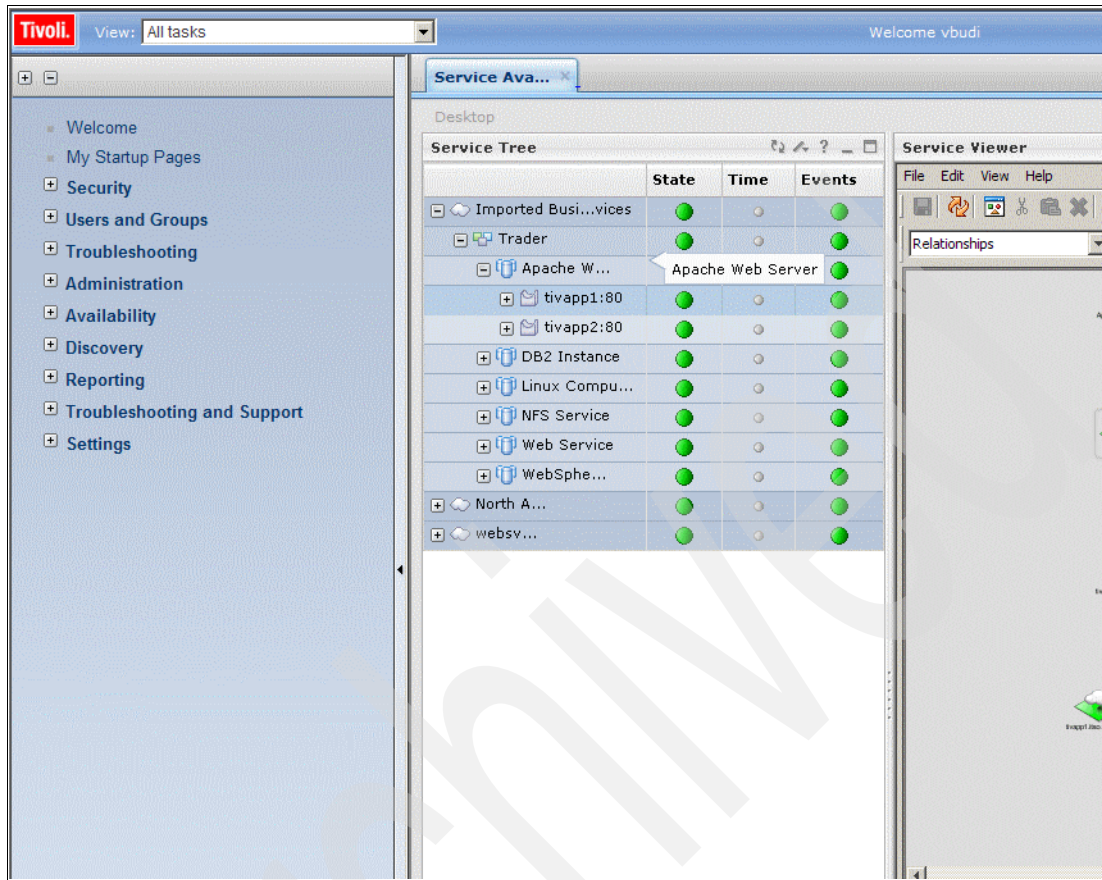


Figure 10-3 IBM Tivoli Business Service Manager Service Tree

3. Satisfied that the new customer application operates well, the executive uses IBM Tivoli Business Service Manager to navigate into IBM Tivoli Network Manager for IP to view the network topology and device details for the network supporting the customer application. The executive is able to see real-time status and device information showing the current availability, and resiliency, of the network on which their customers now depend.

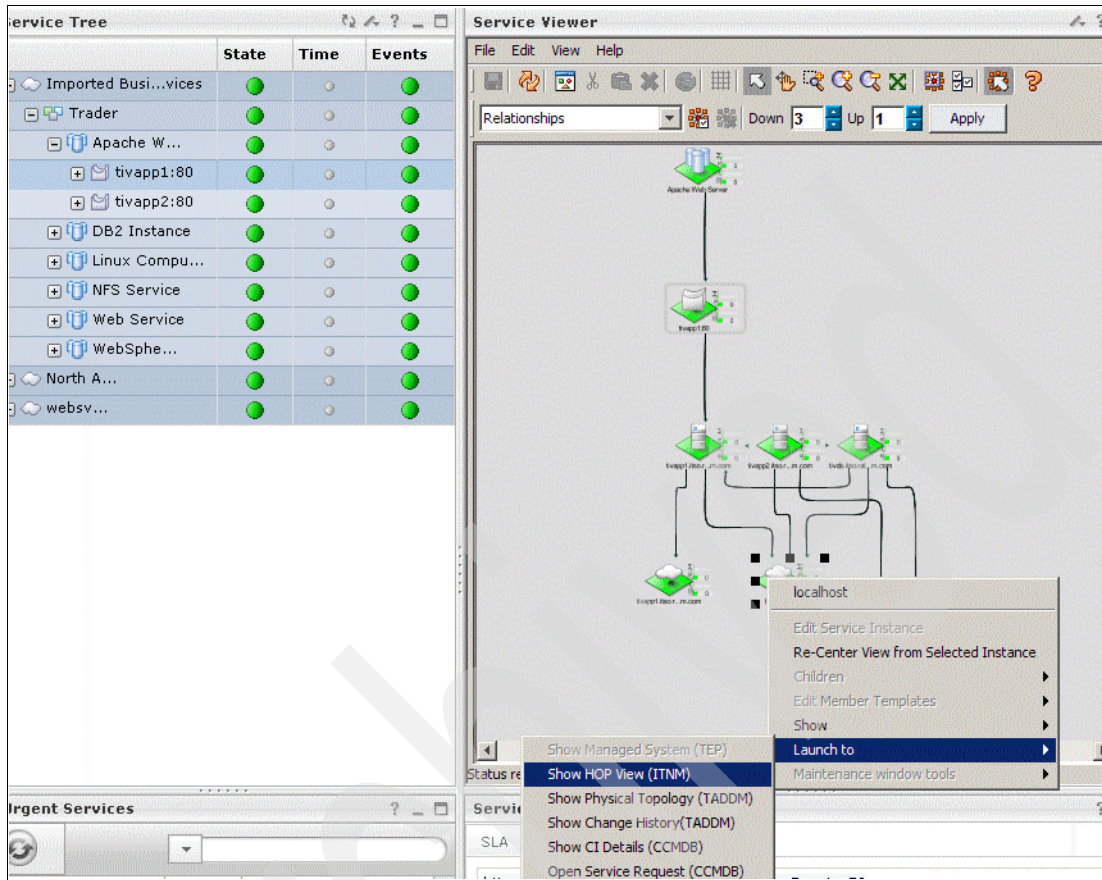


Figure 10-4 IBM Tivoli Business Service Manager Launch to IBM Tivoli Network Manager for IP

4. Not quite convinced that everything is perfect, the executive can navigate from IBM Tivoli Business Service Manager into IBM Tivoli Application Dependency Discovery Manager to see the current configuration and inter-connectivity of the multiple systems (Web server, database, and so forth) that make up the customer application. Using IBM Tivoli Application Dependency Discovery Manager, the executive can review analytics and recent changes to see what corrections or improvements, as well as any corporate policy violations, have been implemented on the system.

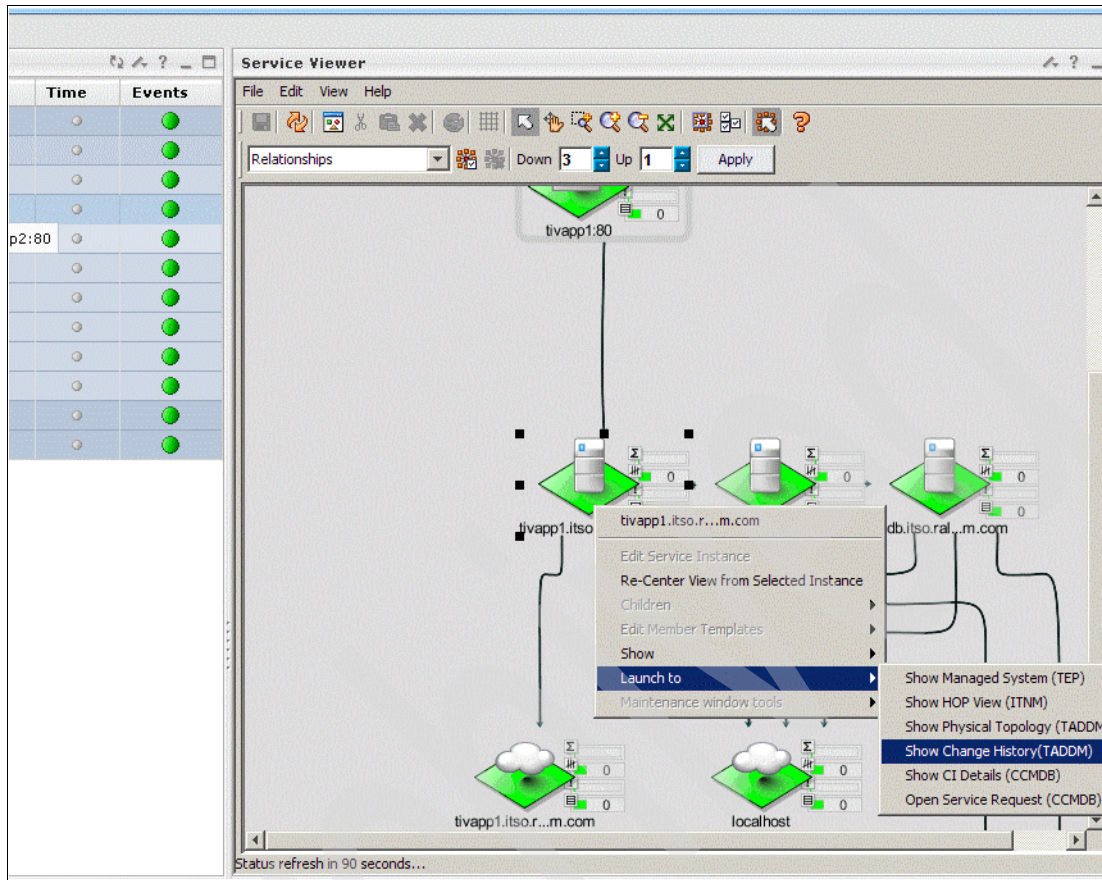


Figure 10-5 IBM Tivoli Business Service Manager Launch to IBM Tivoli Application Dependency Discovery Manager

5. If the executive discovers a system change or policy violation that needs to be reviewed, the executive can launch into IBM Tivoli Service Request Manager to create a service request. Navigating from IBM Tivoli Business Service Manager into IBM Tivoli Service Request Manager will pre-populate the service request with the system or details, leaving the executive to only have to type a short comment before clicking submit.

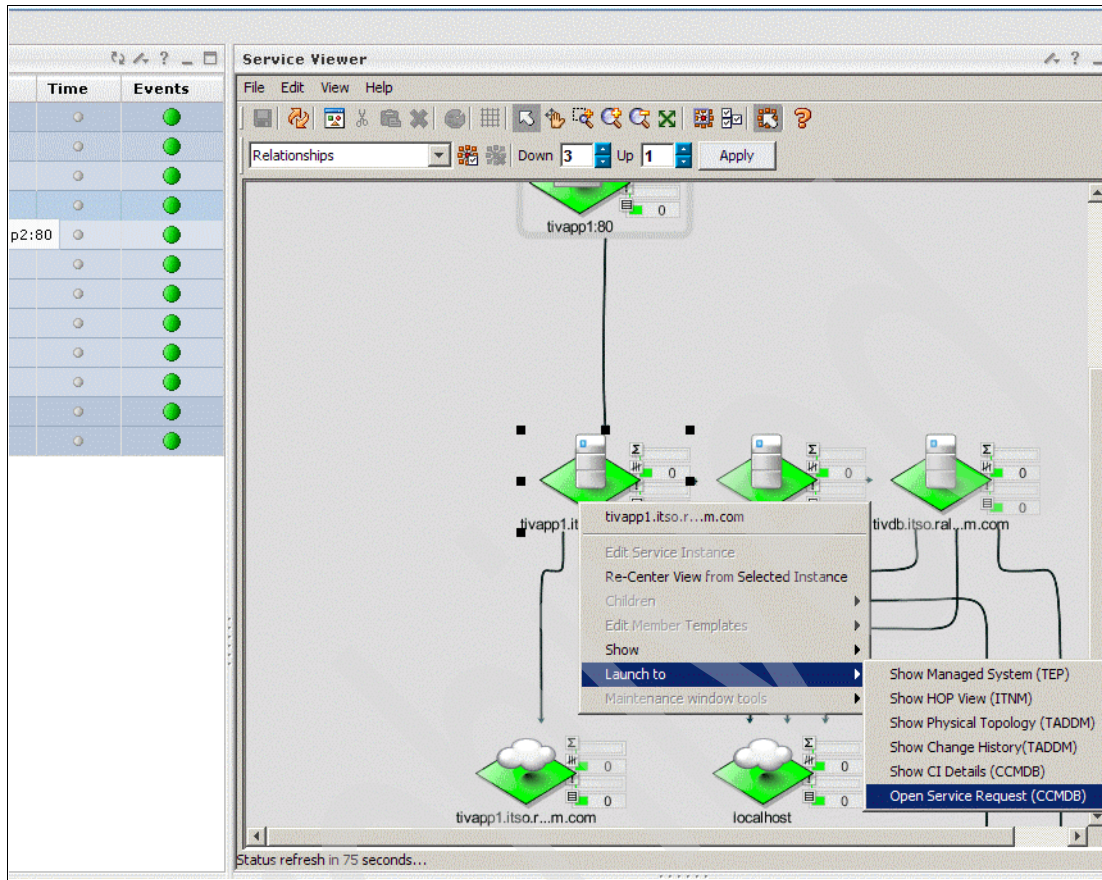


Figure 10-6 IBM Tivoli Business Service Manager Launch to IBM Tivoli Service Request Manager

6. Any system or device problem that is resolved, such as through a IBM Tivoli Service Request Manager service request closure, is reflected in the IBM Tivoli Business Service Manager status for the system or device.

10.4 Summary of benefits

The benefits of the product integrations demonstrated in this scenario fall under these categories:

- ▶ Executive Overview and Service Assurance:
 - Integrations with IBM Tivoli Monitoring, IBM Tivoli Network Manager for IP, IBM Tivoli Application Dependency Discovery Manager, and IBM Tivoli

Service Request Manager allow the executive to quickly determine that everything is operating smoothly across multiple management applications.

- The Single Sign-On feature reduces the need for multiple sets of credentials to be remembered and entered when checking data from multiple sources.

Archived

Change and release management scenario

In this chapter, we discuss the implementation of the change and release management scenario.

We discuss the following sections:

- ▶ 11.1, “Scenario overview” on page 414
- ▶ 11.2, “Scenario setup” on page 417
- ▶ 11.3, “Sample walk-through” on page 418
- ▶ 11.4, “Summary of benefits” on page 438

11.1 Scenario overview

In this scenario, we describe deploying the WebSphere Application Server configuration file, which is software, by a change request, to be used by IBM Service Management solutions.

A problem was identified by a operator through IBM Tivoli Business Service Manager, and the operator asked to deploy the master WebSphere Application Server configuration file. This point is the starting point of this scenario.

Note: Release Process Manager V 7.1.1.3 does not support the IBM Tivoli Provisioning Manager 7.1 integration module, so we do not use IBM Tivoli Provisioning Manager integration module features for this scenario. In our environment, IBM Tivoli Provisioning Manager, Red Hat Package Manager, and IBM Tivoli Change and Configuration Management Database are installed on the same machine.

Figure 11-1 shows the scenario overview.

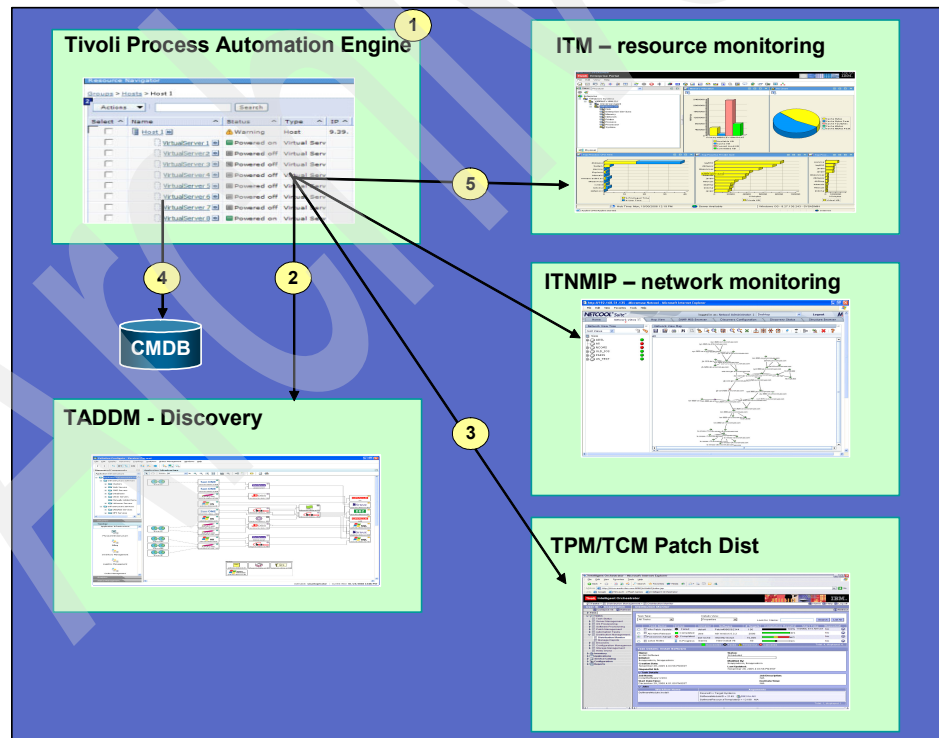


Figure 11-1 Change and release management overview

11.1.1 Products involved

This scenario includes the following products:

- ▶ Tivoli Application and Dependency Discovery Manager V 7.1.2
- ▶ Tivoli Provisioning Manager V 7.1
- ▶ IBM Tivoli Change and Configuration Management Database V 7.1.1.3
- ▶ Discovery Library Adapters
- ▶ IBM Tivoli Monitoring

11.1.2 Adoption route

In this section, we discuss the best practices in the adoption route of the scenario. This adoption route allows a fast delivery of a services engagement based on a known installation experience. Figure 11-2 depicts the steps for implementing the scenario.

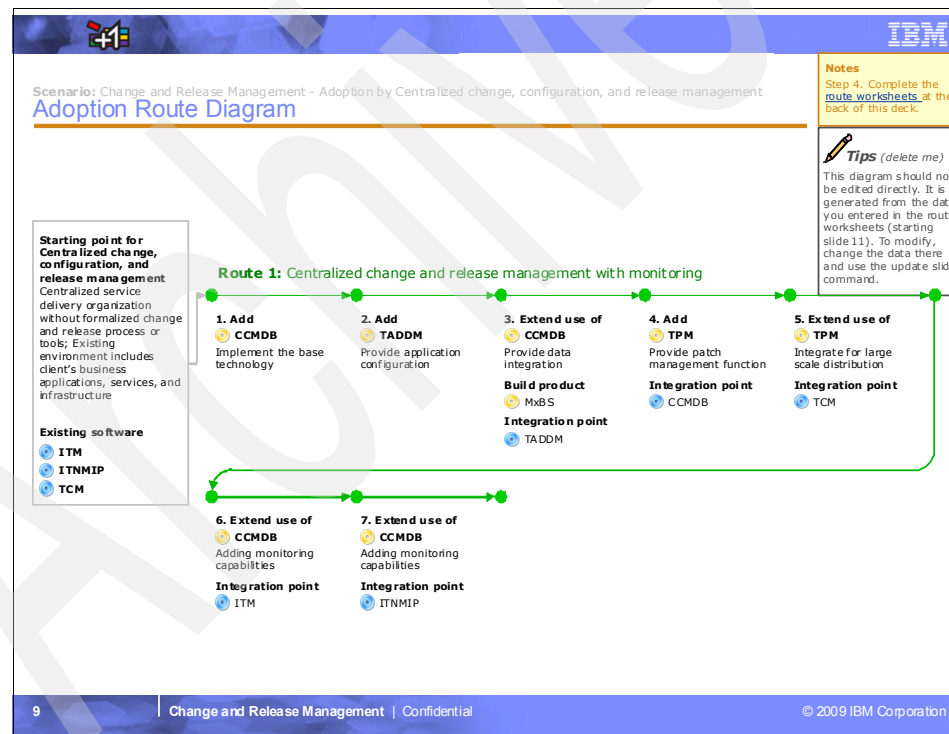


Figure 11-2 Adoption route

The scenario provides the following logical functions that are shown in Figure 11-3.

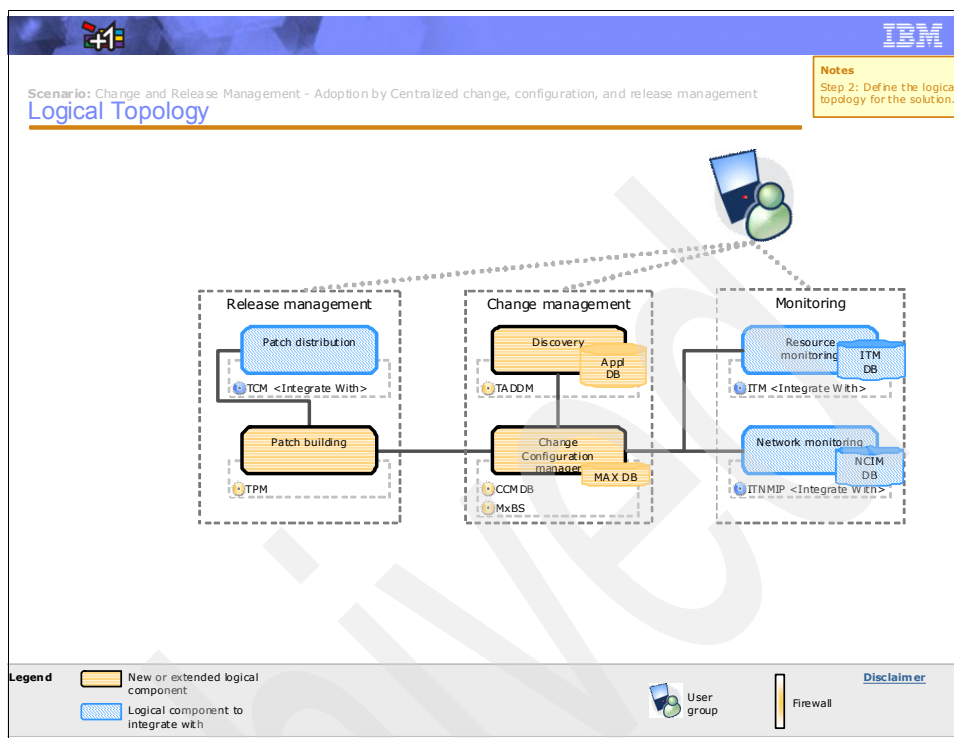


Figure 11-3 Logical topology

From the logical topology, we can develop a physical topology of the implementation. A large enterprise implementation has different components than a proof of concept installation. The physical topology that we provide in Figure 11-4 on page 417 is related to the proof of concept environment. Consult the appropriate manuals for designing an enterprise implementation by considering performance balancing, server capacity, and solution reliability.

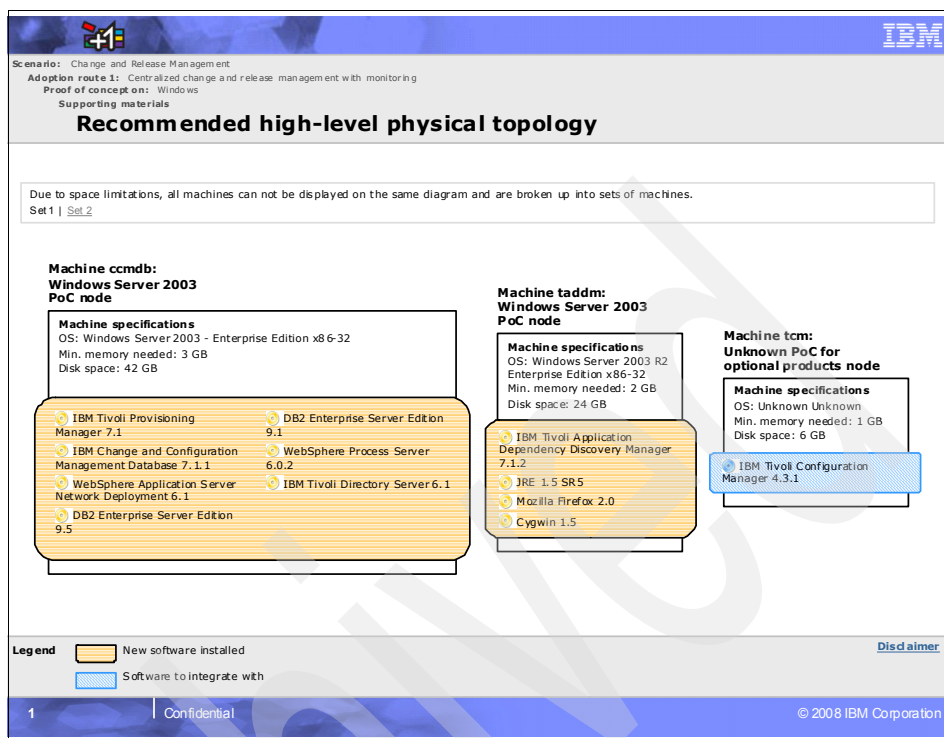


Figure 11-4 Proof of concept physical topology

11.2 Scenario setup

We need to perform these prerequisite steps to run this scenario:

1. Generate IBM Tivoli Provisioning Manager Discovery Library Adapter using the DiscoveryLibraryAdapter workflow and import the Identity Markup Language (IDML) file to IBM Tivoli Application Dependency Discovery Manager. Follow the instructions in 5.2.3, "IBM Tivoli Provisioning Manager integration" on page 142.
2. Generate the IDML file by using IBM Tivoli Monitoring Discovery Library Adapter and import the IDML file to IBM Tivoli Application Dependency Discovery Manager. Follow the instructions in 5.2.4, "IBM Tivoli Monitoring integration" on page 150.
3. Populate IBM Tivoli Application Dependency Discovery Manager data to IBM Tivoli Change and Configuration Management Database using IBM Tivoli Integration Composer. Follow the instructions in 5.2.1, "IBM Tivoli Integration Composer" on page 136.

4. After populating IBM Tivoli Application Dependency Discovery Manager data to IBM Tivoli Change and Configuration Management Database, we can see IBM Tivoli Application Dependency Discovery Manager data by using the actual configuration items (Actual CIs) application from IBM Tivoli Change and Configuration Management Database. We need to promote Actual CIs to Authorized CIs. Refer to 5.2.2, “Promoting Actual CIs to Authorized CIs” on page 137.
5. We need to make CIs known to IBM Tivoli Provisioning Manager so that data model computers can be linked to them using IBM Tivoli Application Dependency Discovery Manager discovery in IBM Tivoli Provisioning Manager. Follow the instructions in 5.2.3, “IBM Tivoli Provisioning Manager integration” on page 142.

11.3 Sample walk-through

The scenario consists of the following steps:

1. The operator submits a request for change (RFC) for changing a WebSphere Application Server configuration file.
2. The Change Owner reviews the RFC and accepts.
3. The Change Manager reviews the RFC and creates a Work Order and tasks to handle this request.
4. The Change Manager examines the impact of the change.
5. The Change Manager views the problem on the WebSphere Application Server through Tivoli Enterprise Portal.
6. The Change Manager approves the RFC.
7. The Deployment Specialist is notified of a new task.
8. The Deployment Specialist implements the task using Tivoli Provisioning Manager

The scenario begins with an operator creating a Request for Change (RFC):

1. The operator creates an RFC from the Start Center. Click **New Process Request** as shown in Figure 11-5 on page 419.

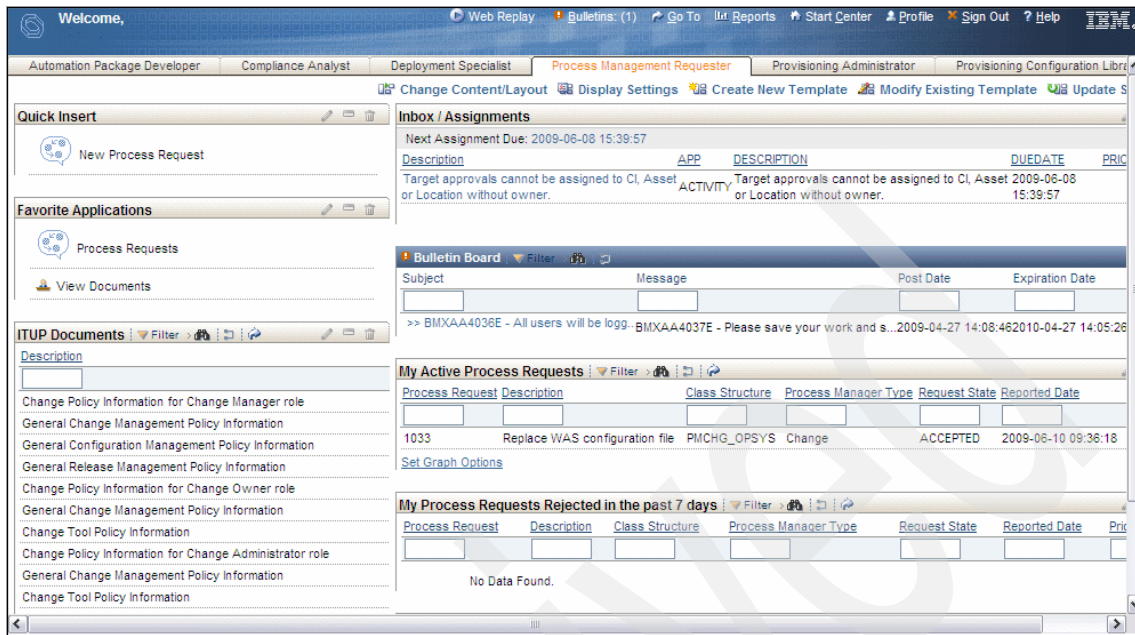


Figure 11-5 New Process Request

2. The operator specifies the request as shown in Figure 11-6 on page 420 and clicks **Submit**:

Description: Request Summary

Detail: Request Detail Information

Configuration Item: Click on Arrow Icon beside Configuration Item and choose the Configuration Item. In our case, it's TIVAPP2.ITS0.RAL.IBM.COM~2.

Target CIs: Choose the target CIs. In our case, it's TIVAPP2.ITS0.RAL.IBM.COM~2.

Process Requests

Find: [] Select Action []

Requestor: MAXADMIN
Name: MAXADMIN
Phone: []
E-mail: []

Process Request Details

Description: Replace WAS configuration file
Priority: 1
Process Manager Type: Change
Site: DFT
Requested Completion: 2009-06-10 23:37:31
Asset: []
Location: []
Configuration Item: TIVAPP2.ITSO.RAL.IBM.COM-2

Request Classification Details

Classification: PMCHG \ PMCHGSFW \ OPSYSYSTEM
Class Description: Operating System

Classification Attributes

Attribute	Description
PKGNAME	Name
VERSION	Version

Target CIs

Asset	Location	Configuration Item	Target Description
[]	[]	TIVAPP2.ITSO.RAL.IBM.COM-2	[]

Buttons: Select, Clear All, New Row

Figure 11-6 Create a Process Request

3. The operator can see the request from this Start Center, as shown in Figure 11-7.

My Active Process Requests

Process Request	Description	Class Structure	Process Manager Type	Request State	Reported Date	Priority
1033	Replace WAS configuration file	PMCHG_OPSYS	Change	SUBMITTED	2009-06-10 09:36:18	1

My Process Requests Rejected in the past 7 days

No Data Found.

My Draft Process Requests

Process Request	Description	Class Structure	Process Manager Type	Request State	Reported Date	Priority

Figure 11-7 Submitted process request

4. The Change Owner sees the new RFC, as shown in Figure 11-8 on page 421, and opens the RFC.

My Overdue or past due start Changes				
PMCHGPROGRESS	Priority	Scheduled Finish	Scheduled Start	Change
No Data Found.				

All Submitted RFCs				
Process Request	Description	Priority	Reported By	Rep
1033	Replace WAS configuration file	1	MAXADMIN	2009-06-10 23:37:31
Set Graph Options				

Overdue or past due start Activities for my Changes					
Priority	Scheduled Finish	Scheduled Start	Status	Parent	Activity
No Data Found.					

Figure 11-8 RFC

- The Change Owner reviews the RFC and accepts it, as shown in Figure 11-9.

The screenshot displays the 'Process Requests' interface. At the top, there's a navigation bar with links like 'Web Replay', 'Bulletins: (1)', 'Go To', 'List Reports', 'Start Center', 'Profile', 'Sign Out', and 'Help'. Below this is a search bar and a 'Select Action' dropdown. The main content area is divided into several sections:

- Process Request Summary:** Shows 'Process Request: 1033', 'Owner: [blank]', 'Owner Group: [blank]', 'Process State: SUBMITTED', and 'Attachments: [blank]'.
- User Information:** Includes fields for 'Requestor: MAXADMIN', 'Name: MAXADMIN', 'Phone: [blank]', and 'E-mail: [blank]'.
- Process Request Details:** Contains 'Description: Replace WAS configuration file', 'Priority: 1', 'Process Manager Type: Change', 'Site: DFT', 'Requested Completion: 2009-06-10 23:37:31', 'Asset: [blank]', 'Location: [blank]', and 'Configuration Item: TIVAPP2.ITSO.RAL.IBM.COM~2'.
- Request Classification Details:** Shows 'Classification: PMCHG\PMCHGSFW\OPSYSTEM' and 'Class Description: Operating System'.
- Classification Attributes:** A table with columns 'Attribute' and 'Description'. It lists 'PKGNAME' and 'VERSION'.
- Target CIs:** A section at the bottom with a 'Filter' dropdown and a 'Download' button.

Figure 11-9 Change Owner reviews the RFC

- The Change Owner chooses **Review and refine the new Change Work Order**, as shown in Figure 11-10 on page 422. In our case, the same person is responsible for the Change Manager and Change Owner roles.

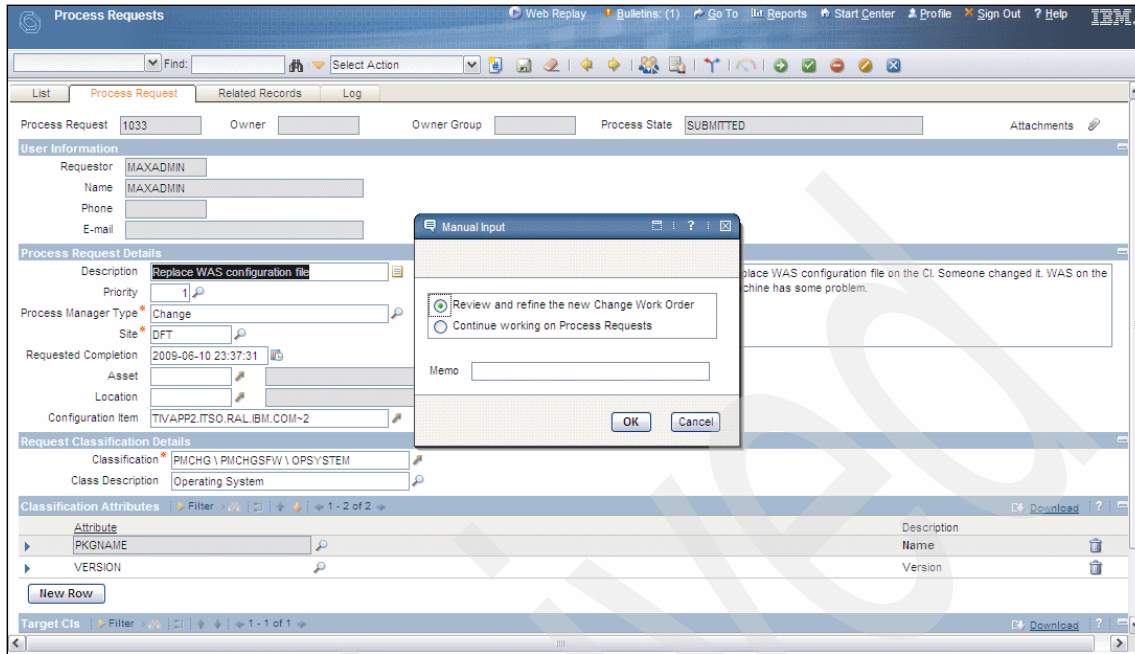


Figure 11-10 Choose Review and refine the new Change Work Order

7. The Change Manager sees the change, as shown in Figure 11-11 on page 423.

The screenshot displays the IBM Change Manager web application. The top navigation bar includes links for Web Replay, Bulletins (1), Go To, Reports, Start Center, Profile, Sign Out, and Help. Below this is a toolbar with various icons for actions like Find, Select Action, and navigation. The main interface is divided into tabs: List, Change, Plans, Related Records, Actuals, Log, Specifications, Process Details, and Impact Analysis. The 'Change' tab is active, showing details for change 1044, owned by MAXADMIN, with a status of WAPPR and progress of ACCEPTED.

Change Details

Summary: Replace WAS configuration file

Classification: PMCHG \ PMCHGSFW \ OPSYSTEM

Class Description: Operating System

Priority: 1

Priority Justification:

Service Group:

Service:

Vendor:

Site: DFT

SLA Applied?

Under Flow Control? ☒

Suspend Flow Control?

Flow Action:

Flow Action Assist?

Launch Entry Name:

Predecessors:

BPEL Enabled?

BPEL In Progress?

Configuration Item: TIVAPP2.ITSO.RAL.IBM.COM~2

GL Account:

Target Description:

Asset/Location Priority:

Change Type:

Reason for Change:

Verification:

Risk Assessment:

Back Out Plan:

Source Cts:

CI Number	Description	Classification	Source Description	Sequence	Progress?
...No rows to display...					

Target Cts, Assets and Locations:

Figure 11-11 Change Manager reviews the change

- The Change Manager creates the Work Order for the RFC. From the **Select Action** menu, choose **Create** → **Work Order**. Click the **Related Records** tab to see the new work order, as shown in Figure 11-12 on page 424.

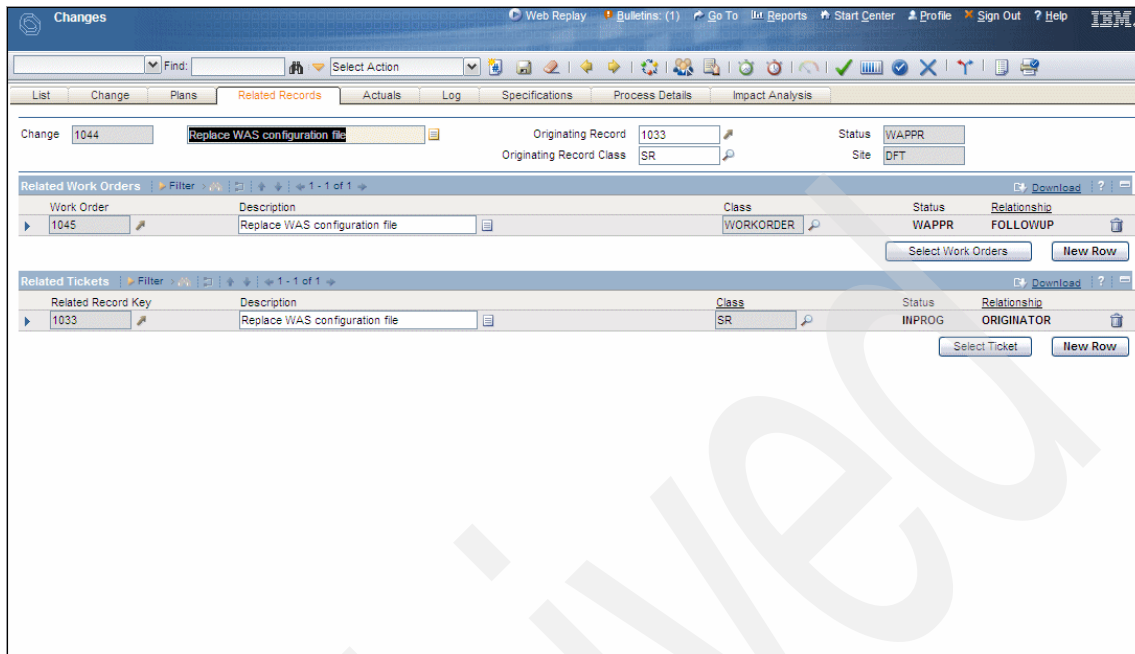


Figure 11-12 Created Work Order

- The Change Manager creates tasks for the Work Order. Click the arrow icon beside the Work Order field in Related Work Orders and choose **Go To Work Order Tracking**. The Change Manager sees the Work Order, as shown in Figure 11-13 on page 425, and clicks the **Plans** tab.

The screenshot displays the IBM Work Order Tracking application. The main form is titled 'Work Order Tracking' and includes a 'Find:' field and a 'Select Action' dropdown. The interface is organized into several sections:

- Work Order Details:** Includes fields for Work Order (1045), Location, Asset, Configuration Item (TIVAPP2.ITSO.RAL.IBM.COM~2), Parent WO, Classification, Class Description, and Launch Entry Name.
- Job Details:** Includes fields for Job Plan, PM, Safety Plan, and Contract.
- Asset Details:** Includes fields for Asset Up?, Warranties Exist?, SLA Applied?, and Charge to Store?.
- Priority:** Includes fields for Asset/Location Priority, Priority (1), Priority Justification, and Risk Assessment.
- Multiple Assets, Locations and CIs:** Includes a table with columns for Asset, Location, Configuration Item, Target Description, Sequence, Progress?, and Site. A 'Download' button is also present.
- Scheduling Information:** Includes fields for Target Start, Target Finish, Scheduled Start, Actual Start, Actual Finish, and Duration (0:00).
- Follow-up Work:** Includes fields for Originating Record (1044), Originating Record Class (CHANGE), and Has Follow-up Work?.

The bottom status bar shows 'Changes > Work Order Tracking'.

Figure 11-13 Work order tracking

10. The Change Manager creates two tasks by clicking **Examine impact of this change** and **Install software**. The first task is manual work. The Change Manager needs to review the change impact. The software is installed using Tivoli Provisioning Manager. Choose **New Row** in the Tasks for Work Order section. Provide the following information, as shown in Figure 11-14 on page 426:

Classification Description : Choose TPSWINSTALL from Select Value Pop up window. TPTASK \ TPSWINSTALL will be assigned to Classification automatically.

Click **Save**.

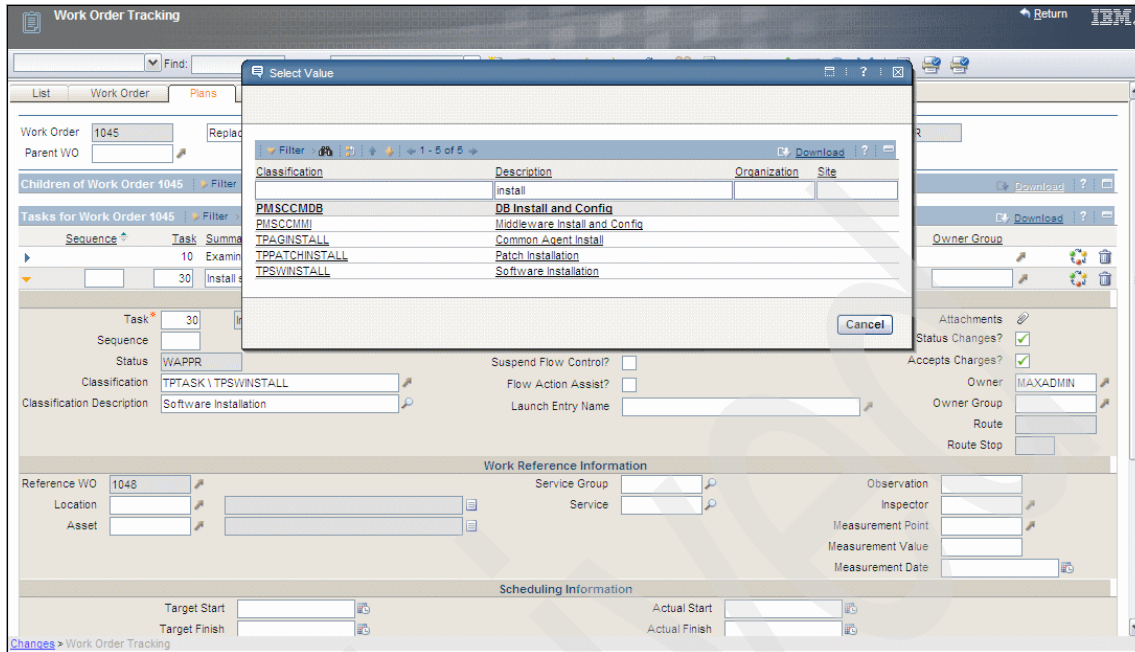


Figure 11-14 Create tasks for the work order

11. From the Reference WO field, click **Go To Activities and Tasks**. You can see the Activities and Tasks view. In the Assisted Workflow field, assign **TPCREATASK** workflow. Specify the change schedule, and click **Save**, as shown in Figure 11-15 on page 427.

The screenshot displays the 'Activities and Tasks' application interface. The top navigation bar includes a 'Return' link and a search bar. Below the navigation bar, there are tabs for 'List', 'Activities and Tasks', 'Resources', 'Related Records', and 'Log'. The main content area is divided into several sections:

- Task Information:** Includes fields for Task (30), Location, Asset, Configuration Item (TIVAPP2.ITSO.RAL.IBM.COM-2), Target Description, Classification (TPTASK\TPSWINSTALL), Class Description (Software Installation), Launch Entry Name, and Parent Process (1045).
- Activity Type:** Includes fields for Activity Type (1045), Priority (1), Reported By (MAXADMIN), and checkboxes for SLA Applied?, Under Flow Control?, Suspend Flow Control?, Flow Action, Flow Action Assist?, and Implementation Task?
- Attachments:** Includes fields for Status (INPRG), Status Date (2009-06-10 10:32:), Service Group, Service, Owner (MAXADMIN), Owner Group, and Assisted Workflow (TPCREATASK). A 'Start Assisted Work' button is also present.
- Scheduling Information:** Includes fields for Target Start, Target Finish (2009-06-10 23:33:42), Scheduled Start (2009-06-10 22:33:56), Scheduled Finish (2009-06-10 23:00:20), Actual Start, Actual Finish, Estimated Duration (0:00), and Predecessors. A 'Launch Task Scheduler' button is located below this section.
- Multiple Assets, Locations and Cts:** A table with columns for Asset, Location, Configuration Item, and Target Description. The table currently shows 'No rows to display...'. Buttons for 'Select' and 'Clear All' are located to the right of the table.

The bottom of the window shows a breadcrumb trail: 'Changes > Work Order Tracking > Activities and Tasks'.

Figure 11-15 Assign assisted workflow and scheduling

12. Click the Return link in the upper-right corner.
13. Click the Return link again on the Work Order Tracking application. The Change Manager sees the change and chooses the **Impact Analysis** tab, as shown in Figure 11-16 on page 428.

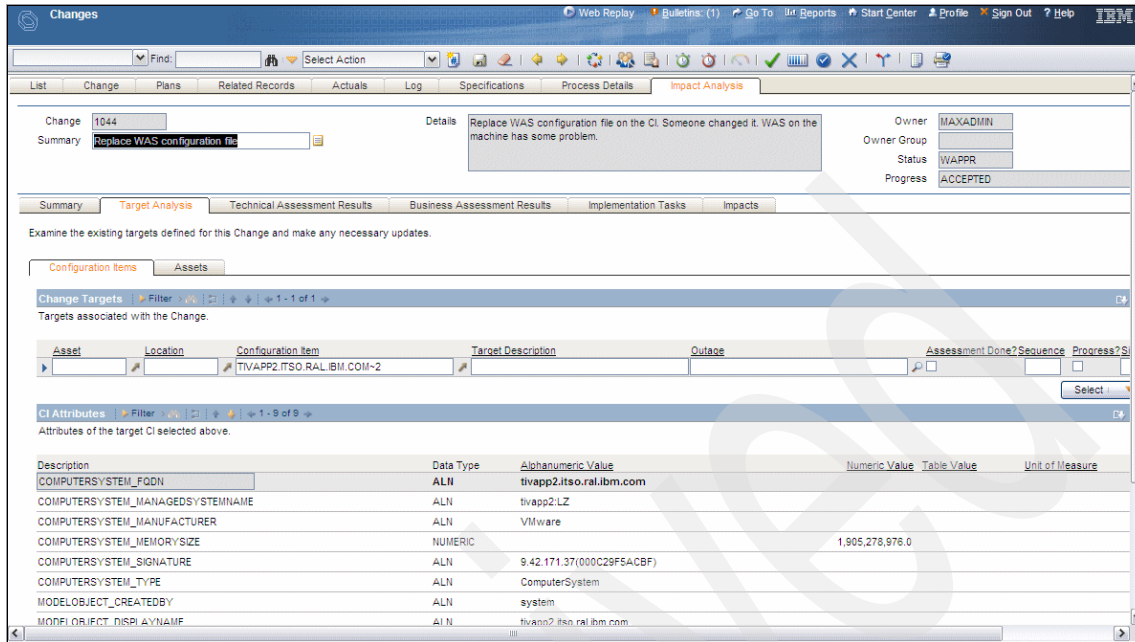


Figure 11-16 Impact Analysis

14. The Change Manager reviews the change impact and approves the change, as shown in Figure 11-17 on page 429.

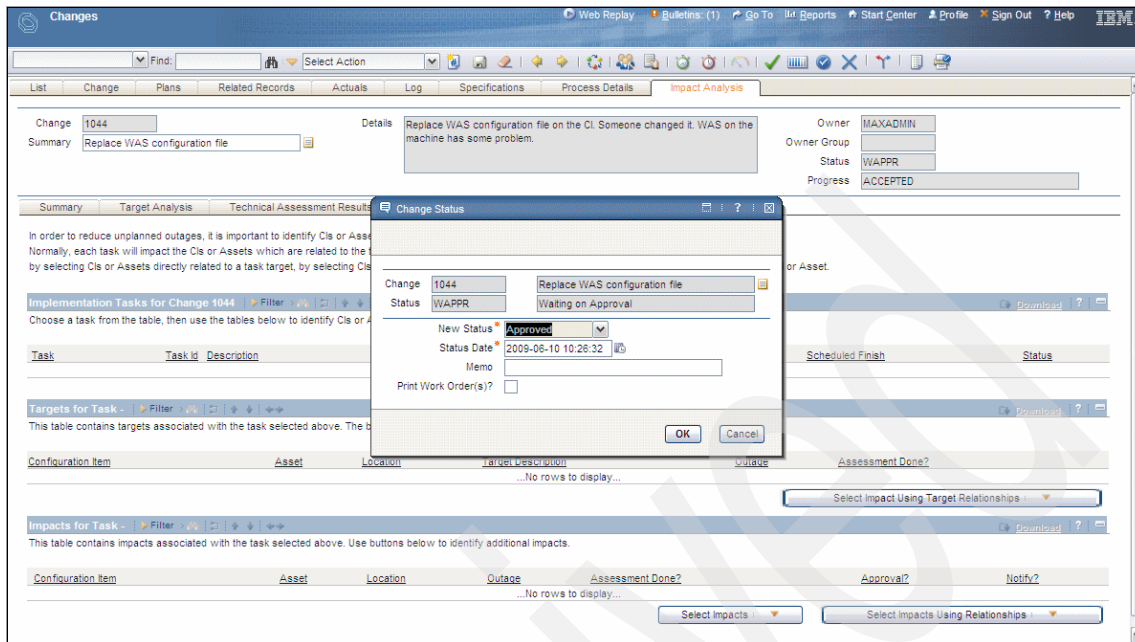


Figure 11-17 Approval of the change

15. From the IBM Tivoli Application Dependency Discovery Manager Console, run **Discovery**.

16. After the IBM Tivoli Application Dependency Discovery Manager discovery completes, run the IBM Tivoli Integration Composer mapping, as shown in Figure 11-18.

```
C:\IntegrationComposer\bin>commandLine.bat TADDM-CCMDB-Katory maximo
itso4you administration collation maximo itso4you
Integration Composer will now start
IBM Tivoli Integration Composer 7.1.13 Build 200809100832
Mapping: TADDM-CCMDB-Katory
Creating data source Fusion... done
IBM Tivoli Integration Composer database version: V7113-80
Retrieving mapping definition TADDM-CCMDB-Katory... done
Creating source data source TADDM-7.1-VBD-Source... done
Connecting to source data source TADDM-7.1-VBD-Source...
```

Figure 11-18 Running the mapping

17. From the Change tab, select the configuration item menu, and select **Go to Configuration Item**, as shown in Figure 11-19 on page 430.

Changes

Web Replay | Bulletins: (1) | Go To | All Reports | Start Center | Profile | Sign Out | Help

Find: | Select Action

List | **Change** | Plans | Related Records | Actuals | Log | Specifications | Process Details | Impact Analysis

Change: 1044 | Owner: MAXADMIN | Owner Group: | Status: APPR | Progress: ACCEPTED | Attachments

Change Details

Summary: Replace WAS configuration file

Parent: | Asset: | Location: | Configuration Item: TIVAPP2.ITSO.RAL.IBM.COM-2 | GL Account: | Target Description: | Asset/Location Priority: | Change Type: | Reason for Change: | Verification: | Risk Assessment: | Back Out Plan: |

Classification

Classification: PMCHG \ PMCHGSFW \ OPSYSTEM | Class Description: Operating System | Priority: 1 | Priority Justification: | Service Group: | Service: | Vendor: | Site: DFT | SLA Applied?: | Under Flow Control?: ☒ | Suspend Flow Control?: | Flow Action: | Flow Action Assist?: | Launch Entry Name: | Predecessors: | BPML Enabled?: | BPML In Progress?:

Source CIs | Filter | Description | Classification | Source Description | Sequence | Progress?

Figure 11-19 Change detail information

18. In the Configuration Items window, click **Select Actions** and select **View Actual CI Topology** → **Business Applications** to view the changes in IBM Tivoli Application Dependency Discovery Manager (Figure 11-20 on page 431).



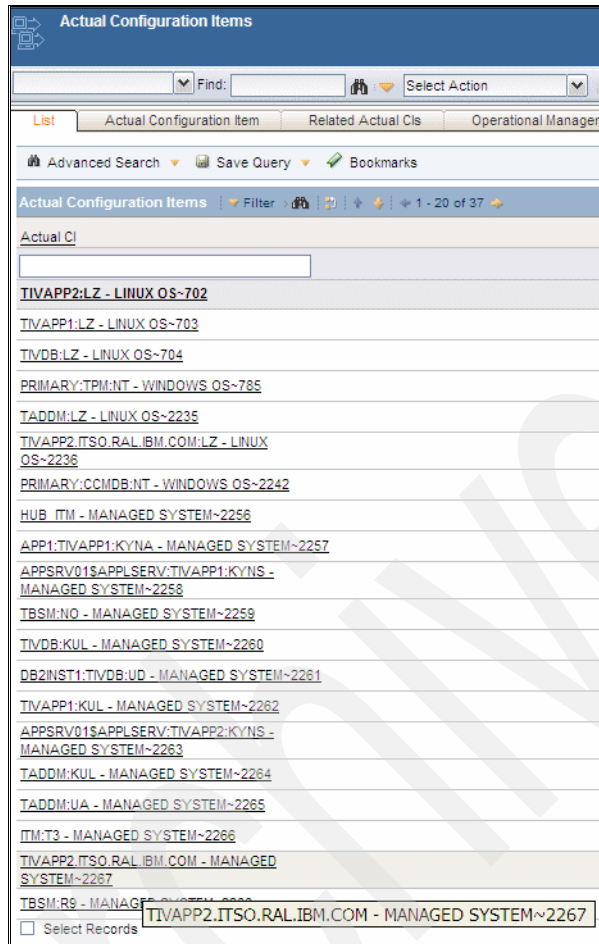


Figure 11-21 Choosing the CI that will be launched into IBM Tivoli Monitoring

20. Click **Select Action** and click **Tivoli Monitoring 6.2.2** (Figure 11-22 on page 433).

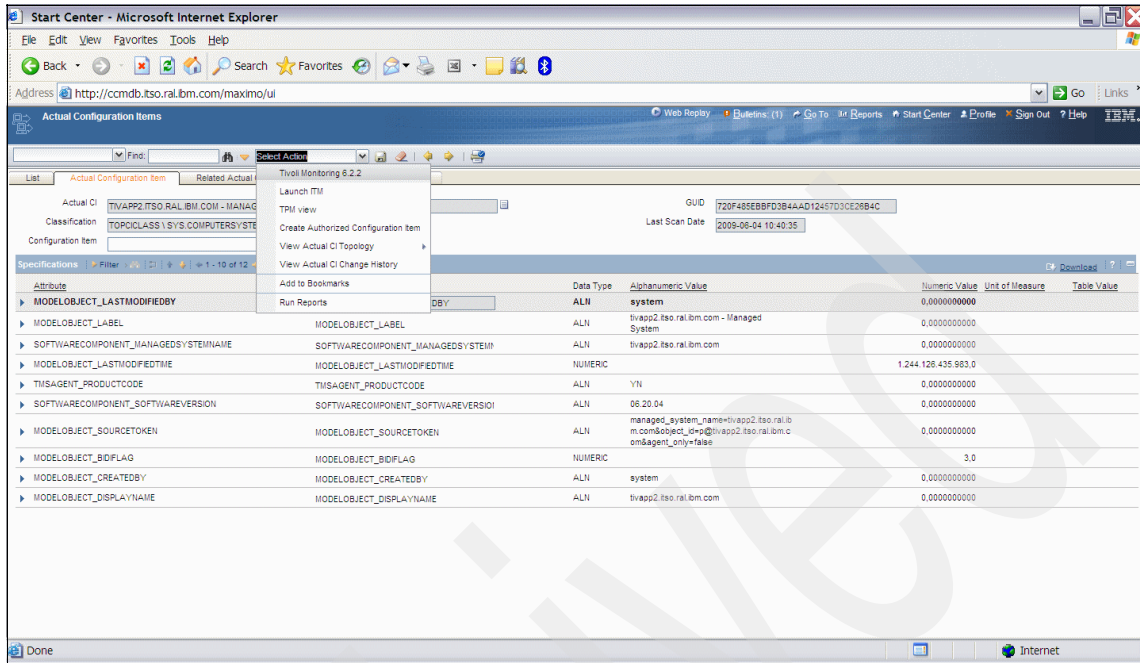


Figure 11-22 Launching IBM Tivoli Monitoring 6.2.2

21. Search for the WebSphere Application Server problem in this workspace (Figure 11-23 on page 434).

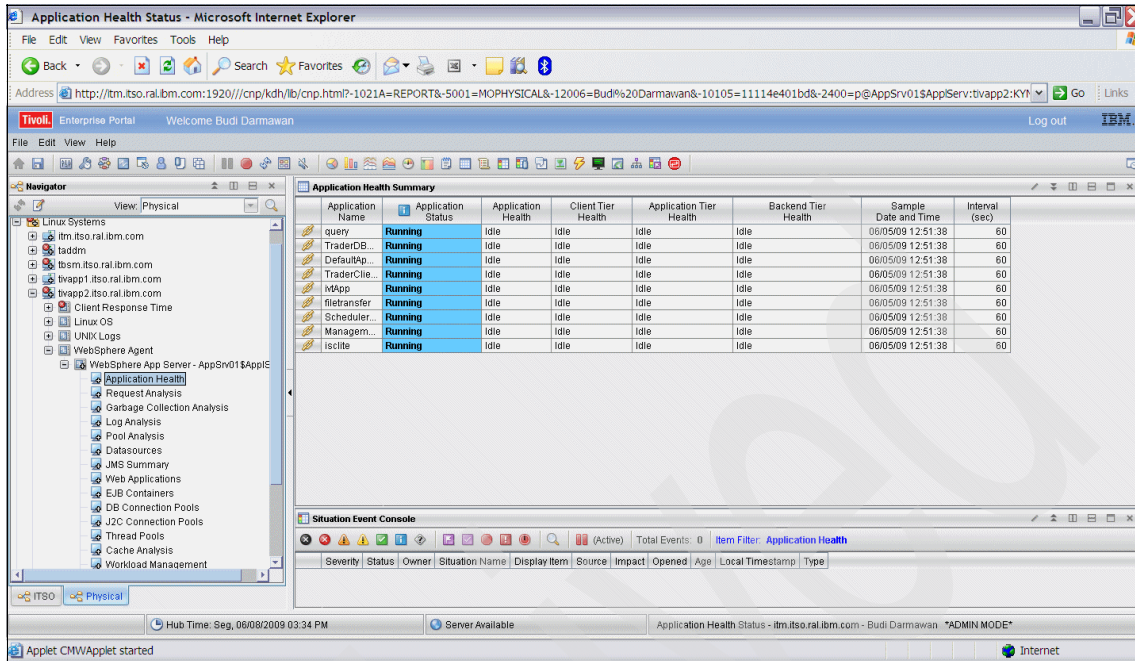


Figure 11-23 WebSphere Application Health View

22. The Change Manager finishes examining the impact of the change. Then, Change Manager changes the status of the Examine impact of the change task to **Closed** and the status of the Install software task to **Approved**. To do so, the Change Manager opens the change, chooses the **Related Records** tab, and chooses **Go To Work Order Tracking** from the work order field in the Related Work Orders section. The Change Manager chooses the **Plans** tab. The Change Manager sees the assigned tasks for the work order and changes the status as shown in Figure 11-24 on page 435.

Figure 11-24 Change the status of the tasks

23.The Deployment Specialist is notified of a new task, as shown in Figure 11-25.

Figure 11-25 Deployment Specialist is notified of a new task

24.The Deployment Specialist opens the task, as shown in Figure 11-26 on page 436.

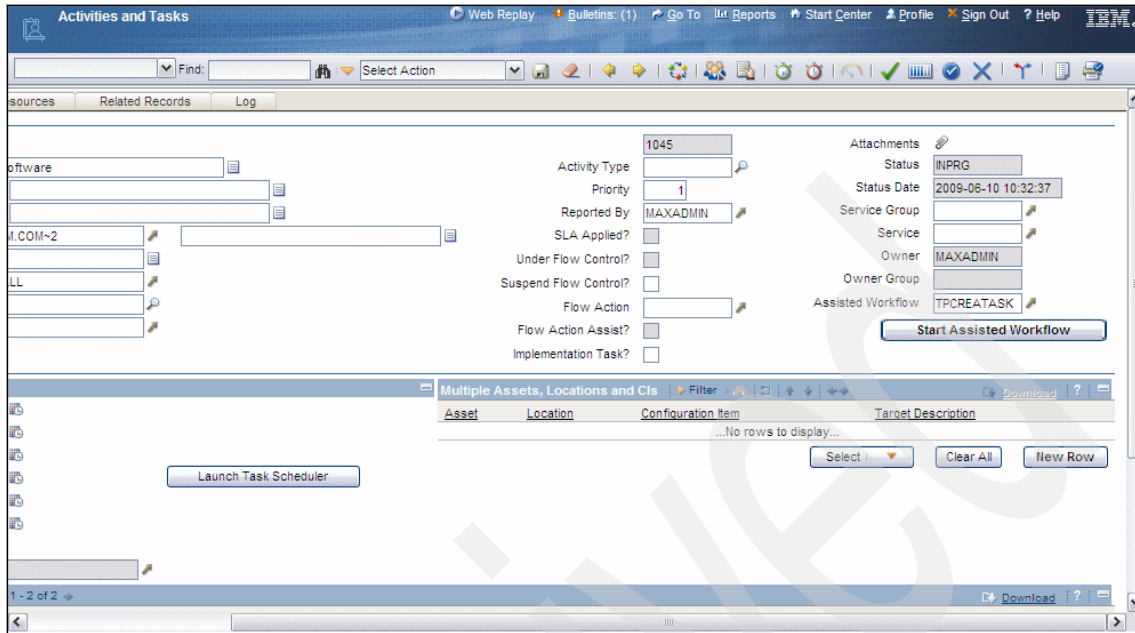


Figure 11-26 Open the assigned task

25. The Deployment Specialist reviews the task, schedules it, and clicks **Start Assisted Workflow**. The Tivoli Provisioning Manager Software Product Installation application opens. The Deployment Specialist clicks **Select Software** and chooses the **WebSphere Application Server 6.1 configuration file**, as shown in Figure 11-27 on page 437.

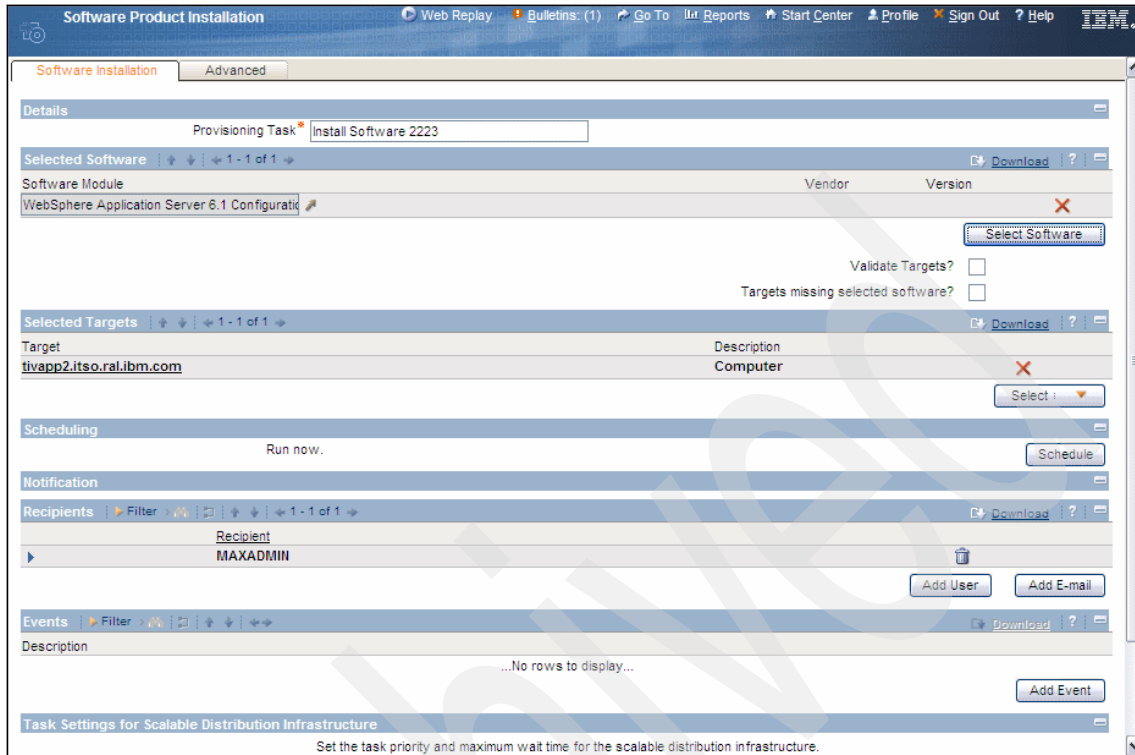


Figure 11-27 Software product installation

26. The Deployment Specialist chooses **Submit**, and the task succeeds, as shown in Figure 11-28 on page 438.

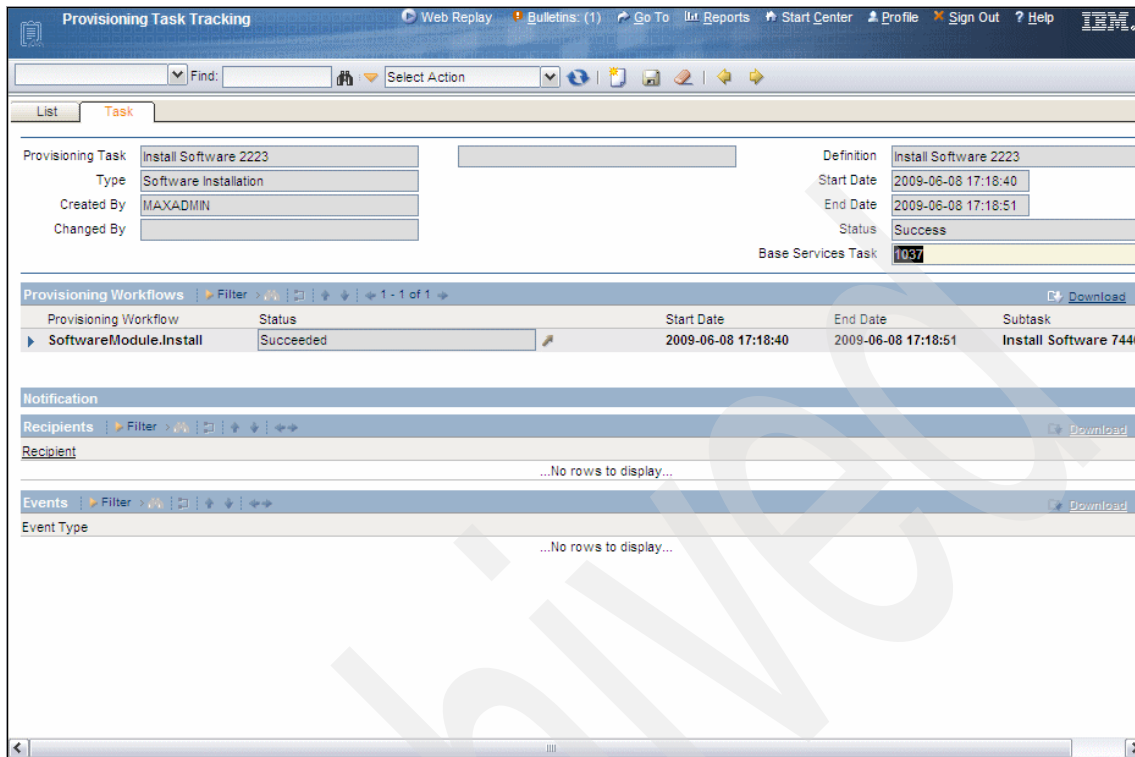


Figure 11-28 Task result

11.4 Summary of benefits

The benefits of the product integrations that are demonstrated in this scenario fall under these categories:

- ▶ Operational simplification: Assisted navigation and flow integration allows operation personnel to focus on the tasks of managing and implementing changes instead of having to deal with procedures and processes that need to be automated.
- ▶ Reduced errors: Automated data integration and the inherent navigation allows change record fields to be displayed for other applications, hence reducing operator work and minimizing human error.
- ▶ Faster operational response: The automation of several manual stages can improve the change implementation response time.

Abbreviations and acronyms

ACTCI	Actual Configuration Item	DMS	Device Management System
AEL	Active Event List	DMTF	Distributed Management Task Force
AIX	Advanced Interactive eXecutive	DNS	Domain Name Service
AMD	Advanced Micro Device	DN	Distinguished Name
API	Application Programming Interface	DSA	Data Source Adapter
BIRT	Business Intelligence and Reporting Tool	EIF	Event Information Facility
BPEL	Business Process Execution Language	ESDA	Enhanced SQL Data Adapter
BSM	Business Service Management	ESS	Embedded Security Service
CAS	Common Agent Services	FIPS	Federal Information Processing Standard
CCMDB	Change and Configuration Management Database	FQDN	Fully Qualified Domain Name
CDM	Common Data Model	GSK7	Global Security Key management 7
CDS	Content Delivery System	GUI	Graphical User Interface
CEC	Central Event Console	GUID	Globally User Identifier
CGI	Common Gateway Interface	HATR	How Things Are Running
CI	Configuration Item	HTML	Hyper Text Markup Language
CICS	Customer Information Control System	HTTP	Hyper Text Transfer Protocol
CIM	Common Information Model	HTTPS	Hyper Text Transfer Protocol Secure
CLI	Command Line Interface s	IBM	International Business Machines Corp
CMDB	Configuration Management Database	ICMP	Internet Control Message Protocol
CN	Common Name	IDML	Identity Markup Language
COI	Composite Offering Installer	IDS6	IBM Directory Server V6
CPU	Central Processing Unit	IETF	International Engineer Task Force
CSV	Comma Separated Values	IMS	Information Management System
DB2	Database 2	ISC	Integrated Solution Console
DLA	Discovery Library Adapter	ISM	IBM Service Management
DLFS	Discovery Library File Store		

ITCAM	IBM Tivoli Composite Application Manager	RAID	Redundant Array of Independent Disks
ITIC	IBM Tivoli Integration Composer	RDN	Relative Distinguished Name
ITIL	Information Technology Infrastructure Library	RFC	Request For Comments
ITM	IBM Tivoli Monitoring	RMI	Remote Method Invocation
ITNM	IBM Tivoli Network Manager	RMID	Resource Manager Identifier
ITSO	International	RPM	Red Hat Package Manager
J2EE	Java 2 Enterprise Edition	RSH	Remote Shell
JCE	Java Community Edition	SME	Subject Matter Expert
JDBC	Java Database Connectivity	SNIA	Storage Networking Industry Association
JNDI	Java Naming and Directory Interface	SNMP	Simple Network Management Protocol
JRE	Java Runtime Environment	SOA	Service-Oriented Architecture
LAN	Local Area Network	SOAP	Simple Object Access Protocol
LDAP	Lightweight Directory Access Protocol	SQL	Structured Query Language
LIC	Launch In Context	SSH	Secure Shell
LTPA	Lightweight Third Party Authentication	SSL	Secure Socket Layer
MIB	Management Information Base	SSO	Single Sign-On
MOSWOS	Manage Our Stuff With Our Stuff	STS	Secure Token Services
MTTR	Mean Time To Repair	TADDM	Tivoli Application Dependency Discovery Manager
NIC	Network Interface Card	TAM	Tivoli Access Manager
NTP	Network Time Protocol	TAMIT	Tivoli Asset Management for IT Resources
ODBC	Open Database Connectivity	TBSM	Tivoli Business Services Manager
OMP	Operation Management Product	TCP/IP	Transmission Control Protocol/Internet Protocol
OPAL	Open Process Automation Library	TCR	Tivoli Common Reporting
PAM	Pluggable Authentication Module	TDI	Tivoli Directory Integrator
PDF	Portable Document Format	TDW	Tivoli Data Warehouse
PID	Process Identifier	TEC	Tivoli Enterprise Console
PMR	Problem Management Record	TEMA	Tivoli Enterprise Management Agent

TEMS	Tivoli Enterprise Management Server
TEP	Tivoli Enterprise Portal
TEPS	Tivoli Enterprise Portal Server
TIP	Tivoli Integrated Portal
TPM	Tivoli Provisioning Manager
TTL	Time to Live
TWS	Tivoli Workload Manager
UID	User Identifier
URL	Uniform Resource Locator
WMI	Windows Management Instrumentation
XML	eXtensible Markup Language
XSD	XML Schema Definition

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 and IBM Redpaper publications

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 450. Note that several of the documents referenced here might be available in softcopy only:

- ▶ *End-to-End Patch Management Solution Using IBM Tivoli Configuration Manager 4.2.1*, SG24-6334
- ▶ *Deployment Guide Series: IBM Tivoli Configuration Manager*, SG24-6454
- ▶ *Certification Study Guide: IBM Tivoli Configuration Manager (ITCM) Version 4.2.2*, SG24-6691
- ▶ *End-to-end Automation with IBM Tivoli System Automation for Multiplatforms*, SG24-7117
- ▶ *Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments*, SG24-7143
- ▶ *IBM Tivoli Composite Application Manager Family Installation, Configuration, and Basic Usage*, SG24-7151
- ▶ *Getting Started with IBM Tivoli Workload Scheduler V8.3*, SG24-7237
- ▶ *Migrating to Netcool/Precision for IP Networks --Best Practices for Migrating from IBM Tivoli NetView*, SG24-7375
- ▶ *IBM Tivoli Asset Management for IT Portfolio Overview*, SG24-7376
- ▶ *IBM Tivoli Usage Accounting Manager V7.1 Handbook*, SG24-7404
- ▶ *IBM Tivoli Configuration Manager and Tivoli Provisioning Manager for Software Coexistence and Migration Considerations*, SG24-7426
- ▶ *IBM Tivoli Monitoring: Implementation and Performance Optimization for Large Scale Environments*, SG24-7443
- ▶ *Deployment Guide Series: IBM Tivoli Monitoring V6.2*, SG24-7444
- ▶ *Certification Study Guide Series: IBM Tivoli Monitoring V6.2*, SG24-7456

- ▶ *Deployment Guide Series: IBM Tivoli Composite Application Manager for Web Resources V6.2*, SG24-7485
- ▶ *IBM Tivoli Application Dependency Discovery Manager Capabilities and Best Practices*, SG24-7519
- ▶ *Deployment Guide Series: IBM Tivoli Workload Scheduler V8.4 and IBM Tivoli Dynamic Workload Broker V1.2*, SG24-7528
- ▶ *Best Practices for IBM Tivoli Enterprise Console to Netcool/OMNIBus Upgrade*, SG24-7557
- ▶ *Deployment Guide Series: IBM Tivoli CCMDB Overview and Deployment Planning*, SG24-7565
- ▶ *IBM Tivoli CCMDB Implementation Recommendations*, SG24-7567
- ▶ *Deployment Guide Series: IBM Tivoli Usage and Accounting Manager V7.1*, SG24-7569
- ▶ *Certification Guide Series: IBM Tivoli Composite Application Manager for Response Time V6.2 Implementation*, SG24-7572
- ▶ *Implementing IBM Tivoli Service Request Manager V7.1 Service Desk*, SG24-7579
- ▶ *Integration Guide for IBM Tivoli Service Request Manager V7.1*, SG24-7580
- ▶ *IT Asset Management Processes using Tivoli Asset Manager for IT*, SG24-7601
- ▶ *Deployment Guide Series: Tivoli IT Asset Management Portfolio*, SG24-7602
- ▶ *Implementing IBM Tivoli Service Request Manager V7.1 Service Catalog*, SG24-7613
- ▶ *Deployment Guide Series: IBM Tivoli Application Dependency Discovery Manager V7.1*, SG24-7616
- ▶ *Certification Guide Series: IBM Tivoli Workload Scheduler V8.4*, SG24-7628
- ▶ *Certification Guide Series: IBM Tivoli Usage and Accounting Manager V7.1 Implementation*, SG24-7692
- ▶ *Certification Guide Series: IBM Tivoli Netcool/OMNIBus V7.2 Implementation*, SG24-7753
- ▶ *Certification Guide Series: IBM Tivoli Netcool/Webtop V2.0 Implementation*, SG24-7754
- ▶ *Certification Guide Series: IBM Tivoli Netcool/Impact V4.0 Implementation*, SG24-7755
- ▶ *Certification Guide Series: IBM Tivoli Business Service Manager V4.1.1 Implementation*, SG24-7756

- ▶ *Certification Study Guide Series: IBM Tivoli Asset Management for IT V7.1, SG24-7762*
- ▶ *Certification Study Guide Series: Foundations of Tivoli Process Automation Engine, SG24-7763*
- ▶ *Certification Study Guide Series: IBM Tivoli Application Dependency Discovery Manager V7.1, SG24-7764*
- ▶ *IBM Tivoli Provisioning Manager V7.1.1: Deployment and IBM Service Management Integration Guide, SG24-7773*
- ▶ *Certification Study Guide for IBM Tivoli Configuration Manager 4.2, REDP-3946*
- ▶ *Provisioning J2EE Applications with Tivoli Intelligent Orchestrator and IBM WebSphere Application Server Network Deployment, REDP-4097*
- ▶ *Deploying Applications Using IBM Rational ClearCase and IBM Tivoli Provisioning Manager, REDP-4105*
- ▶ *Deploying Rational Applications with IBM Tivoli Configuration Manager, REDP-4171*
- ▶ *Composite Application Provisioning with IBM Tivoli Provisioning Manager V3.1, REDP-4222*
- ▶ *IBM Tivoli Business Service Manager V4.1, REDP-4288*
- ▶ *Synchronizing Data with IBM Tivoli Directory Integrator 6.1, REDP-4317*
- ▶ *Managing an SOA Environment with Tivoli, REDP-4318*
- ▶ *Creating EIF Events with Tivoli Directory Integrator for Tivoli Netcool/OMNibus and Tivoli Enterprise Console, REDP-4352*
- ▶ *Monitoring the IBM Tivoli Composite Application Management Server V6.1, REDP-4353*
- ▶ *IBM Tivoli Common Data Model: Guide to Best Practices, REDP-4389*
- ▶ *Tivoli Integration Scenarios, REDP-4401*
- ▶ *Creating IDML Discovery Books with IBM Tivoli Directory Integrator, REDP-4492*
- ▶ *Upgrading from Tivoli NetView 7.1.4/5 to IBM Tivoli Network Manager IP Edition 3.8 Version 1.0, REDP-4508*
- ▶ *End-To-End Availability and Performance Management using IBM Tivoli Solutions, REDP-4518*

Online resources

These Web sites are also relevant as further information sources:

- ▶ DB2 Information Center link
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/com.ibm.db2.luw.qb.server.doc/doc/t0008921.html>
- ▶ Tivoli Information Center links for IBM Tivoli Change and Configuration Management Database
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.ccmb.doc_7.1.1/install/c_ccmb_ccmbcmiddlewareoverview.html
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.ccmb.doc_7.1.1/install/c_ccmb_icoverview.html
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.ccmb.doc_7.1.1/install/t_ccmb_installfoundation.html
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.ccmb.doc_7.1.1/install/t_lta_acinstall.html
- ▶ Tivoli Information Center link for Tivoli Release Process Manager
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.rpm.doc_7.1.1/rpm/t_rpm_install_gui.html
- ▶ Tivoli Information Center link for IBM Tivoli Service Request Manager
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.srm.doc_7.1/installing/src/t_installing_srm.html
- ▶ Tivoli Information Center links for IBM Tivoli Application Dependency Discovery Manager
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/AdminGuide/c_ccmb_moswos_using.html
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/DLADevGuide/c_cmdbsdk_dla_introducing.html
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/InstallGuide/c_ccmb_install_troubleshooting.html
http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/InstallGuide/c_ccmb_moswos_installoverview.html

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/InstallGuide/t_cmdb_install_prereq.html

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/InstallGuide/t_cmdb_install_simple_db2.html

http://publib.boulder.ibm.com/infocenter/tivihelp/v10r1/topic/com.ibm.taddm.doc_7.1.2/SDKDevGuide/r_cmdbsdk_cliapi_commandsyntax.html

► Tivoli Information Center links for IBM Tivoli Provisioning Manager

http://publib.boulder.ibm.com/infocenter/tivihelp/v11r1/topic/com.ibm.tivoli.tpm.admin.doc/book/part_intro.html

http://publib.boulder.ibm.com/infocenter/tivihelp/v11r1/topic/com.ibm.tivoli.tpm.ins.doc/install/tins_cygwin.html

http://publib.boulder.ibm.com/infocenter/tivihelp/v11r1/topic/com.ibm.tivoli.tpm.ins.doc/install/tins_insttpmweb.html

http://publib.boulder.ibm.com/infocenter/tivihelp/v11r1/topic/com.ibm.tivoli.tpm.ins.doc/install/tins_srmwithtpm71.html

► Tivoli Information Center links for IBM Tivoli Monitoring

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp?topic=/com.ibm.itm.doc_6.2.1/itm_install122.htm

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp?topic=/com.ibm.itm.doc_6.2.1/itm_install161.htm

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp?topic=/com.ibm.itm.doc_6.2.1/itm_install33.htm

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.1/itm_admin.htm

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.1/itm_install191.htm

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.1/main_win.htm

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.2/launch_intro_c.htm

► Tivoli Information Center links for IBM Tivoli Directory Server

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?toc=/com.ibm.IBMD5.doc/toc.xml>

<http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.IBMDI.doc/adminguide.htm>

► Tivoli Information Center links for IBM Tivoli Workload Scheduler

<http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?toc=/com.ibm.tivoli.itws.doc/toc.xml>

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tivoli.itws.doc/awspimst176.htm#webui_wizard

<http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tivoli.itws.doc/awspimst61.htm#freshinstall>

► Tivoli Information Center links for IBM Tivoli Usage and Accounting Manager

<http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=/com.ibm.ituam.doc/welcome.htm>

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.ituam.doc/config/c_configuring_after_installation.html

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.ituam.doc/install/c_supported_hardware_and_software.html

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.ituam.doc/install/t_installing_ecp_on_unix.html

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.ituam.doc/install/t_installing_ecp_on_win.html

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.ituam.doc/install/t_installing_ee_ve_on_unix.html

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.ituam.doc/install/t_installing_ee_ve_on_win.html

http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.ituam.doc_7.1.2/admin_win_dc/c_tdw_collector.html

- ▶ Tivoli Information Center links for IBM Tivoli Asset Manager for IT Resources
http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tamit.doc_7.1/pdf/tamit71_install_was.pdf
- ▶ Tivoli Information Center links for IBM Tivoli Business Service Manager
http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.tivoli.itbsm.doc/installguide/bsmi_t_installing_tbsm_agent.html
- ▶ Tivoli Information Center links for IBM Tivoli Netcool/OMNIBus
http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?topic=/com.ibm.netcool_OMNIBus.doc/welcome.htm
http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.netcool_OMNIBus.doc_7.2.1/admin/concept/omn_adm_pa_usingpctrltorunxtrnlprocs.html
http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.netcool_OMNIBus.doc_7.2.1/om_net_agent_template26.htm
- ▶ Tivoli Information Center link for IBM Tivoli Network Manager for IP
http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?topic=/com.ibm.networkmanagerip.doc_3.8/itnm/ip/3.8/admin/reference/nmi_p_adm_urltooleg.html
- ▶ Tivoli Open Process Automation Library pages
<http://www.ibm.com/software/tivoli/opal>
<http://www.ibm.com/software/tivoli/opal/details?catalog.label=1TW10C1Q>
<http://www.ibm.com/software/tivoli/opal/details?catalog.label=1TW10C1U>
<http://www.ibm.com/software/tivoli/opal/details?catalog.label=1TW10C1Y>
<http://www.ibm.com/software/tivoli/opal/details?catalog.label=1TW100T02>
- ▶ Tivoli Common Reporting
<http://www.ibm.com/developerworks/spaces/tcr>
- ▶ WebSphere Security white paper
http://www.ibm.com/developerworks/websphere/techjournal/0701_ilechko/0701_ilechko.html

- ▶ Tivoli Support site
http://www.ibm.com/software/sysmgmt/products/support/__TivoliProduct__.html
- ▶ Support page for Tivoli Workload Scheduler
<http://www-01.ibm.com/support/docview.wss?rs=672&uid=swg27012175>
- ▶ Discovery Library Adapters list
<https://www.ibm.com/developerworks/wikis/display/tivoliaddm/Discovery+Library+Adapters>

How to get IBM Redbooks publications

You can search for, view, or download IBM Redbooks publications, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy IBM 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

Symbols

`./loadidml.sh` command 168
`./nco_g_tivoli_eif` command 213
`./nco_p_tivoli_eif` command 215
`./nco_patch` command 192
`./ncp_dla.sh` command 168
`./start_tdi.sh` command 212
`./startSUF.sh` command 187
`./tbsmrdr_start.sh` command 162
`.bash_profile` 39
`/ncp_dla.sh` command 175
`$COLLATION_HOME` 93
`$NCHOME/bin/nco_igen` command 199
`$OMNIHOME/bin/nco_g_crypt` command 199
`$OMNIHOME/bin/nco_pa_status` command 215

A

Actual CI 136
agent deployment 18
alerts.status table 196
Apache Tomcat 51
`api.sh` command 135, 175
aspects of integration 5
Authorized CI 137

B

`bindDN` 48
BIRT 219
BPEL
`bulkload.properties` 144
Business Intelligence Reporting Tool, see BIRT
Business Process Execution Language, see BPEL

C

CAS 25
`cat` command 162
CCMDB 19, 131
CDM 130, 132, 153, 155, 246
CDS 25
CEC 370, 390
CGI program definition 313
Change and Configuration Management Database,

see CCMDB

CI 135

CIM

`cinfo` command 330

Client Response Time, see CRT

`cmdbdiscovery` command 168

CN 123

COI 19

`COLLATION_HOME` 94

collector 229

commands

`./loadidml.sh` 168

`./nco_g_tivoli_eif` 213

`./nco_p_tivoli_eif` 215

`./nco_patch` 192

`./ncp_dla.sh` 168

`./start_tdi.sh` 212

`./startSUF.sh` 187

`./tbsmrdr_start.sh` 162

`/ncp_dla.sh` 175

`$NCHOME/bin/nco_igen` 199

`$OMNIHOME/bin/nco_g_crypt` 199

`$OMNIHOME/bin/nco_pa_status` 215

`api.sh` 135, 175

`cat` 162

`cinfo` 330

`cmdbdiscovery` 168

`enableISCLite.sh` 127

`export` 215

`itmcmd` 29, 124, 368

`ITMconfig.sh` 332

`keytool` 73

`launchpad.sh` 331

`loadidml` 151

`man` 97

`nco_config` 100

`nco_pa_crypt` 98

`nco_pa_status` 98

`nco_pad` 213

`nco_patch` 197

`nco_sql` 183

`start` 301

`tacmd` 367

`tbsm_suite.sh` 301

- tmsdla 151
- tmsdla.sh 247
- touch 103
- update-impact-roles.sh 117
- updateTEPSEPass.sh 123
- wsadmin 55
- xslt.sh 175
- Common Agent Services, see CAS
- Common Data Model, see CDM
- common data model, see CDM
- common elements 12
- Common Event Console, see CEC
- Common Information Model, see CIM
- Common Name, see CN
- Common Offering Installer, see COI
- configuration 13
- configuration item, see CI
- Content Delivery Service, see CDS
- cron task 87
- CRT 374
- Cygwin 26

D

- data integration
 - event 130
 - resources 130
- Data server port 256
- DB2 Runtime Client 41
- db2inst1 32
- DefaultWIMFileBasedRealm 54, 71
- DEPOTHOME 34
- Device Management Service, see DMS
- Directory Information Tree, see DIT
- directory tree 46
- Discover Library Adapter, see DLA
- Discovery Library Adapter, see DLA
- Discovery Library Adapters, see DLA
- discovery library books 144
- Discovery Library File Store, see DLFS
- Discovery Library Toolkit 157
- distinguished name, see DN
- Distributed Management Task Force, see DMTF
- DIT 46
- DLA 132, 151, 155
- DLAtoCCMDB.log 174
- DLAtoCCMDB.properties 170
- DLFS 152, 261
- DMS 25

- DMTF
- DN 46
- document organization 6
- duplicate users 113

E

- Embedded Security Services, see ESS
- embedded WebSphere 56
- enableISCLite.sh 122
- enableISCLite.sh command 127
- Enterprise Collector Pack 42
- Enterprise Edition 42
- entityGUIDcache 163
- ESS 50
- event data 176
- event data integration 130
- event filtering 218
- event forwarding 189
- event integration 176
- export command 215

F

- Federated repositories 48
- financial reports 229

G

- Graphical User Interface, see GUI
- GUI 232

H

- high availability 19
- Hub Tivoli Enterprise Monitoring Server 31

I

- IBM 35
- IBM Service Management 19
- IBM Tivoli Application Dependency Discovery Manager 22
- IBM Tivoli Business Service Manager 35
- IBM Tivoli Change and Configuration Management Database 21
- IBM Tivoli Directory Integrator 194
- IBM Tivoli Directory Server 21, 54
- IBM Tivoli Integration Composer 22, 136
- IBM Tivoli Monitoring 28
- IBM Tivoli Netcool/Impact 35
- IBM Tivoli Netcool/OMNIBus 35

IBM Tivoli Netcool/Webtop 35
IBM Tivoli Network Manager for IP 35
IBM Tivoli Provisioning Manager 24
IBM Tivoli Service Request Manager 22
ICMP 330
IETF 48
Information Technology Infrastructure Library, see
ITIL
installation 18
integration
 aspects 5
 launch 5
integration elements 12
Integration Toolkit 194
Internet Control Message Protocol, see ICMP
Internet Engineering Task Force, see IETF
isclite.ear 56
IT Infrastructure Library, see ITIL
ITIL 19
itm_db_update.sql 183
itm_proc.sql 183
itm_sync.sql 184
itm_tivoli_eif.rules 186
itmcmd command 29, 124, 368
ITMconfig.sh command 332
ITMSSOEntry 55, 121
itmuser 32
itsorealm 54

K

keytool command 73
klz_tmsdla.xml 155
knt_tmsdla.xml 155
kux_tmsdla.xml 155

L

launch 5
launch in context 5
launch in context, see LIC
launchpad.sh command 331
LDAP 120
ldap.conf 96
leaf nodes 47
LIC 13
Lightweight Directory Access Protocol, see LDAP
Light-weight Third Party Authentication, see LTPA
loadidml command 151
loadidml.sh 144

LTPA 52
LTPA key 85
LTPA token 52

M

man command 97
MAXGROUP table 92
Mean Time to Recovery, see MTTR
Menu item definition 315
middleware installer 21
Monitoring coverage report 153
MTTR 402
MXOS 209

N

NCIM 37
nco_config command 100
nco_objserv 99
nco_pa_crypt command 98
nco_pa_status command 98
nco_pad command 213
nco_patch command 197
nco_sql command 183
nco-g-tivoi-eif 192
ncp_dla.properties 168
network time protocol, see NTP
NTP 53

O

OPAL 134, 220
Open Process Automation Library, see OPAL
Operational Management Products 143
organization 6

P

PAM
pam_ldap.so 96
pam.d 97
PERSON table 92
Pluggable Authentication Modules, see PAM
product installation 18

R

RAID disks 19
Rational Agent Controller 21
realm name 54, 85
Redbooks Web site 450

- Contact us xvi
- redundant environment 19
- relative distinguished names 46
- report integration 130
- report list 227
- reports packages 221
- resource data integration 130
- rootbinddn 96

S

- Scalable Deployment Infrastructure, see SDI
- SCR 155
- SDI 27, 241
- Security Configuration Wizard 58
- Security Token Services, see STS
- security.xml 112
- Service Component Repository, see SCR
- session cookie 56
- shared console 6
- Simple Network Management Protocol, see SNMP
- Single Sign On, see SSO
- Situation Update Forwarder 180
- SNIA
- SNMP 330
- SOAP server 181
- Software Distribution Infrastructure, see SDI
- SSO 67, 120
- SSO key 56
- start command 301
- Storage Networking Industry Association, see SNIA
- STS 50
- suffix 64
- Summarization and Pruning Agent 32
- System Automation 19

T

- tacmd command 367
- TADDM 133
- tbsm_suite.sh command 301
- TCP domain 84
- TCR
- TEMS 189
- TEP 343
- TEPS/e 121
- TIP 6, 218
- Tivoli Application Dependency Discovery Manager, see TADDM
- Tivoli Base Services 21

- Tivoli Common Reporting 21
- Tivoli Common Reporting, see TCR
- Tivoli Data Warehouse 32
- Tivoli Dynamic Workload Console 39
- Tivoli Enterprise Monitoring Server 31
- Tivoli Enterprise Monitoring Server, see TEMS
- Tivoli Enterprise Portal 31
- Tivoli Enterprise Portal Server 31
- Tivoli Enterprise Portal, see TEP
- Tivoli Integrated Portal 35
- Tivoli Integrated Portal, see TIP
- Tivoli Provisioning Manager 24
- Tivoli Provisioning Manager, see TPM
- Tivoli Usage and Accounting Manager 41
- Tivoli Workload Scheduler 39
- tmsdla command 151
- tmsdla.sh command 247
- TOPCICLASS 137
- touch command 103
- TPM 266, 343
- TPM_V710_CoreComp_Win32 26
- TPM_V710_Install 26
- TPM_V710_Midlwr_Win32 26
- tsrm_eif.sql 198

U

- UML 132
- Unified Modelling Language, see UML
- update-impact-roles.sh command 117
- updateTEPSEPass.sh command 123
- usage data 229
- user roles 117

V

- Virtual Member Manager, see VMM
- VMM 48
- VMMSYNC 87

W

- Warehouse Proxy Agent 32
- wasadmin 32
- WebSphere Identity Manager, see WIM
- WIM 48
- wimconfig.xml 49
- wsadmin command 55

X

XML Schema Document, see XSD

XSD 208

xslt.sh command 175

Archived



Redbooks

Integrating Tivoli Products

(1.0" spine)
0.875" <-> 1.498"
460 <-> 788 pages



Integrating Tivoli Products



Describes security, data, navigation, reporting, and task integration

Provides extensive samples with a scenario-based approach

Discusses Tivoli product interoperability

This IBM Redbooks publication attempts to provide a broad view of how Tivoli system management products work together for use in several common scenarios. Seamless integration must be achieved for operation personnel to work with the solution. This integration is necessary to ensure that the product can be used easily by the user. Product integration contains multiple dimensions. We evaluate the implementation of the following areas:

- ▶ Security integration provides authentication to users from a single repository and a single login to multiple applications.
- ▶ Navigation integration allows multiple management applications to work with each other so that users can get more information across multiple applications.
- ▶ Data integration allows the exchange of resource or object information between solutions to allow the context to be established seamlessly.
- ▶ Task integration permits one product to use and invoke a facility of another product.

We discuss both the implementation and also provide sample scenarios of how these integrations work. We based the scenarios on common real-life examples that IT operations often experience.

Finally, we also include additional information about topics, such as agent management, reporting, and product adoption routes.

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